EXHIBIT 9 PART 1

Expert Report of Edward Fox, Ph.D. On the Invalidity of U.S. Patent No. 7,917,843

Arendi S.A.R.L. v. Google LLC, Case No. 13-0919-LPS (D. Del.)
Arendi S.A.R.L. v. Motorola Mobility LLC, Case No. 12-1601-LPS (D. Del.)

TABLE OF CONTENTS

Page

I. INTRODUCTION

A. My Assignment

1. I have been retained as an expert in this case by counsel for Defendants Google LLC ("Google") and Motorola Mobility LLC ("Motorola"). Although this report contains my opinions with regard to the asserted claims against both Google and Motorola, I intend to offer testimony at trial as to Google and Motorola, individually. I expect to testify at trial regarding the matters set forth in this report, including the technical background and state of the art relevant to the asserted claims of U.S. Patent No. 7,917,843 ("the '843 patent"), if asked about these matters by the Court or by the parties' attorneys. Additionally, I may discuss my own work in the field, and knowledge of the state of the art in the relevant time period. I may rely on handbooks, textbooks, technical literature, and the like to demonstrate the state of the art in the relevant period and the evolution of the relevant technologies. I may also rely upon witness testimony or any other evidence that I obtained during the pendency of this case or was introduced at trial. A list of the materials I have considered in preparing this report is identified in my report or is listed in Ex. A.

2. I have been asked for my expert opinion concerning the validity of the asserted claims and related issues presented in this report. I conclude that the asserted claims are invalid for the reasons stated below.

3. I understand that the Plaintiff Arendi S.A.R.L. ("Arendi") may submit one or more expert reports responding to this report. I reserve the right to rebut any positions taken in those reports and to consider any new evidence presented in the case. I also reserve the right to supplement my opinions based on any new or clarified claim constructions provided by the Court or any other rulings from the Court impacting the scope of the asserted claims.

B. Background and CV Materials

4. As shown by a copy of my CV attached as Ex. B, my professional career has spanned over 36 years. Based on my academic and work experience with information retrieval, interactive information systems, web archiving, digital libraries, hypertext/hypermedia, and Web/Internet technologies, I believe I am well-positioned to understand and address the skills and mindset of a person of ordinary skill in the art ("POSITA") in this field circa 1997-1998.[1]

5. My education includes a B.S. in Electrical Engineering (Computer Science Option) from the Massachusetts Institute of Technology (MIT) completed in 1972, followed by an M.S. in Computer Science from Cornell University in 1981. I received a Ph.D. in Computer Science from Cornell University in 1983. My undergraduate advisor was J. C. R. Licklider, then Director of Project MAC, who, when working at DARPA, managed projects that led to the Internet. My graduate advisor was Gerard Salton, often called the father of information retrieval (the field that works with search engines). My doctoral dissertation (1983) considered bibliographic records with author names, document text analysis, text matching, and database calls in systems with multiple interacting programs.

6. Before college, starting in 1965, I took courses about computing, first at Columbia University on Saturdays, and then at Stevens Institute of Technology in the summer, working with parsing and analysis of text files. As an undergraduate at MIT, I worked with early editors and text processors. During one summer job, I worked to explore how to automate newspapers, including studying the latest electronic publishing technologies. My B.S. thesis concerned collecting electronic texts, document text analysis, searching over a collection, and text matching such as of queries with documents. During another summer I programmed a PDP-8 with display processor and light pen, exploring early human-computer interaction

[1] I provide the definition of a POSITA later in my report.

methods and user interfaces. During the academic year I was paid to help users of early time-sharing computers such as CTSS. As founder of the Student Information Processing Board (SIPB, still operating at MIT), I interacted with many around campus with computing systems, helping students gain access.

7. I have been a professor of Computer Science for more than 36 years, teaching courses each year, including about information retrieval, digital libraries, hypertext, database management, data structures, artificial intelligence, networked information, big data, text summarization, computational linguistics / natural language processing, and multimedia as well as textual information. From September 1983 through May 1988, I served as an Assistant Professor of Computer Science at Virginia Polytechnic Institute and State University (Virginia Tech). I served Virginia Tech as an Associate Professor until April 1995, when I was promoted to Professor. I have continued in that capacity since, but also became a Professor, by courtesy, in Virginia Tech's Department of Electrical and Computer Engineering, in February 2016. Since January 1998, I have been the Director of the Digital Library Research Laboratory at Virginia Tech. From June 1990 to June 2014, I was the Associate Director for Research at Virginia Tech's Computing Center, a position that evolved to Faculty Advisor to Information Technology. I received an award on 16 October 2015 for Research Impact in Human-Computer Interaction, conferred by the VT Center for HCI, for which I served as a founding member; my relevant activities included working on information visualization, user interfaces, forms for managing information, and connections with database and information retrieval systems.

8. Before that, from August 1982 to April 1983, I was Manager of Information Systems at the International Institute of Tropical Agriculture, in Ibadan, Nigeria, where I helped with automation of library operations. From September 1978 to August 1982, I was Instructor,

Research Assistant, and Teaching Assistant at the Department of Computer Science at Cornell University. From September 1972 to August 1978, in Florence, SC, I was the Data Processing Manager in the Vulcraft Division of NUCOR Corporation. From September 1971 to June 1972, I was a Data Processing Instructor at Florence Darlington Technical College.

9. While at Vulcraft, I led implementation of a system for payroll, and supported the sales department. These activities involved working with names and addresses, pulling that information from databases, inserting names and addresses into documents and forms, and allowing editing and updating of those names and addresses.

10. During the summer of 1979, I worked at IBM FSD in Owego, NY. There I served as assistant to the database administrator. I worked with System R, often regarded as the first large relational database system, which was a precursor to other IBM products like DB2. Also during my period at Cornell, I served as graduate teaching assistant for a course on database management. By 1982 I had led implementation of a new version of the SMART system, a research vehicle for study of search engines. This stored information in the INGRES database management system, allowing pulling out from the database of information like names. It supported document text analysis, text matching, and database calls.

11. I am a member of the Association for Computing Machinery (ACM) (since 1967) and its Special Interest Group on Information Retrieval (SIGIR), which I served from 1987-1995 as vice chairman and then chairman. I served from 1988-1991 on the ACM Publications Board, and again in 2018-2019 on that Board as co-chair of its Digital Libraries Committee. ACM has awarded me seven recognition of service awards. I am a Fellow of ACM (cited for contributions in information retrieval and digital libraries) and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) (cited for leadership in digital libraries and information retrieval),

as well as a member of IEEE-CS. Regarding its Technical Committee on Digital Libraries, I served through 2018 on its Executive Committee, and was its chairman 2004-2008. I also am a member of the Association for Information Science & Technology and Sigma Xi (since 1972).

12. Since 1987, I have led activities so that theses and dissertations could be prepared, archived, and made accessible in electronic forms (e.g., over the World Wide Web). Since 1996, I have served as Executive Director of the Networked Digital Library of Theses and Dissertations (NDLTD), which was incorporated in 2003. I also serve as Founder and Chairman of the Board for NDLTD, which now has over 5.9 million records for theses and dissertations in its Union Catalog. Since the beginning, this work has concerned electronic publishing, including use of word processors, with varied formats of textual documents, including Word and PDF, and related handling of metadata, including author names, as well as related search and archiving technologies and standards. In addition, I have been involved since 1983 in another part of widespread sharing of scholarly works, by way of technical reports, and so can attest to their suitability as prior art publications. For example, my Cornell University Department of Computer Science technical report TR83-561, "Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts," from 1983, has been cited more than 125 times according to Google Scholar. My expertise regarding technical reports is further documented by professional service, publication, and funded project activities such as: (a) K. Maly, E. Fox, J. French, and A. Selman. Wide Area Technical Report Service, TR_92_44, Old Dominion Univ. Dept. of Computer Science, Dec. 1992; (b) E. Fox, Technical Reports / Dissertations. Invited project presentation for Monticello Electronic Library meeting, sponsored by SURA, SURAnet, SOLINET, and NSF, July 29-30, 1993, Atlanta; (c) National Science Foundation CISE Grant NSF-CDA-9308259: Wide Area

Technical Report Server, principal investigators: K. Maly (ODU), E. Fox, J. French (UVA), and A. Selman (SUNY Buffalo); (d) Chair: Workshop for Working Group on Theses, Technical Reports, and Dissertations, in Monticello Electronic Library Initiative, sponsored by SURA and SOLINET, Aug. 12-13, 1994, Virginia Tech, Blacksburg, VA; (e) K. Maly, J. French, A. Selman and E. Fox. The Wide Area Technical Report Service. In Proc. Second International WWW '94: Mosaic and the Web, WWW'94, Chicago, IL, Oct. 17-20, 1994, 523-533; (f) E. Fox. World-Wide Web and Computer Science Reports. Commun. of the ACM, Apr. 1995, 38(4):43-44; (g) J. French, E. Fox, K. Maly, and A. Selman. Wide Area Technical Report Service --- technical reports online. Commun. of the ACM, Apr. 1995, 38(4):45; (h) 1995 Organizing Committee member, NSF hosted workshop on Computer Science Technical Reports, NSF, Arlington, VA, April 7-8, 1995; and (i) 1995-2001 Member, NCSTRL (Networked CS Tech. Report Library) working group (part of D-Lib working group program).

13. Since December 2018, I have served as Chief Technology Officer for Mayfair Group, LLC. Mayfair develops World Wide Web based services that include the application of natural language processing, machine learning, and other technologies to summarizing text documents, in support of the insurance and legal industries.

14. I have been the recipient of a number of honors and awards relating to my work in information retrieval and other areas. One related honor, at Virginia Tech, is the Xcaliber Award 2016 "for extraordinary contributions to technology-enriched learning activities" for the project "Enhanced problem-based learning connecting big data research with classes." I have taught classes on information retrieval since the early 1980s and about multimedia/hypertext/information access since the early 1990s. I am one of the inaugural inductees to the ACM SIGIR Academy, class of 2020, aimed "to honor and recognize

individuals who have made significant, cumulative contributions to the development of the field of information retrieval (IR). Inductees to the SIGIR Academy are the principal leaders in IR, whose efforts have shaped the discipline and/or industry through significant research, innovation, and/or service." (*See* Announcement and Call for Nominations: ACM SIGIR Academy, Ryen W. White, ACM SIGIR Forum Vol. 54 No. 1 (June 2020).)

15. I have held numerous board positions in various editorial, professional, and industry organizations and groups. For example, I served from 2010-2013 as an elected member of the Computing Research Association Board, broadly representing the U.S. computing research community.

16. I have a background in many of the key areas related to handling information with computers, including information retrieval, digital libraries, Web archiving, and related human-computer interaction. This work involves theory, algorithms, systems, interfaces, and user studies, specifically involving search engines, database management, big data, data analytics, machine learning, and natural language processing.

17. In these areas, I have participated in, organized, and presented at numerous conferences and workshops, and have conducted 87 tutorials in over 28 countries. I have received 135 grants to fund my research and have (co)authored 19 books and edited book series for two publishers. I have (co)supervised over 80 graduate students for their dissertation, thesis, or project. In addition, I have (co)authored 136 journal or magazine articles. I have 634 related keynotes, papers, book chapters, posters, demonstrations, and reports as well as over 350 other presentations. Google Scholar has reported that my works have been cited over 18,160 times, with an h-index of 60 and i10-index of 253. Topics covered in the above-mentioned works include: computers, programs, files, memory, access, communication, networks, links, linked

data, representations, tables, databases, servers, World Wide Web, HTML, proxies, hypertext, hypermedia, and queries. Since the late 1980s, my research has dealt with document text analysis, text matching, hypertext creation, distributed processing including Application Programming Interfaces ("APIs") and other communication mechanisms, building and using database systems, and applications with user interfaces.

18. I have been involved in numerous software and information systems projects and products over the years and have supervised hundreds of student team projects of these types. I have managed workstations since 1982, have worked with Apple Macintosh computers since 1985, and have employed WWW browsers since 1993.

19. I am an inventor on U.S. Patent No. 7,346,621, issued March 18, 2008, titled "Method and System for Ranking Objects Based on Intra-type and Inter-type Relationships." I am lead inventor on U.S. Provisional Patent Application Serial No. 62/945,202; filed December 8, 2019; titled "Methods and Systems for Generating Declarative Statements Given Documents with Questions and Answers," as well as U.S. Provisional Patent Application Serial No. 63/039,725; filed June 16, 2020; titled "Methods and Systems for Generating Summaries Given Documents with Questions and Answers." I have a general understanding of the U.S. patent prosecution process and of the novelty and non-obviousness requirements for patentability.

20. I am not, and have never been, an employee of the Plaintiff Arendi S.A.R.L., or of any of the defendants of the cases captioned above.

21. I have received no compensation for this Declaration beyond my normal hourly compensation based on my time actually spent studying the matter and working on the

Declaration, and I will not receive any added compensation based on the outcome of the above-mentioned patent infringement suit.

C. Previous Testimony

22. My Consulting Summary List (Ex. C) includes a list of all cases during which in the past 5 years I have testified as an expert at trial or deposition.

D. Compensation

23. I am being compensated for my time for all work in conjunction with this expert report at $500/hour, but $600/hour for time spent testifying in deposition or trial. This was my customary rate for this kind of work at the time I was engaged on this matter.

24. The opinions expressed in this report are my own, unless specifically otherwise stated in this report. My compensation does not depend on my testimony, nor on the outcome of this matter.

II. THE LAW OF PATENT INVALIDITY

25. The '843 patent relates back to an application filed in 1998; therefore, I understand that pre-AIA law applies to this patent.

26. I am not an attorney and will offer no opinions on the law. I have relied on instructions from counsel as to the applicable legal standards to use in arriving at my opinions in this report.

27. Below is my understanding regarding the applicable legal standards related to anticipation, obviousness, written description, enablement and related topics. I have been informed of the same by counsel.

28. I understand that a patent is presumed valid. A party asserting invalidity of a U.S. Patent bears the burden of proving invalidity by clear and convincing evidence. Clear and convincing evidence is an evidentiary standard that is higher than a preponderance of the

evidence but lower than beyond a reasonable doubt. Such questions of validity are applied from the perspective of a POSITA at the time of the alleged invention, using the claim language and the Court's claim constructions.

A. Anticipation

29. Evaluation of whether a patent claim is "anticipated" is a two-step process. In the first step, the language of the claim is construed as it would be understood by one of ordinary skill in the art at the time of the filing of the patent application. The claim is construed by referring to intrinsic evidence, which includes the claim language, the patent specification, and the prosecution history. The words of patent claims are to be given their ordinary or customary meaning unless the inventor has defined them (acted as his/her own lexicographer) or used them differently (i.e., in a manner inconsistent with the ordinary and customary meaning). The prosecution history of a patent, and related patents and applications, including the history of post-grant proceedings like reexaminations, may limit the interpretation of the claim, especially if the patentee disavowed or disclaimed any claim scope in order to obtain allowance of the claim.

30. I understand that the Court has issued a Markman Opinion and Order in which it has construed a number of claim terms in the patent, and if there are any remaining disputed claim terms, the Court may construe those as well. My testimony is based upon a review of, and application of, the Court's claim constructions. In the absence of a guiding claim construction as to a particular term, I have applied my understanding of how one of ordinary skill in the art would have interpreted the term. As I explain further later in my report, in some instances I have used Arendi's application of a claim term.

31. The second step of the anticipation evaluation, after the language of the patent claims has been construed, is a comparison of the properly construed claim language to the prior art on a limitation-by-limitation basis.

32. A claimed invention is "anticipated" if each and every limitation of the claim, as properly construed, has been disclosed in a single prior art reference, or has been embodied in a single prior art device, system, or practice, either explicitly or inherently (i.e., necessarily present even if not expressly disclosed), and the claimed arrangement or combination of those limitations was also disclosed either expressly or inherently, in the same prior art reference.

33. I have been informed by counsel that anticipation cannot be established by combining references unless one reference is expressly incorporated in another reference. However, I understand that other references may be used to interpret an anticipating reference by, for example, indicating how a POSITA would have understood the anticipating reference, or to demonstrate the inherency of certain disclosures within the anticipating reference.

34. In addition, the description provided in the anticipatory prior art must enable a person of ordinary skill in the art in the field of invention to practice the invention without undue experimentation. Prior art patents are presumed to be enabled, but the holder of the asserted patent, who bears the burden of proof on this issue, may overcome this presumption by showing that the prior art is not enabled. I also understand that any showing by the patent holder regarding the lack of enablement of any prior art can be rebutted.

35. I understand that the pre-AIA patent statute applies to the patent at issue in this case because it issued prior to the enactment of the AIA. I further understand that pre-AIA 35 U.S.C. § 102 provides various ways for a reference to qualify as a prior art reference. It is my

understanding from counsel that all of the prior art I have relied upon qualifies as prior art to the patent.

B. Obviousness

36. Below is my understanding regarding the applicable legal standards related to obviousness. I have been informed of the same by counsel.

37. A patent claim is rendered obvious if the claimed invention would have been obvious to a POSITA as of the date of invention. A determination of obviousness is made after weighing the following factors: (1) level of ordinary skill in the pertinent art; (2) the scope and content of the prior art; (3) the differences between the prior art as a whole and the claim at issue; and (4) as appropriate, secondary considerations of non-obviousness.

38. In evaluating obviousness, both the prior art and the claimed invention should be viewed through the knowledge and understanding of a POSITA at the time of the invention – one should not use his or her own insight or hindsight in deciding whether a claim is obvious. A claim may be rendered obvious if a POSITA would understand the claimed invention as a predictable variation of a known reference.

39. An obviousness evaluation can be based on a single reference or a combination of several prior art references. An obviousness analysis involving two or more references generally requires a reason why a person of ordinary skill in the relevant field would have combined aspects of those references in the way the asserted patent claims do. I understand that the prior art references themselves may provide a suggestion, motivation, or reason to combine, but that other times the link may be simple common sense. An obviousness analysis can recognize that market demand, rather than scientific literature, often drives innovation, and that can be a sufficient motivation to combine references.

40. A particular combination of prior art references may be made by showing that it was obvious to try the combination. For example, common sense is a good reason for a person of ordinary skill to pursue known options when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions.

41. A proper obviousness analysis focuses on what was known or obvious to a POSITA, not just the patentee. For example, any need or known problem in the field at the time of invention can provide a reason for combining references to arrive at the claimed invention if the combination of prior art would address the same.

42. I understand that at least the following rationales may support a finding of obviousness: (1) combining prior art elements according to known methods to yield predictable results; (2) simple substitution of one known element for another to obtain predictable results; (3) use of a known technique to improve similar devices (methods, or products) in the same way; (4) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (5) "obvious to try"—choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; (6) a predictable variation of work in the same or a different field of endeavor if a person of ordinary skill would be able to implement the variation; (7) there existed a known problem for which there was an obvious solution encompassed by the patent's claims at the time of the claimed invention; (8) known work in one field may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art; and (9) some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

43. I also understand that an indication of obviousness may be that others having ordinary skill in the field independently made the claimed invention at about the same time the inventor made the invention.

C. Secondary Considerations of Non-Obviousness

44. I have been informed that one or more so-called "secondary considerations of non-obviousness" may impact the obviousness analysis if they are present. I have been informed that secondary considerations may include: (1) whether the invention proceeded in a direction contrary to accepted wisdom in the field; (2) whether there was a long felt but unresolved need in the art that was satisfied by the invention; (3) whether others had tried but failed to make the invention; (4) whether others copied the invention; (5) whether the invention achieved unexpected results; (6) whether the invention was praised by others; (7) whether others have taken licenses to use the invention; (8) whether experts or those skilled in the art at the making of the invention expressed surprise or disbelief regarding the invention; and (9) whether products incorporating the invention have achieved commercial success that is attributable to the invention.

45. I also understand that for any such secondary consideration to be relevant, there must be a connection or "nexus" between the secondary consideration and the claimed invention. For example, commercial success is relevant to obviousness only if the success of the product is attributable to the patented features of the claimed invention. If, on the other hand, commercial success is due to features of the product not claimed in the patent, due to claimed features that existed in the prior art, and/or due to advertising, promotion, salesmanship or the like, then any commercial success should not be considered an indication of non-obviousness.

46. I understand that Arendi has not identified any secondary considerations of non-obviousness for the asserted claims of the '843 patent. I understand that it is Arendi's duty, not

Google's or Motorola's, to identify any and all applicable secondary considerations of non-obviousness should Arendi wish to rely on such considerations. However, I will note that based on my review of materials in this case, and based on my knowledge of the state of the art in 1996-1998, I do *not* believe (1) that the alleged invention proceeded in a direction contrary to accepted wisdom in the field; (2) that there was a long felt but unresolved need in the art that was satisfied by the alleged invention; (3) that others had tried but failed to make the invention; (4) that others copied the invention; (5) that the invention achieved unexpected results; (6) that the invention was praised by others; (7) that others have taken non-settlement licenses to use the invention; (8) that experts or those skilled in the art at the making of the invention expressed surprise or disbelief regarding the invention; or (9) that products incorporating the invention, including Postnummer and OneButton Contact Manager, achieved commercial success at all, let alone commercial success that is attributable to the invention.

47. I reserve the right to modify or supplement my opinions regarding any secondary considerations of non-obviousness should Arendi or its experts raise such considerations in the future.

D. Ineligibility (35 U.S.C. § 101)

48. I understand that the patent statute defines four categories of inventions that are eligible for protection: processes, machines, manufactures, and compositions of matter. I also understand that courts have found ineligible a process or a system directed to an abstract idea with no practical application or which preempts substantially all practical applications. I also understand that courts require that meaningful limitations beyond conventional or routine steps or components such as general-purpose computer hardware be added to the abstract idea to avoid preempting all practical applications of the idea.

49. I understand that a claim that recites no more than an abstract idea expressed, for example, as software, logic, or a data structure, does not qualify for patent protection. I am informed that a claim that presents an abstract idea and that requires no more than a generic computer to perform generic computer functions that are well-understood, routine, and conventional activities previously known to the industry would not qualify for patent protection. I am informed that the recitation of generic all-purpose computing devices, including a processor, computer, computer system, or computer-readable medium, or generic components thereof, may be insufficient to render abstract ideas patentable, for example, where the claim does not include meaningful limitations beyond generally linking the use of an abstract idea to implementation via computers.

50. I understand that, under Arendi's operative complaints, it asserted U.S. Patent Nos. 8,306,993 ("the '993 patent"); 7,496,854 ("the '854 patent"); and 7,921,356 ("the '356 patent") against Google, and asserted the '993 and '854 patents against Motorola. On December 20, 2019, Judge Leonard Stark held a hearing on defendants' motion for judgment on the pleadings pursuant to 35 U.S.C. § 101. At the conclusion of the hearing, Judge Stark granted defendants' motion with regard to the '993, '854, and '356 patents and declared all asserted independent claims of those patents invalid. (*See* D.I. 201, C.A. 13-919.) On January 7, 2020, Arendi confirmed that it would not pursue any of the dependent claims of the '993, '854, or '356 patents, and that it only would be pursuing the '843 patent claims for purposes of this litigation. (*See* D.I. 202, C.A. 13-919.)

51. Based on the Court's Section 101 ruling, and on Arendi's confirmation that the ruling applies to all asserted claims of the '993, '356 and '854 patents, I have not included my

invalidity opinions regarding those patents in this report, though it is my opinion that all asserted claims from those patents are invalid under Sections 101, 102, 103, and/or 112.

E. Written Description and Enablement

52. I am informed and understand that 35 U.S.C. § 112 imposes certain requirements on the form of a patent's disclosure. In particular, I understand that the specification must meet the "written description," "enablement," and "best mode" requirements.

53. I understand that the "written description" requirement can be summarized as follows: A patent specification must contain a written description of the invention, and the written description must be sufficient to teach those of ordinary skill in the art that the named inventor(s) had possession of that claimed invention at the time the inventor(s) filed the patent application.

54. To demonstrate possession of the invention, I understand that the specification must permit a person with ordinary skill in the art to visualize or recognize the identity of the subject matter purportedly described. Possession may be shown in a variety of ways including description of an actual reduction to practice, or by showing that the invention was "ready for patenting," such as by the disclosure of drawings or structural chemical formulas that show that the invention was complete, or by describing distinguishing identifying characteristics sufficient to show that the applicant was in possession of the claimed invention. I also understand that although the specification need not describe the claimed subject matter verbatim to demonstrate possession of the invention, generalized language will not suffice if it does not convey the detailed identity of an invention. I further understand that to have an adequate written description, each element of the claimed invention must be actually or inherently disclosed in the specification. It is not sufficient that the missing claim element may be obvious from the specification. Also, the claimed invention as a whole may not be adequately described if the

claims require an essential or critical feature which is not adequately described in the specification and which is not conventional in the art or known to one of ordinary skill in the art.

55. I further understand that the patent specification must describe the manner and process of making and using the claimed invention, in such full, clear, concise, and exact terms as to enable any POSITA to which it pertains, or with which it is most nearly connected, to make and use the invention.

56. To be an enabling disclosure, I understand that the specification must adequately disclose to one skilled in the art how to make or carry out the claimed invention without undue experimentation. In other words, a patent satisfies the enablement requirement only if one skilled in the art, after reading the specification, could then practice the invention claimed without undue experimentation. The fact that some experimentation is necessary does not preclude enablement; what is required is that the amount of experimentation must not be unduly extensive. The omission of minor details does not cause a specification to fail to meet the enablement requirement. However, when there is no disclosure of any specific starting material or of any of the conditions under which a process can be carried out, undue experimentation is required and there is a failure to meet the enablement requirement that cannot be rectified by asserting that all the disclosure related to the process is within the skill of the art.

57. I understand that the factors that are considered when determining if undue experimentation is required include: (1) the quantity of experimentation necessary; (2) the amount of direction or guidance presented; (3) the presence or absence of working examples; (4) the nature of the invention; (5) the state of the prior art; (6) the relative skill of those in the art; (7) the predictability or unpredictability of the art; and (8) the breadth of the claims.

58. I further understand that an application must enable one of ordinary skill in the art to practice the full scope of the claimed invention. I also understand, however, that the specification does not necessarily need to describe how to make and use every possible variant of the claimed invention because the artisan's knowledge of the prior art and routine experimentation may be sufficient to fill certain gaps in the disclosure. However, while an applicant need not supplement the disclosure with every aspect of details well known in the prior art, this rule of supplementation is not a substitute for the requirement to provide a basic enabling disclosure and the inventors cannot simply rely on the knowledge of a POSITA to serve as a substitute for missing information from the specification.

F. Indefiniteness

59. I understand the Supreme Court has stated that a claim is indefinite when it fails to inform, with reasonable certainty, those skilled in the art about the scope of the invention. I also understand that the USPTO may employ a lower "plausible" indefiniteness standard. In particular, I understand that if a claim is amenable to two or more plausible claim constructions, the USPTO requires the applicant to more precisely define the metes and bounds of the claimed invention by holding the claim unpatentable for being indefinite. These two different standards would not change my opinion set out below.

G. The Person of Ordinary Skill in the Art

60. I understand that central to the process of understanding the disclosures in a patent and assessing the validity of a patent is the notion of a person of ordinary skill in the art (noted in my report, as above, as a "POSITA").

61. I understand that a POSITA is a hypothetical person who is used to analyzing the prior art without the benefit of hindsight. A POSITA is presumed to be one who thinks along the lines of conventional wisdom in the art.

62. I understand that the hypothetical person of ordinary skill is presumed to have knowledge of all references that are sufficiently related to one another and to the pertinent art, and to have knowledge of all arts reasonably pertinent to the particular problem that the claimed invention addresses.

H. Critical Date and Priority Date

63. I understand that there is certain terminology typically used to describe dates relevant to the questions of what qualifies as prior art to patents under the first-to-invent system predating the America Invents Act.

64. I understand that the "critical date" for a patent is one year prior to its effective filing date. It is my understanding that the critical date is significant because patents, systems, or documents that are available to the public prior to the critical date, if they qualify as prior art, will invalidate a claim regardless of when the inventors invented the claimed invention.

65. I further understand that the "priority date" of a patent is the date on which the application that leads to issuance of the patent is filed. I further understand that the priority date is significant because patents, systems, or documents that are available to the public less than one year prior to the priority date may invalidate the claims.

III. THE '843 PATENT

66. U.S. Patent No. 7,917,843 ("the '843 patent") is entitled "Method, System and Computer Readable Medium for Addressing Handling from a Computer Program." The '843 patent was filed on July 29, 2008, and issued on March 29, 2011, as a continuation of an application that issued as U.S. Patent No. 7,496,854 ("the '854 patent"). The '843 and '854 patents claimed priority as continuations of an application that issued as U.S. Patent No. 6,323,853 ("the '853 patent"), which was filed in the United States on November 10, 1998. The '843, '854 and '853 patents ultimately claimed priority to a Norwegian patent application filed

on September 3, 1998. Accordingly, the earliest possible priority date for the asserted '843 patent claims is September 3, 1998, and thus the earliest possible critical date for the asserted '843 patent claims is September 3, 1997.

A. The Specification of the '843 Patent

67. The technology in the '843 patent generally relates to a button and a computer program "coupled to an information management source for providing address handling within a document created by the computer program." '843 patent at 1:18-26.

68. The '843 patent describes a method, system, and computer readable medium that includes a "function item, such as a key, button, icon, or menu," which, when clicked, "initiates retrieval of name and addresses and/or other person or company related information, while the user works simultaneously in another program, e.g., a word processor." '843 patent at Abstract. The information retrieval is conducted by "a program connected to the button to search a database or file available on or through the computer, containing [the relevant information], in order to look up data corresponding to what the user types, or partly typed." '843 patent at Abstract. The retrieved information can be "displayed and possibly entered into the word processor, if such related information exists." '843 patent at Abstract.

69. The '843 patent explains how "[i]n recent years, with the advent of [word processor] programs, . . . users may require retrieval of information, such as name and address information, etc., for insertion into a document . . . created with the word processor. Typically, the information is retrieved by the user from an information management source external to the word processor, such as a database program . . . or from the word processor itself, for insertion into the document." '843 patent at 1:28-37.

70. By using a button and program that implements the actions of querying and updating a database for relevant information, the '843 patent seeks to avoid a problem described

in the '843 patent whereby "the user [must] learn how to use and have access to the database," and whereby changes to information in the database must be implemented by "the user of the word processor," or alternatively "by a database administrator." '843 patent at 1:43-49.

71. Thus, the '843 patent teaches a method, system, and computer readable medium that, via a button, can itself retrieve and display relevant information from a database, and insert that information into the word processor. '843 patent at 2:13-33. Since the '843 Other Publications section identifies more than 30 works from the field of hypertext/hypermedia, it is clear, such as from the Abstract, that the '843 patent is describing hypertext/hypermedia coupled with information retrieval and database utilization.

72. The '843 patent describes seven examples of functionalities that may be implemented by the disclosed invention: 1) Retrieving an existing address from the database; 2) adding a new contact to the database; 3) trying to retrieve an existing address, but the contact is not in the database; 4) adding a new address for an existing contact (titled as a "short version"); 5) selecting between several possible matching addresses; 6) adding a new address for an existing contact (titled as a "long version" since more database control is enabled); and 7) applying the invention to a spreadsheet. '843 patent at 5:59-8:67; Figs. 1-15.

73. These functionalities generally are contained within Figures 1 and 2, which depict flowcharts for implementing the various functionalities via, for example, user prompts and database querying and management, as depicted in more detail in Figures 3-15. Figures 1 and 2 depict as follows:

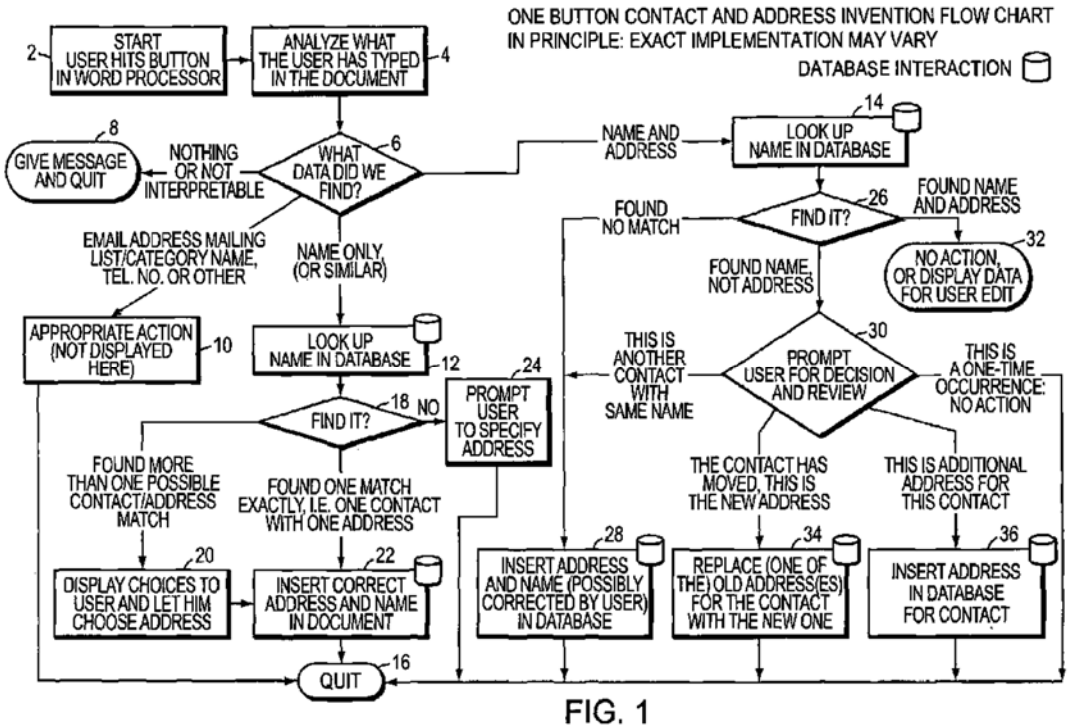


FIG. 1

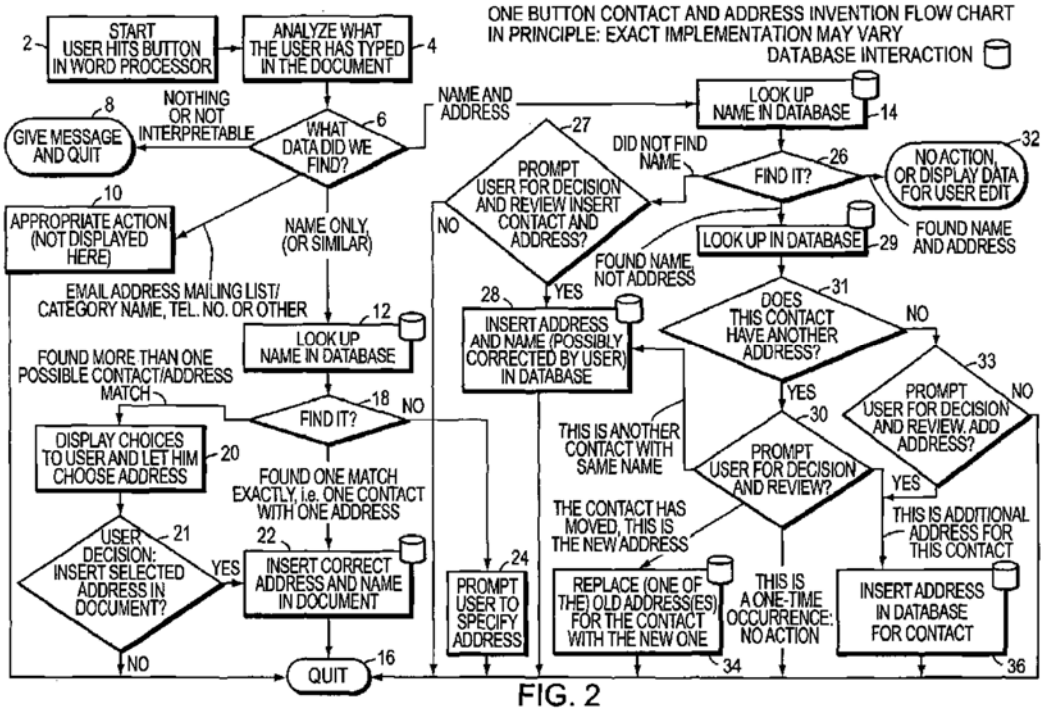


FIG. 2

74. Example 1 of the '843 patent describes retrieving an existing address from the database whereby the user types a name into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the name, searches a

database for the name, and inserts the retrieved address into the document. '843 patent at 4:25-39, 5:59-6:5, Figs. 1-4. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

75. Example 2 of the '843 patent describes adding a new contact to the database whereby the user types a new contact name and address into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact name and address, searches for the name and address in the database, informs the user that the contact is not in the database, and gives the user the option to add the new contact to the database. '843 patent at 4:25-39, 6:7-39, Figs. 1-2, Figs. 5-7.

76. Example 3 of the '843 patent describes trying to retrieve an existing address when the relevant contact is not in the database, whereby the user types a contact name into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact name and searches for the contact name in a database, but informs the user when a searched-for contact does not exist in the database via a screen shown in Figure 8, and allows the user to specify an address which can then be added as described in Example 2 of the '843 patent. '843 patent at 4:25-39, 6:40-59, Figs. 1-3, Fig. 8. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

77. Example 4 of the '843 patent describes adding a new address for an existing contact whereby the user types a name and new address of an existing contact into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact information and searches for the existing contact in a database, and informs the user of the existing contact as depicted in Figure 9, thereby prompting the user to add a new contact, update the address in the contact to the new address, use the existing address in

the document, or add the new address to the contact. '843 patent at 4:25-39, 6:61-7:23, Figs. 1-2, Fig. 4, Fig. 9. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

78. Example 5 of the '843 patent describes selecting between several possible matching addresses, whereby the user types a name and possibly an address of an existing contact into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact information and searches for the existing contact in a database, and, as depicted in Figure 10, informs the user that the contact name corresponds to several existing addresses and possible contacts in the database, and prompts the user to choose to use a particular address in the document, change the information in the database, or add an address to the database. '843 patent at 4:25-39, 7:25-8:5, Figs. 1-3, Figs. 10-11. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

79. Example 6 of the '843 patent describes adding a new address for an existing contact whereby the user types a name and new address of an existing contact into a document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact information and searches for the existing contact in a database, and informs the user that the contact already exists in the database, and allows the user to choose another address, enter a database program, add the new address to the database, or update the contact address with new address information. '843 patent at 4:25-39, 8:7-49, Figs. 1-2, Fig. 4, Figs. 9-11. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

80. Example 7 of the '843 patent describes applying Examples 1-6 to a spreadsheet document, whereby the user types a name into a spreadsheet document and presses a "OneButton" button to execute a program, which then analyzes what the user has typed to detect the contact information and searches for the name in a database, and inserts the retrieved address into the spreadsheet document. '843 patent at 4:25-39, 8:51-67, Figs. 1-2, Figs. 14-15. The '843 patent does not disclose the details of how the analysis of user-entered information occurs.

81. The only structure disclosed by the '843 patent for implementing the claimed invention is a general-purpose computer system. The patent illustrates the computer system in Figure 16 as follows:

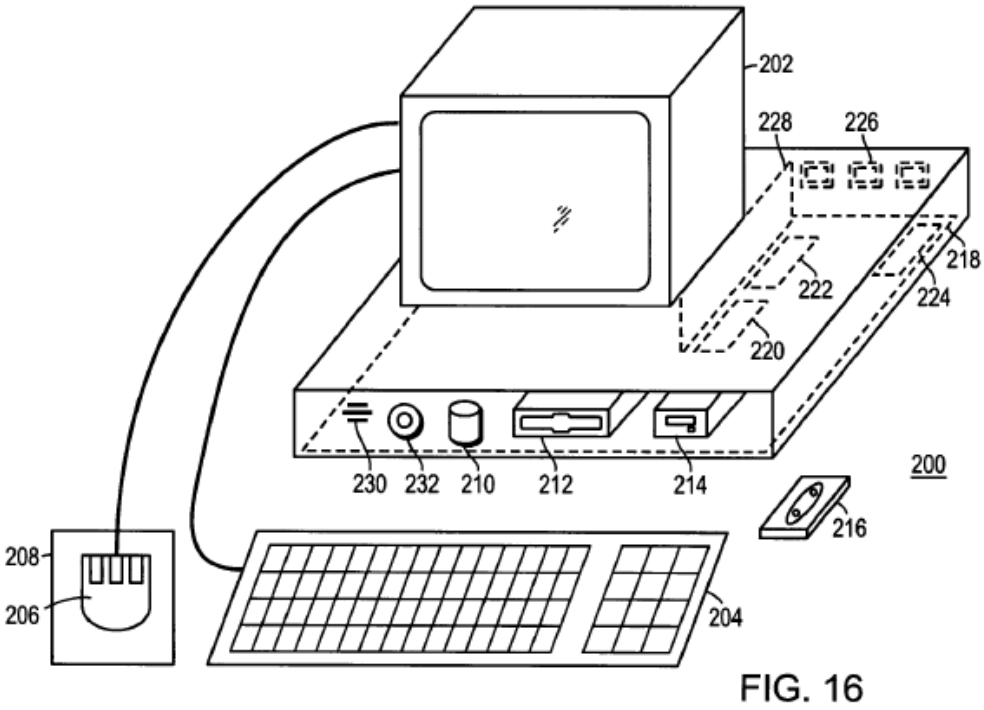


FIG. 16

With respect to Figure 16, the specification states:

> FIG. 16 is a schematic illustration of a computer system for implementing the single button addressing according to the present invention. A computer 200 implements the method of the present invention, wherein the computer includes, for example, a display device 202, such as a conventional display device or a touch screen monitor with a touch screen interface, etc., a keyboard 204, a pointing device 206, a mouse pad or

> digitizing pad 208, a hard disk 210, or other fixed, high density media drives, connected using an appropriate device bus (e.g., a SCSI bus, an Ultra DMA bus, a PCI bus, etc.), a floppy drive 212, a tape or CD ROM drive 214 with tape or CD media 216, or other removable media devices, such as magneto-optical media, etc., and a mother board 218. The mother board 218 includes, for example, a processor 220, a RAM 222, and ROM 224 (e.g., DRAM, ROM, EPROM, EEPROM, SRAM, SDRAM, and Flash RAM, etc.), I/O ports 226 which may be used to couple to external devices, networks, etc., (not shown), and optional special purpose logic devices (e.g., ASICs) or configurable logic devices (e.g., GAL and re-programmable FPGA) 228 for performing specialized hardware/software functions, such as sound processing, image processing, signal processing, neural network processing, object character recognition (OCR) processing, etc., a microphone 230, and a speaker or speakers 232.

'843 patent at 9:1-24.

B. The Asserted Claims of the '843 Patent

82. I understand that Arendi asserts that Google infringes claims 1, 8, 13, 15, 17, 18, 19, 23, and 30 of the '843 patent, and that Motorola infringes claims 1, 8, 23, and 30 of the '843 patent.

83. I further understand that Arendi specifically accuses the following Google devices as infringing the asserted claims: Nexus One, Nexus S, Galaxy Nexus, Nexus 4, Nexus 5, Nexus 6, Nexus 5X, Nexus 6P, Nexus 7 (1st generation), Nexus 7 (2nd generation), Nexus 9, Nexus 10, Pixel, Pixel XL, Pixel 2, Pixel 2 XL, Pixel 3, Pixel 3 XL, Pixel C, Pixel Slate, Chromebook Pixel, Chromebook Pixel (new edition), and Pixelbook (hereinafter "Accused Devices"; Feb. 6, 2019 Am. Suppl. Para. 4(a) Identification of Accused Products). I understand that Arendi also accuses the following Google products as infringing the asserted claims: Gmail, Gmail application, Google Docs, Google Docs application, Google Slides, Google Slides application, Google Sheets application, Google Chrome application, Google Calendar application, Google Hangouts application, Google Contacts application, Google Messages application, Inbox by

Gmail application, Google Keep application, and Google Tasks application (hereinafter "Accused Products"; *see id.*)

84. I also further understand that Arendi specifically accuses numerous Motorola products and devices as infringing the asserted claims. These products and devices are shown in Plaintiff's Amended Disclosure of Supplemental Accused Products and Asserted Patents of Motorola.

85. Claims 1 and 23 of the '843 patent are independent claims. Claims 8, 13, 15, 17, 18, and 19 depend from claim 1, and claim 30 depends from claim 23.

C. The Prosecution of the '843 patent

86. The application that led to the '843 patent (App. No. 12/182,048, "the '048 application") was filed with the U.S. Patent and Trademark Office ("USPTO") on July 29, 2008. (*See* '843 patent.) The application included 20 claims. Claims 1 and 18 were independent claims.[2]

87. On October 28, 2010, the USPTO issued its first non-final rejection of claims 1-20 based on the obviousness-type double patenting, and concluded that these claims were unpatentable over claims 1-79 of U.S. Patent No. 6,323,853 (the parent patent of the '843 patent). (*See* '843 patent file history, at 10/28/2010 '048 Appl. Non-Final Rejection, p. 4.) In its notice of rejection the USPTO noted that although the claims of the '048 application and the claims of the '853 patent were not identical, they both utilized the same method "for providing an input device configured to enter an execute command which analyzing [sic] the document to determine if the first information is contained therein, and if the first information is contained in

[2] The '993 patent claimed priority to application No. 09/390,303, which was filed on September 3, 1999, and which is now U.S. Patent No. 7,272,604. (*Id.*) There are material differences between the specification disclosures in the '993 Patent and those in the '843 Patent. I may rely on these differences in disclosure, if necessary or appropriate.

the document, ***searching***, using the record retrieval program, the information source for second information associated with the first information; then ***performing an operation using the second information***." (10/28/2010, '048 Appl. Non-Final Rejection, p. 6 (emphasis in original).) In addition, the USPTO rejected claims 1-20 under 35 U.S.C. § 103(a) as being unpatentable in view of U.S. Patent No. 6,085,201A ("Tso"), in view of the publication, "Special Edition Using Microsoft Word 97" by Que Publisher, December 16, 1996, pp. 475-514 ("Person"). (*Id.* at p. 8.) The examiner concluded that because Tso enables users to generate a text string that is responsive to the content of an input text string and a parser that decomposes the input text string into a set of search words/keywords that are used to retrieve templates corresponding to the context of the input text string, it taught almost all of the limitations of independent claims 1 and 18 except for "if searching finds any second information related to the first information, performing an operation using the second information." (*Id.* at p. 9.) This limitation is taught by Person and enables users to manually/automatically merge a mailing list with a main document to fill in blanks. (*Id.*) The examiner also concluded that it would have been obvious to combine Tso and Person in order to meet the limitations of dependent claims 2-17, 19, and 20. The inventor, Atle Hedloy submitted an amendment and a request for reconsideration on December 8, 2010. Mr. Hedloy amended representative claim 1 in the following manner:

> 1. (Currently Amended) A computer-implemented method for finding data related to the contents of a document using a <u>first</u> computer program running on a computer, the method comprising:
>
> <u>displaying the document electronically using the first computer program;</u>
>
> <u>while the document is being displayed</u>, analyzing, in a computer process, ~~the document to identify any~~ first information <u>from the document to determine if the first information is at least one of a plurality of types of information</u> that can be searched for in order to find second information related to the first information;
>
> retrieving the first information ~~thus identified~~;
>
> <u>providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the</u>

> operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated in an information source with the search term, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;
>
> in consequence of ~~following~~ receipt by the first computer program of a the user command from the input device, causing a search ~~searching~~ for the ~~first information~~ search term in the information source, using a second computer program, in order to find second information related to the search term ~~first information~~; and
>
> if searching finds and second information related to the search term ~~first information~~ performing the action an ~~operation~~ using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.

('843 patent file history, at 12/8/2010 Amend. pp. 7-8.) Mr. Hedloy also amended dependent claims 3-10, 12-17, and 19-20, amended independent claim 18, and submitted two new claims – 21 and 22 – that covered the insertion of second information into a document:

> Claim 21: (New) A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.
>
> Claim 22: (New) A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.

(*Id.* at pp. 8-12.) The applicant argued that the amendments were not made "to overcome the cited prior art, but instead to provide more context and clarity to the claims." In regards to the cited prior art, the applicant stated that because Tso decomposes and compares every word in an input text string, irrespective of the type of information that is within the text string, it does not meet the limitation that first information from the document is analyzed "to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information." (*Id.* at p. 17.) The applicant also attempted to distinguish the amended claims from the Person prior art reference and the Miller

prior art reference (U.S. Patent No. 5,946,647)—Miller was not used in the rejection of claims 1-20, but had been used in rejecting claims from a related application. According to the applicant, although Miller discloses a method for detecting data in a document and performing a particular action on the detected data, applicant argued that Miller does not disclose "performing a search . . . in order to find . . . second information, of a specific types or types, associated in an information source" as no search allegedly is performed in Miller. (*Id.* at pp. 18-19.) Finally, the applicant also filed a terminal disclaimer to overcome the non-statutory double patenting rejection (*id.* at p. 14.)

88. On December 21, 2010, the applicant submitted a supplemental response to the October 28, 2010 Office Action, adding two additional new claims (23-44), which were computer readable medium claims that mirrored claims 1-22. ('843 file history, at 12/21/2010 Supp. Resp. p. 1.)

89. On January 19, 2011, the examiner allowed claims 1-44. To do so, the examiner amended claims 1 and 23 in order to expressly state what previously had been implicit—that the search is performed "in an information source that is external to the document," (rather than merely "in an information source")—as well as other examiner-initiated amendments to various claims in order to improve readability. ('843 file history, at 1/19/2011 Examiner Interview Summary Record, p. 2; *id.* at 1/19/2011 Notice of Allowance.) The patent issued on March 29, 2011. *See* '843 patent.

90. Notably, the '853 patent, which the patent examiner relied on for its obviousness-type double patenting rejection of the '843 claims, was declared invalid by the PTAB in separate IPR proceedings. I understand that this invalidity finding with respect to the '853 patent was

affirmed by the Federal Circuit Court of Appeals. The rulings regarding the invalidity of the '853 patent claims are instructive regarding the '843 patent claims.

D. Inter Partes Review

91. On December 2, 2013, Apple Inc., Google Inc., and Motorola Mobility LLC filed a petition for inter partes review challenging all of the claims (1-44) of the '843 patent. Claims 1-44 were challenged pursuant to 35 U.S.C. §103(a) based on 1) two publications from the April 1998 issues of SIGCHI Bulletin: "From Documents to Object: An Overview of LiveDoc" and "Drop Zones: An Extension of LiveDoc" ("LiveDoc/DropZones"), and 2) U.S. Patent 5,946,647 ("Miller").

92. Additionally, claims 1-7, 10-29, and 32-44 were challenged pursuant to 35 U.S.C. §103(a) based on U.S. Patent 5,644,735 ("Luciw"), and claims 1, 2, 8, 14-17, 20, 21, 23, 24, 30, 36-39, 42, and 43 were challenged pursuant to 35 U.S.C. § 103(a) based on U.S. Patent No. 5,859,636 ("Pandit"). On June 11, 2014, the PTAB instituted inter partes review for claims 1, 2, 8, 14-17, 20, 21, 23, 24, 30, 36-39, 42, and 43 of the '843 patent only in light of Pandit, and in its institution decision explained,

> Pandit[3] teaches that, from pulled down-menu 20 (Fig. 1f), programs that can be called may include a writeable computer database of telephone and telefax numbers. Ex. 1009, col. 3, ll 1-3. Dynamically linked libraries may contain subroutines for implementing the invention with respect to telephone and telefax numbers. *Id.* at col. 4, ll. 20-31. It would be reasonable to presume, as a matter of common sense, that the subroutine would search for duplicate telephone numbers and, upon locating a duplicate entry, both the first information and associated (or second) information, such as the name and/or address associated with the telephone number, would be displayed to the user. A person having a bound paper address book would look first to determine if a potential new contact had been entered previously. A computerized search for duplicate entries would be a search 'in

[3] Pandit was filed on December 27, 1995, and teaches recognizing different classes of text in a document and providing suggestions based on it.

> order to find the second information, of a specific type or types,' as claimed, in the same sense that the '843 patent's search is in order to find the second information. As shown, for example, in Figure 1 of the '843 patent, a name (first information) can be searched for in a database (12), and more than one possible contact or address (containing second information) may be found to match with the first information (18). The first and the second information are displayed to the user for user action (20). Searching a database for a telephone number in Pandit's system, and displaying results, would be no different in substance from searching a database for a name, and displaying results, in the disclosed example in the '843 patent.

(June 11, 2014, Decision, Institution of Inter Partes Review of '843 Patent, pp. 17-18.) The PTAB declined to institute IPR of challenged claims 3-7, 9-13, 18, 19, 22, 25-29, 31-35, 40, 41, and 44 of the '843 patent. (*See id.*)

93. On June 9, 2015, the PTAB found '843 Patent claims 1, 2, 8, 14-17, 20, 21, 23, 24, 30, 36-39, 42, and 43 unpatentable based on Pandit, together with consideration of common sense. (*See* Final Written Decision.) When applying the teachings of Pandit to the challenged claims, Petitioner submitted that Pandit disclosed each limitation of claim 1 except for "performing a search as specified in step (i) of the claim," which is "performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information." ('843 Patent, col. 10, ll. 52-59.) The PTAB agreed and found the claim obvious:

> We find it reasonable to presume, as a matter of common sense and at the time of the invention, that the subroutine in Pandit would search for duplicate telephone numbers and, upon locating a duplicate entry, both the first information and associated (or second) information, such as the name and/or address associated with the telephone number, would be displayed to the user. A person having a bound paper address book would look first to determine if a potential new contact had been entered previously.

> A computerized search for duplicate entries would be a search "in order to find the second information, of a specific type or types," as claimed, in the same sense that the '843 patent's search is in order to find the second information. . . . . Searching a database for a telephone number in Pandit's system, and displaying results, would be no different in substance from searching a database for a name, and displaying results, in the disclosed example in the '843 patent. "What matters is the objective reach of the claim. If the claim extends to what is obvious, it is invalid under § 103." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 419 (2007).

(Final Written Decision, pp. 10.)

94. Petitioner had argued that the missing limitation was obvious, and stated that in order to avoid multiple duplicate address entries, it would have been obvious "that the first step in adding to an address book is to search the address book to determine if an entry already exists with the entered information, and displaying any associated information that is located." In considering Petitioner's argument, the PTAB explained that an obviousness inquiry "not only permits, but *requires*, consideration of common knowledge and common sense." (Final Written Decision, p. 12, quoting *DyStar Textilfarben GmbH & Co. v. C.H. Patrick Co*., 464 F.3d 1356, 1367 (Fed. Cir. 2006)).

95. Arendi appealed the PTAB's conclusion that claims 1, 2, 8, 14-17, 20, 21, 23, 24, 30, 36-39, 42, and 43 were unpatentable. Then, on August 10, 2016, the Federal Circuit reversed the PTAB's decision based on a finding that the PTAB "misapplied our law on the permissible use of common sense in an obviousness analysis." (Aug. 10, 2016 Appellate Decision, p. 2.) According to the Federal Circuit, the Petitioners failed to support their argument before the PTAB with substantial, concrete evidence that a search for duplicate phone numbers would result from application of common sense. (*Id.* at p. 20.) The Federal Circuit held that common sense usually cannot be invoked to supply a missing claim limitation, but only to provide a motivation to combine prior art elements. (*Id.* at 10.) According to the Court, in the event that common

sense was used to fill-in a missing limitation, the limitation was unusually simple and the technology at issue was straightforward. (*Id.*) The Federal Circuit determined that such a scenario did not apply with regard to the missing limitation of the '843 patent, which it considered to play an important role in the subject matter being claimed. (*Id.* at 11.) Finally, the Federal Circuit noted that common sense could not be used as "a wholesale substitute" to fill a missing limitation without proper evidentiary support. Specifically, the Federal Circuit reasoned that "a search for a phone number would not reveal that entering the number and the name would create a duplicate name entry where the number is brand new, but the contact name already existed," which was unlike a search based on a contact name (*Id.* at 18.) The Federal Circuit's holding focused only on the use of common sense to fill-in the missing limitation—a search for duplicative information—and did not disturb the PTAB's conclusion that Pandit disclosed all of the remaining elements of the petitioned claims of the '843 patent.

96. On November 8, 2016, Google and Motorola petitioned the Supreme Court of the United States for a writ of certiorari to review the judgment of the Federal Circuit, and presented the question, "[d]id the Federal Circuit err in restricting the Board's ability to rely on the common sense and common knowledge of skilled artisans to establish the obviousness of patent claims?" (Pet. Writ of Certiorari, (I)). This petition was denied on March 20, 2017. (137 S. Ct. 1329-30 (2017).)

E. Claim Construction

97. I understand that when the district court litigation resumed after conclusion of the IPR proceedings, the Court was asked to, and did, provide constructions for certain terms of the asserted claims of the '843 patent. I have studied these constructions and applied them in my report. I present here a table of the Court's constructions for convenience.

Claim Term	Court's Construction
document	a word processing, spreadsheet, or similar file into which text can be entered
first information	text in a document that can be used as input for a search operation in a source external to the document
computer program	a self-contained set of instructions, as opposed to a routine or library, intended to be executed on a computer so as to perform some task
to determine if the first information is at least one of a plurality of types of information that can be searched for	to determine if the first information belongs to one or more of several predefined categories of identifying information (e.g., a name) or contact information (e.g., a phone number, a fax number, or an email address) that can be searched for in an information source external to the document
that allows a user to enter a user command to initiate an operation	that allows a user to enter an input or series of inputs to initiate an operation
providing an input device configured by the first computer program	providing an input device set up by the first computer program for use by the user

IV. ACCELERATED EXAMINATION SUPPORT DOCUMENTS

98. Arendi (I will refer to Arendi as the Applicant though Atle Hedloy was the named Applicant) filed Petitions and Supporting Documents for Accelerated Examination at least in support of (i) the application for the '356 patent (Application No. 12/841,302) (*See* AHL0164696-164845); (ii) the application for U.S. Patent No. 9,201,950 (*See* AHL0195028-195107); and (iii) Application No. 13/111,639 (Pub. No. 2011/0214052), which is now-abandoned (*See* ARENDI 142811-970; ARENDI_G180880-901). All of these applications share the same specification as the '843 patent and claim priority to the same parent applications as the '843 patent. All of these applications also included numerous claim elements substantially

similar to those in the asserted '843 patent claims. As described below, Arendi made various admissions in its Accelerated Examination Support Documents that substantiate the invalidity of the asserted '843 patent claims. Arendi did not submit a similar Petition for Accelerated Examination in connection with the application for the '843 Patent.

99. In the Petition for Accelerated Examination Support Document that Arendi filed during the prosecution of the '356 patent, Arendi described and mapped several pieces of prior art that also are disclosed below in Section VIII: (i) Pandit; (ii) Miller; (iii) Tso; (iv) U.S. Patent No. 6,377,965 ("Hachamovitch"); (v) Apple Data Detectors User's Manual; (vi) "Collaborative Programmable Intelligent Agents" by Nardi and Miller; (vii) "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services" by Dey, Abowd, and Wood; and (viii) "Getting Results with Microsoft Office 97." (Pet. Accl. Examination Support Doc, '356 Patent, at 10, AHL0164696-845 at 0164705.) In its Accelerated Examination Support Documents, Arendi made a number of admissions regarding (a) the status of specific publications as prior art, (b) whether and where specific claim elements were disclosed in particular prior art references, (c) the claim elements allegedly absent from the specific prior art publications, and (d) the important differences between the claimed invention(s) and the specific prior art publications. Notably, in its Accelerated Examination Support Documents, Arendi did not disclose or discuss any prior art products or systems, nor did Arendi discuss any potential combinations of prior art references and/or prior art features that might have been obvious to a person of skill in the art at the time of invention. (*See generally id.*) Given that many of the asserted '843 patent claim elements are substantially similar to elements discussed and mapped by Arendi in the Accelerated Examination Support Documents, I may rely on any and all of Arendi's admissions to support my conclusions regarding the invalidity of the asserted '843 patent claims.

100. In the Petition for Accelerated Examination filed during the prosecution of U.S. Patent No. 9,201,950, Arendi also made several admissions in an effort to distinguish specific prior art. In the Petition, Arendi cited and described several prior art references that it "deemed most closely related to the subject matter of the claim," including the following references that also are identified below in Section VIII: Tso; "From Documents to Objects, An Overview of LiveDoc" by Miller and Bonura ("LiveDoc"); and "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services," in Knowledge-Based Systems, Vol. 11. (*See* Ex. Pet. Accl. Examination Support Doc, '950 Patent, at 7, AHL0195028-107 at 0195034.) In its Accelerated Examination Support Documents, Arendi made a number of admissions regarding (a) the status of specific publications as prior art, (b) whether and where specific claim elements were disclosed in particular prior art references, (c) the claim elements allegedly absent from the specific prior art publications, and (d) the important differences between the claimed invention(s) and the specific prior art publications. Notably, in its Accelerated Examination Support Documents, Arendi did not disclose or discuss any prior art products or systems, nor did Arendi discuss any potential combinations of prior art references and/or prior art features that might have been obvious to a person of skill in the art at the time of invention. (*See generally id.*) Given that many of the asserted '843 patent claim elements are substantially similar to elements discussed and mapped by Arendi in these Accelerated Examination Support Documents, I may rely on any and all of Arendi's admissions to support my conclusions regarding the invalidity of the asserted '843 patent claims.

101. In the Petition for Support of Accelerated Examination that was filed during the prosecution of Application No. 13/111,639, which was an application (now abandoned) that was a continuation of the '356, '853, and '854 patents, Arendi cited the following references as "most

closely related to the subject matter of the claim," which also are referenced below in the Prior Art Section VIII: (i) U.S. Patent No. 5,392,386 ("Chalas"); (ii) Pandit; (iii) Miller; (iv) Tso; (v) U.S. Patent No. 6,377,965 ("Hachamovitch"); (vi) Apple Data Detectors User's Manual; (vii) "Collaborative Programmable Intelligent Agents" by Nardi and Miller; (viii) "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services" by Dey, Abowd, and Wood; (iv) and "Getting Results with Microsoft Office 97." (*See* Ex. Pet. Accl. Examination Support Doc, '052 Patent Pub., at 10, ARENDI 142811-970 at 142820.) In its Accelerated Examination Support Documents, Arendi made a number of admissions regarding (a) the status of specific prior art publications as prior art, (b) whether and where specific claim elements were disclosed in particular prior art references, (c) the claim elements allegedly absent from the specific prior art publications, and (d) the important differences between the claimed invention(s) and the specific prior art publications. Notably, in its Accelerated Examination Support Documents, Arendi did not disclose or discuss any prior art products or systems, nor did Arendi discuss any potential combinations of prior art references and/or prior art features that might have been obvious to a person of skill in the art at the time of invention. (*See generally id.*) Given that many of the asserted '843 patent claim elements are substantially similar to elements discussed and mapped by Arendi in these Accelerated Examination Support Documents, I may rely on any and all of Arendi's admissions to support my conclusions regarding the invalidity of the asserted '843 patent claim.

V. THE STATE OF THE ART IN 1997-1998

102. At the time of the filing of the original applications that gave rise to the '843 patent, and the previously asserted '993, '854, and '356 patents, each and every element of the claimed invention was well known.

103. The state of the art was described, in part, during the prosecution of the '843, '993, '854, and '356 patents, and in admissions in the specifications of these patents.

104. For example, the '854 patent specification contains at least the following admissions:

- "In recent years, with the advent of programs, such as word processors, spreadsheets, etc. (hereinafter called "word processors") users may require retrieval of information, such as name and address information, etc., for insertion into a document, such a letter, fax, etc., created with the word processor. Typically, the information is retrieved by the user from an information management source external to the word processor, such as a database program, contact management program, etc., or from the word processor itself, for insertion into the document. Examples of such word processors are WORD™, NOTEPAD™, EXCEL™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™, etc., and examples of such information management sources are ACCESS™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™, etc." ('854 patent, col. 1, ll. 29-43.)

As another example, the '993 patent specification contains at least the following admissions:

- "In recent years, with the advent of programs, such as word processors, spreadsheets, etc. (hereinafter called "word processors") and operating systems, such as WINDOWS™ operating system, MACINTOSH™ operating system, etc., users may require retrieval of information, such as name and address information, etc., for insertion into a document, such a letter, fax, etc., created with the word processor or for contact management at the operating system level. Typically, the information is retrieved by the user from an information management source external to the word processor, such as a database program, contact management program, etc., or from the word processor itself, for insertion into the document. Examples of such word processors are WORD™, NOTEPAD™, EXCEL™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™, etc., and examples of such information management sources are ACCESS™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™, etc." ('993 patent, col. 1, ll. 32-48.)

- *See also, e.g.,* '993 file history, 2011-09-09 Response After Final Action ("it is clear that a document can be a WordTM document . . . . It is well known that a WordTM document is configured to be stored with textual information for later retrieval. Also, Figs 3, 4, and 5, illustrating an embodiment of the invention, show a document in Microsoft WordTM

with its well-known user interface including icons for saving the document and for opening documents that have been saved.")

Additionally, the following technologies and references were part of the state of the art, and would have been known to a person of skill in the art, at the time of invention:

- Word processing and spreadsheet programs and processes, including, but not limited to, those referred to in the patent specification(s) (e.g., WORD™, NOTEPAD™, EXCEL™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™)

- Information management sources, programs, and processes, including, but not limited to, those referred to in the patent specification(s) (e.g., ACCESS™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™):

- Hyperlink and hypertext technologies, protocols, and processes, including, but not limited to, the more than 30 works cited in '843 Other Publications, and those described in the following:

    - IETF RFC 2068, "Hypertext Transfer Protocol – HTTP/1.1," January 1997

    - IETF RFC 1866, "Hypertext Markup Language – 2.0," November 1995

    - Hypertext/Hypermedia Handbook (Berk & Devlin, eds.) (1991)

    - Jakob Nielsen, Multimedia and Hypertext: The Internet and Beyond, 1995

    - Proceedings of the Hypertext Standardization Workshop, Nat'l Institute of Standards & Technology, January 16-18, 1990

- Spell check technologies and processes, including, but not limited to, those described in the following

    - Domini, U.S. Patent No. 6,085,206 (filed on June 20, 1996 and issued on July 4, 2000)

    - Eudora. *See, e.g.,* Review: Eudora Pro 3.0, by Madill, February 1997

- Web and data search techniques, programs, protocols, and processes

- Human-computer interaction, windows management, user interface, GUI, information display, prompts, user interaction

- Operating systems, processes, inter-process communication
- Apple's 1987 Knowledge Navigator Video
- Email systems, protocols, and processes, including, but not limited to, those described in the following:
    - Designing Electronic Mail Systems that People will Use, by McQuillan and Walden, May 1980
    - CLUES: Dynamic Personalized Message Filtering, by Marx and Schmandt, 1996
    - Eudora. *See* Review: Eudora Pro 3.0, by Madill, February 1997
- Database programs and processes (as referred to by the patent specification(s)), including, but not limited to, those described in the following:
    - Mail Merge as a First Programming Language, by Popyack, 1993
    - Data Access for the Masses through OLE DB, by Blakeley, 1996
    - Prospects for Active Help in OnLine Documentation, by Carey, Nonnecke, and Mitterer, 1992
- Contact management programs and processes (as referred to by the patent specification(s)), including, but not limited to, those described in the following:
    - Name Searching and Information Retrieval, by Thompson and Dozier, 1997
    - CDAM-Compound Document Access and Management, by Herzner and Hocevar, April 1991
    - Microsoft Outlook, copyrights 1995-96
- OCR and text mining, analysis and/or extraction programs and processes, including, but not limited to, those described in the following:
    - COLING 1996 Vol. 1: The 16th Int'l Conference on Computational Linguistics (Aug. 5-9, 1996)
    - Natural Language Dialogue Service, by S. Busemann et al., in ANLC '97: Proceedings of the fifth conference on Applied natural language processing, March 1997

- TextFinder: An Automatic System to Detect and Recognize Text in Images, by Wu, Manmatha, Riseman, 1997
- Document Image Analysis, by O'Gorman and Kasturi, 1997
- IBM Research Report, Extracting Names from Natural-Language Text, by Ravin and Wacholder, 1997
- Automatic Structuring of Text Files, by Salton, Buckley and Allan, 1992
- Information Extraction: Beyond Document Retrieval, by Gaizauskas and Wilks, 1998
- Hybrid Text Mining for Finding Abbreviations and Their Definitions, by Park and Byrd, 1997
- The Selection Recognition Agent: Instant Access to Relevant Information and Operations, by Pandit and Kalberg, 1997
- Apple Newton MessagePad Review, by Mossberg, 1993
- Message Understanding Conference–6: A Brief History, in COLING '96: Proceedings of the 16th Conference on Computational Linguistics, Vol. 1 at 466-471, by Grishman and Sundheim (Aug. 5-9, 1996)
- The DARPA TIPSTER project, Harman, ACM SIGIR Forum, Vol. 26, Oct. 1992
- TIPSTER Text Program

- Caller ID programs and processes, including, but not limited to, those described in the following:
  - Bellcore Caller ID Specification, Dec. 1, 1991 (available at https://www.epanorama.net/documents/telecom/cid_bellcore.html)
  - CallAudit Product Summary Information (available at https://web.archive.org/web/19961111153139/http://mtnsys.com/pages/summary.htm)
  - "The Analog Display Services Interface," by Schwartz, in IEEE Comms. Magazine, Apr. 1993
  - "Caller ID Gains, Despite Limitation," New York Times, Feb. 6, 1997.

- "Caller ID System," July 1, 1997
- "New Caller ID uses promise impressive gains," Aug. 25, 1997

VI. THE PERSON OF ORDINARY SKILL IN THE ART AS TO THE '843 PATENT

105. In the expert declaration accompanying the inter partes review petition of the '843 patent, which was a petition filed by Google, Motorola Mobility, and Apple on December 2, 2013, Dr. Daniel A. Menascé stated the following:

> 32. The claims of the '843 patent are directed to computer-implemented methods and non-transitory readable media for displaying a document electronically and analyzing, in a computer process, the document while it is being displayed. The analysis determines if a first information from the document is one of a plurality of types of information that can be searched in order to find a second information related to the first information. The first information is retrieved and an input device (which could be a graphical device) is provided and configured by the first computer program.
>
> 28. In my opinion, a person of ordinary skill in the art pertaining to the '843 patent at the relevant date discussed below would have at least a Bachelor's degree in Computer Science or Electrical and/or Computer Engineering or related discipline and approximately two years experience designing applications using databases.

(IPR2014-00208, Menascé Declaration, Ex. 1002, a pp. 12, 13-14, ARENDI 208840-9091 at 208852, 208853-54.) I agree with Dr. Menascé's opinion. Further, I believe that I am familiar with the perspectives and knowledge of a POSITA at the time of invention based on my above-described education, background, and work experience.

VII. THE INVENTOR'S PURPORTED COMMERCIAL EMBODIMENTS

A. Postnummer

106. Postnummer, one of the earliest products developed by Arendi AS, helped people correct addresses or find a zip code while they were working on a Word document. (Hedloy 30(b)(1) Dep. Oct 29, 2019 at 38:11-13.) For example, if a person was typing a partial address

into a Word document, the person "could push a button and it would correct the zip code or complete the address." (*Id.* at 38:17-21; *see also* Hedloy 30(b)(1) Dep. at 372:16-22; Hedloy 30(b)(1) Dep. Ex. 25, Arendi AS Business Plan Version 1.1 5/18/1998 at p. 13, AHL0006264-92 at 0006277 ("With Arendi Postnummer you type the part of the address that you do know, just skipping the parts that you're missing. Then you hit the *Postnummer* button inside Word (this button is supplied with the Arendi Postnummer product), and the rest of the address is automatically completed: street name, town, postal code.").) When the button was clicked, the Postnummer program "would analyze the document and identify an address or partial address and try to search for that address and find the rest of it or see if it was correct." (*Id.* at 39:14-21.) The product worked in conjunction with an address database that Arendi had compiled, and which was available as part of the product's installation. (*Id.* at 47:3-24.) If more than one street or address was located as a result of the search in the Postnummer database, the program "would provide a list of those different options that the user could select one and would be put into the document. If it was only one match then it would just fill it in the document immediately and if it couldn't find there was probably an error message saying I can't find it." (*Id.* at 49:18-25.)

107. The premise behind the invention was to help alleviate issues associated with Norway's change in their zip code system as occurred in or around 1997. (*Id.* at 43:1-7, "[m]any addresses changed, the post numbers, so there was a need for people to get the correct number or zip code. So that was at least one of the problems that was being solved.")

108. According to Arendi, Atle Hedloy, the inventor of the asserted patent-in-suit, also was "the sole inventor of Arendi's patented products and services," which included the Postnummer product. (Arendi Resp. to Def. Joint Interrogatory No. 3, 10/23/2013, p. 10.) Postnummer was "first made and programmed by Atle Hedloy on or about summer 1997, first

offered for sale on or about May 1998, and first sold on or around November 1998." (*Id*; *see also* Atle Hedloy 30(b)(1) Dep. Oct. 29, 2019 at 39:25-40:7.)

109. As recognized by Mr. Hedloy, himself, the Postnummer product was not a commercial success, nor was it well-received by customers or persons of skill in the art. (Atle Hedloy 30(b)(6) Dep. Nov. 6, 2019 at 707:8-23.)

B. OneButton Contact Manager

110. OneButton Contact Manager, an Arendi AS product that launched after Postnummer, focused on providing contact information that correctly corresponded to a specifically identified contact in a Word document. As described in Arendi's marketing documents, the OneButton Contact Manager allowed a user to "simply type the name of a person or company directly into Microsoft Word, hit the button, and the correct address is automatically inserted into the document right after the name . . . [t]o store a new name and/or address, simply type it as you would normally in Word and hit the button; the name and address data is stored directly in Microsoft Outlook™." (OneButton Marketing Documents, AHL0001774-78 at 0001778.) Based on the information provided by a user in a document, OneButton Contact Manager would "initiate a search" in Microsoft Outlook™ and "compare the result of a search to what was in the document. If there was nothing else in the document it would search basically first for the name and then if the name was found and there was an address associated with it, then it had its result." (Hedloy 30(b)(1) Dep. Oct. 29, 2019 at 88:19-25.)

111. OneButton Contact Manager differed from Postnummer in that Postnummer focused on obtaining a correct address regardless of the contact associated with the address. (Hedloy 30(b)(1) Dep. Oct. 29, 2019 at 88:4-10.)

112. Contact Manager was developed in the summer of 1998, and then first offered for sale "in or around November 1998." (Arendi Resp. to Def. Joint Interrogatory No. 3, 10/23/2013, p. 10.)

113. As recognized by Mr. Hedloy, himself, the OneButton Contact Manager product was not a commercial success, nor was it well-received by customers or persons of skill in the art. (*See* Hedloy 30(b)(6) Dep. Nov. 6, 2019 at 393:8-11, 394:9-25.)

VIII. THE PRIOR ART

114. In addition to (a) the admitted prior art described above and in the "Background of the Invention" section of the '843 patent, (b) methods, products, technologies and/or techniques that were generally known and understood in the art prior to the purported invention, and (c) the general knowledge and skill possessed by any POSITA at the time the '843 patent was filed, I considered and rely upon the following prior art references to support my opinions that the asserted '843 patent claims are invalid, as they were not new or non-obvious at the time of alleged invention.

115. I discuss the prior art references and systems in detail in this section and in the claim charts that accompany this report as Exs. D–U. I further discuss in this section and in the accompanying charts the way in which each reference discloses the various elements of the asserted claims. In Section IX, I discuss the way in which a POSITA would have combined these prior art references to arrive at the purported invention of the asserted claims. Importantly, without reprinting the entirety of each prior art reference and/or including all relevant materials relating to a prior art system or product, it would be impossible for me to describe, disclose, and include in this report and the supporting charts every place (e.g., every piece of text and/or every figure) in the identified prior art where relevant teachings, disclosures, and suggestions can be found. Accordingly, I have included illustrative and exemplary citations and quotes; my

selection of particular quotes or figures does not suggest that other portions of the prior art references, products, or systems are not equally relevant, illustrative, or exemplary.

116. As noted earlier in my report, my opinions are based upon a review of, and application of, the Court's claim constructions. In the absence of a guiding claim construction as to a particular term, I have applied my understanding of how one of ordinary skill in the art would have interpreted the term. I note however that in the discussion below and in the accompanying charts, in some instances, the claim construction I apply is relevant only in light of Arendi's infringement theories and/or Arendi's purported application of its asserted claims and the construed claim terms to Accused Devices and the Accused Products. The mere fact that I may rely on Arendi's apparent interpretation of the '843 patent claims and/or on Arendi's purported application of the Court's claim constructions to support certain invalidity opinions contained in my report does not mean that I necessarily share or adopt Arendi's view of its patent claims or Arendi's claim interpretations and applications.

117. Ultimately, each reference must be read and understood (a) in its entirety, and (b) in context, considering when and by whom it was authored and/or received. For the sake of brevity, in the charts that accompany this report, I have cited to and included only exemplary passages that disclose or indicate certain claim elements; I did not reprint the entire reference (which would have provided full textual context, but which would have made the charts unduly lengthy and very difficult to manage). I reserve the right to raise and rely on additional portions or passages of each prior art reference—portions or passages that are not specifically reprinted in the charts—as necessary to support, or provide context for, my conclusion(s) that a particular reference explicitly or inherently discloses a particular claim element to a POSITA at the time of

invention. It is my opinion that all of the elements of each of the asserted claims of the '843 patent are found in the prior art, and all asserted claims are obvious.

118. The following sections explain the basis for my opinions regarding the invalidity of the asserted claims. These sections should be considered together with the rest of my report, including my discussion of the state of the art in 1997-1998, the '843 patent, the admitted prior art, the admissions of Mr. Hedloy, and the accompanying claim charts and other exhibits to my report.

A. The CyberDesk System

119. CyberDesk was both an "infrastructure and a platform that allowed you to select information in some application, have that . . . selected information be processed in a number of different ways, and to cause a number of services to be made available to—in the user interface to somebody who's trying to get more information about the information they had selected. The information could either be completely highlighted with a mouse or if it—a whole e-mail could be considered as a selection. The whole—all the information—content in a browser could be considered the selected information." (Dey Dep. 24:7-19, Nov. 12, 2019.) The goal of the CyberDesk system was to bring together information sources in order to "eliminate the manual steps of switching between all of these different tools, and instead . . . pick[ing] the text that you want more information about or picking the contents that you want more information about and then have the system guess to some degree, but suggest things that you might want to do with that information and then populate a list of—an interface that shows you all of those suggestions, you pick what you want, and then you get the result rather than having to carry this information through multiple tools." (*Id.* at 42:13-43:13.)

120. The principal architect and creator of the CyberDesk System, Anind Dey, was deposed in this case and offered a detailed description of (a) how the CyberDesk System

operated, (b) the optional and standard features of the CyberDesk System in 1997 and 1998 (prior to the '843 priority date and critical date), (c) when and how the CyberDesk System was shown and demonstrated to persons of skill in the art in 1997 and 1998 (prior to the '843 priority date and critical date), and (d) utilities, databases, and programs that could be used with and by the CyberDesk System. I understand that while Google and Motorola attempted to obtain a working version of the CyberDesk System from Dr. Dey and from Georgia Tech (Georgia Institute of Technology), which possessed the last, known working version of the CyberDesk System, no such working version remains. Moreover, as confirmed by Dr. Dey at his deposition, while the various materials identified below describe certain features and certain aspects of the operation of the CyberDesk System, no single material accurately captures or describes the full feature-set and the full operation of the CyberDesk System, as it was demonstrated for persons of skill in the art in 1997 and 1998. (*See generally* Dey Dep.) Accordingly, my understanding of the full feature-set and full operational capability of the CyberDesk System in 1997 and 1998 (prior to the priority date and the critical date) comes from the combined disclosures of (a) the materials identified below (and identified in my list of Materials Considered), (b) code fragments and screen shots from the CyberDesk System as it existed in 1997 and 1998, (c) screen shots and descriptions of utilities and databases that were used with and by the CyberDesk System in 1997 and 1998, (d) slide presentations used to demonstrate and describe the CyberDesk System in 1997 and 1998, (e) laboratory notebooks relating to the development of aspects of the CyberDesk System, (f) conversations with one of the developers of the CyberDesk System (Mike Pinkerton, whose May 1997 thesis, as he explained, provided additional information), and (g) the full Dey deposition transcript.

121. The first working version of the CyberDesk System was created in the summer of 1996, and it was first publicly used and disclosed in the United States by Spring 1997. (*See e.g.*, Dey Dep. at 26:15-27:3; Dey, Anind, et al., "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, Technical Report," GVU Center, Georgia Institute of Technology, GIT-GVU-97-10, June 1997, ARENDI-DEFS00021071-77.) Therefore, it is my understanding that the CyberDesk System constitutes prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and 102(b). As noted above, different aspects of the system, as it existed in mid-1997, are described at least in the following materials: (i) Wood, Andrew et al., CyberDesk: Automated Integration of Desktop and Network Services, CHI 97, Atlanta GA, Mar. 22-27, 1997, ACM 0-89791-802-9/97/03; (ii) Dey, Anind et al., "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, Technical Report," GVU Center, Georgia Institute of Technology, GIT-GVU-97-10, June 1997; (iii) Dey, Anind et al., CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, UIST 97, Oct. 14-17, 1997, ACM 0-89791-881-9/97/10 ("UIST Article"); (iv) Dey, Anind, et al., "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services," in Knowledge-Based Systems, Vol. 11, Issue 1 (Sept. 30, 1998); (v) Future Computing Environment CyberDesk website (https://www.cc.gatech.edu/fce/cyberdesk/), and materials linked therein. As previously mentioned, however, none of these materials describes the full feature-set or the full operation of the CyberDesk System in mid-1997. A full understanding of the publicly demonstrated and described CyberDesk System requires consideration of all of these materials together, combined with additional materials described by me above. However, in order to highlight specific aspects of the system, I will discuss particular features that were discussed and disclosed in the referenced materials and further explained by Dr. Dey.

122. "CyberDesk: Automated Integration of Desktop and Network Services," by Andrew Wood, Anind Dey, and Gregory Abowd, was the first publication regarding the CyberDesk system. (Dey Dep. at 48:18-49:5; Dey Dep. Exs. 7 and 10.) I will refer to this publication as the "Automated Integration Technical Note." This paper was presented at the 1997 Conference on Human Factors in Computing Systems, which was held from March 22-March 27, 1997. (Dey Dep. at 51:10-25.) The presented Automated Integration Technical Note was published as part of the conference proceedings in a Human-Computer Interaction (CHI) publication in 1997. (*Id.* at 49:18-51:1.)

123. The Automated Integration Technical Note describes the first instantiation of the CyberDesk System as

> a set of personal productivity services, Java applets written by other students at Georgia Tech, and several network services, commonly used by Web surfers. The integration of these services occurs automatically based on user interaction with one of them. The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in the other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate "ActOn" window.

(Automated Integration Technical Note, p. 1, ARENDI-DEFS00022052-53 at ARENDI-DEFS00022052.) The Automated Integration Technical Note provides an example of how the CyberDesk prototype used dynamic buttons and separate "ActOn" windows:

> [I]n Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlighting "Gregory Abowd" causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c).

(*Id.*) Further background insight regarding this example was provided by Mr. Dey during his November 12, 2019 deposition. Mr. Dey explained that after the user selected text in the e-mail, the CyberDesk system "used some regular expression matching" to determine that the

highlighted phrase, "Gregory Abowd," was a name. (Dey Dep. at 55:19-25.) Mr. Dey noted that a user also could look-up the selected information using Contact Manager, which was a locally stored "phone book" of information that saved a person's name, phone number, e-mail, URL of a Web page, and address. (*Id.* at 62:1-63:1.) Mr. Dey further explained that after a user decides to search for contact information through the Switchboard Web service, the user engages the Switchboard service by clicking on the button that says, "Lookup Phone Number for Name Using Switchboard." (*Id.* at 60:13-61:5.) After clicking the button, a web browser would be launched, and a webpage with information from the Switchboard website would be displayed–an address, phone number, and name. (*Id.* at 61:1-25.) The first instantiation of the CyberDesk System described in the Automated Integration Technical Note was the basis for, and was built upon to create, the second instantiation of the CyberDesk System described below.

124. "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, Technical Report," GVU Center, Georgia Institute of Technology, GIT-GVU-97-10, June 1997, was authored by Anind Dey, Gregory Abowd, Mike Pinkerton, and Andrew Wood, in April 1997, and was placed on the CyberDesk website no later than June 1997, several months after it was submitted to UIST. (Dey Dep. at 78:20-79:1; Dey Ex. 11 ("Framework Technical Report"), ARENDI-DEFS00021071-77.) I will refer to this publication as the "Framework Technical Report."

125. The Framework Technical Report describes the second instantiation of the CyberDesk system. The difference between the first and second instantiation of CyberDesk was the ability of CyberDesk to read an entire block of text without having a user first identify specific information within the text, and the "chaining" of information. (Dey Dep. at 81:21-24.) The Framework Technical Report describes CyberDesk as "a component software framework

that relieves most of the burden of integrating services from both the designer of individual services and the end user, provides greater flexibility to the user, and automatically suggests how independent services can be integrated in interesting ways." (Framework Technical Report at 1, ARENDI-DEFS00021071-77 at ARENDI-DEFS00021071.) The Framework Technical Report provides an example of a scenario that can be realized and supported using the CyberDesk System:

> The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house and nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an e-mail saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail which has information on a new book written by his favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), saves the e-mail and heads home.

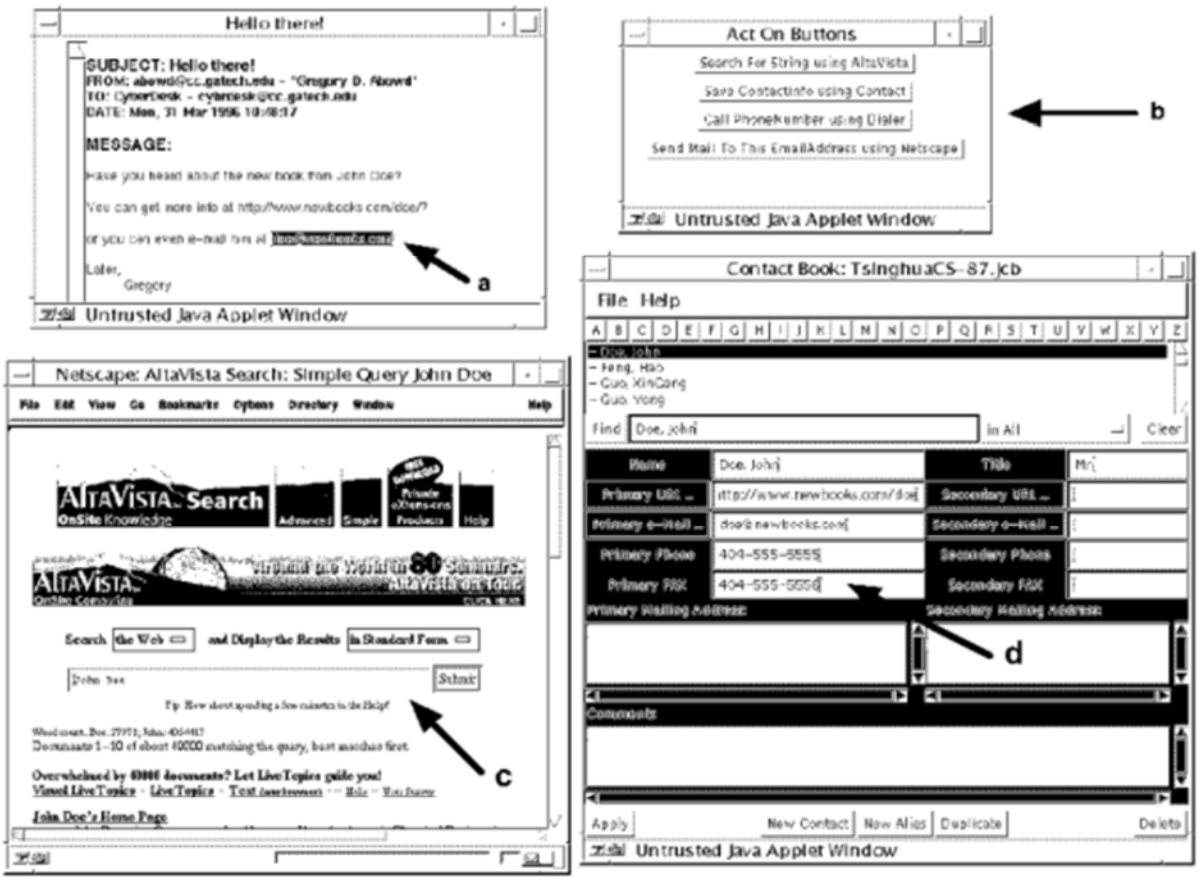


Figure 1. Mock screenshot of above user scenario

(*Id.* at 1-2, ARENDI-DEFS00021072.) The Framework Technical Report goes on to describe, in part, the architecture of the CyberDesk System, and states that the system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. (*Id.* at 2, ARENDI-DEFS00021072.) The IntelliButton component is "really the core of the CyberDesk system, as it provides the automatic integrating behaviour." (*Id.* at 3, ARENDI-DEFS00021073.) The IntelliButton identifies "potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested." (*Id.*)

126. "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services" by Anind Dey, Gregory Abowd, Mike Pinkerton, and Andrew Wood, was published in UIST in October 1997. This publication is an overview for the CyberDesk demonstration that was published in the UIST Conference proceedings. (Dey Dep. at 97:4-24, Dey Ex. 13 ("UIST Article"), ARENDI-DEFS00000778-9.) I will refer to this publication as the "UIST Article."

127. The UIST Article describes the demonstration of the CyberDesk System at UIST 1997, which walked UIST attendees through a simple demonstration of CyberDesk and described the second instantiation of the CyberDesk System that included so-called "chaining" and "combining" features. (Dey Dep. at 97:25-98:22.) The UIST Article describes the CyberDesk System as "a component software framework that automatically integrates desktop and network services, reducing integrating decisions to be made by the tool designers and giving more control to the user." (UIST Article at 1, col. 1, ARENDI-DEFS00000778.) The UIST Article also states that CyberDesk "relieves most of the burden of integrating services from both the designer of individual services and the end user. It provides greater flexibility to the user, and automatically suggests how independent services can be integrated in interesting ways. The components in CyberDesk treat all data uniformly, regardless of whether the data came from a locally running application or from a service running on the World Wide Web (WWW). The services and applications themselves can be running anywhere, meeting CyberDesk's goal of providing ubiquitous access to services." (*Id.*) The UIST Article further explained that the CyberDesk System was designed to be "easily extensible . . . [o]ne extension we've made to CyberDesk is to add services that allow it to access data from mobile devices. In particular, CyberDesk can access name and string data from an Apple Newton PDA." (*Id.* p. 2, col. 1,

ARENDI-DEFS00000779.) An example of CyberDesk being used in conjunction with an e-mail tool is explained within the publication:

> The user receives an e-mail message with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d).

(*Id.* at p. 1, col. 2, ARENDI-DEFS00000778.)

128. Parts of the CyberDesk system also were described in the publication "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services," in Knowledge-Based Systems, Vol. 11 (1998), by Anind Dey, Gregory Abowd and Andrew Wood ("Knowledge-Based System Article"). This article was accepted for publication in the journal "Knowledge-Based Systems" on June 1, 1998. (Dey Dep. at 136:7-16; *see also* Knowledge-Based System Article at p. 1, AHL0121553-63 at AHL0121553.) The version of the publication that was published in the Knowledge-Based Systems journal was an extended version of a paper that was presented at the IUI Conference in January 1998. (Dey Dep. at 137:4-19.) This article also was published on the Future Computing Environment (FCE) CyberDesk website sometime in the fall of 1997. (Dey Dep. at 129:7-130:17.) I will refer to this article as the "Knowledge-Based Systems Article."

129. The Knowledge-Based Systems Article described aspects of the second instantiation of the CyberDesk System, though like all the other articles and publications mentioned in this report, the Knowledge-Based Systems Article, by itself, does not describe or disclose all features and/or the full operation of the CyberDesk System. Nor does the Knowledge-Based Systems Article fully describe and explain the nature, extent, and content of

the public system demonstrations made by Dr. Dey in 1997 and 1998. Once again, in order to fully understand the complete feature-set and operation of the two instantiations of the CyberDesk System that were demonstrated and disclosed publicly by Dr. Dey and his colleagues in 1997 and 1998, it is critical to consider all of the various publications, Dey deposition testimony, and other materials identified previously.

130. The Knowledge-Based Systems Article, which describes the second instantiation of the CyberDesk System, explains that the system is a "framework . . . that supports the automatic integration of various software applications." (Knowledge-Based Systems Article at p. 4, Section 2, AHL0121554.) The article goes on to list several examples of the use of the system—in one example, a user highlights a URL in an e-mail message and CyberDesk offers the user the following options: 1.) search for the URL using a search engine, 2.) find pages that reference the URL using that search engine, or 3.) display the URL using a web browser. (*Id.*) The user selects the third option, and, as a result, the URL is displayed in a web browser. (*Id.* at Fig. 3, p. 4, Section 2, AHL0121554.) In another example, the user highlights a name in the e-mail message and CyberDesk offers various options that are specific to the name (e.g., look up the name in a contact manager) (*Id.* at p. 4, Section 2, AHL0121554.; *see also* Doc. In Support of Pet. Accl. Exam. Appl. No. 13/111,639, p. 106, ARENDI 142811-970 at ARENDI 142916.)

131. The Accelerated Examination Support Document filed in support of the application for the '356 patent maps the Knowledge-Based Systems Article to claim elements that are substantially similar to those included in the asserted '843 patent claims. (Doc. In Support of Pet. Accl. Exam. Appl. No. 12/841,302 ('356 patent), pp. 78-86, AHL0164696-845 at AHL0164773-81.) In this same Support Document, Applicant describes the Knowledge-Based System Article's disclosure and claims that CyberDesk did not include the following claim

elements: i.) "analyzing selected textual information *by the document editing program*," because "CyberDesk itself analyzes text highlighted by the user and CyberDesk is *separate from* any document editing programs"; ii.) "identifying at least part of the textual information to use as a search term," because "it appears that CyberDesk merely searches for what the user has highlighted and, for this reason, CyberDesk does not identify search terms, as required by the claims"; or iii.) "causing insertion of at least part of the second information into the document," because "CyberDesk does not disclose, teach, or suggest inserting second information into documents." (p. 115, AHL0164810.) Similar admissions and explanations regarding the disclosures of the Knowledge-Based Systems Article were included in other Accelerated Examination Support Documents submitted by Arendi to support various patent applications (as referenced above); though in none of its patent applications did Arendi disclose to the Patent Office other publications or materials describing different aspects and features of the CyberDesk System, and the Patent Office was not aware of the various public demonstrations and presentations of the CyberDesk System offered by Dr. Dey in 1997 and 1998. For the reasons given in this report and supporting exhibits, Arendi's claims in its Support Documents are mistaken and do not capture the full extent of CyberDesk's disclosures.

132. Importantly, starting in the mid-1990s, the Georgia Institute of Technology created and maintained a website dedicated to the "Future Computing Environment (FCE)." This website detailed various projects, products, and research being developed or pursued at Georgia Tech. The FCE website included a webpage dedicated to the CyberDesk project. The CyberDesk portion of the FCE website was created and went live in February or March 1997. (*See* Dey Dep. at 28:4-10, 31:5-13.) The first post to the CyberDesk project page on the Future Computing Environment website was made in February or March 1997 after the Automated

Integration Technical Note was published in the proceedings of the 1997 conference on Human Factors in Computing Systems. (*Id.* at 28:4-10; 31:5-13; Dey Dep. Ex. 10, ARENDI-DEFS00021078-79.)

133. The FCE CyberDesk website includes webpages hosted by the Georgia Institute of Technology that Mr. Anind Dey, one of the inventors of CyberDesk, assembled during his doctoral program (1997-1999.) (*See* Dey Dep. 16:20-25.) In addition to the Automated Integration Technical Note, other publications and materials relating to and describing aspects of the CyberDesk System were published on the FCE CyberDesk website. This includes the following materials that describe or relate to specific aspects of the second instantiation of the CyberDesk System: (a) "Context-Awareness in Wearable and Ubiquitous Computing," which was published in May 1997, posted on the FCE CyberDesk website in either May or June 1997; (b) the abbreviated poster-version of "Context-Awareness in Wearable and Ubiquitous Computing," which was presented at the Proceedings of 1st International Symposium on Wearable Computers in October, 1997, and was posted on the FCE CyberDesk website in the summer of 1997 (Dey Dep. at 110:13-20); (c) the extended version of "Context-Awareness in Wearable and Ubiquitous Computing," which was published in 1999 in the Virtual Reality Society International Journal 3, and was posted to the FCE CyberDesk website sometime in mid-1998 (*Id.* at 110:21-111:6); (d) "Applying Dynamic Integration as a Software Infrastructure for Context-Aware Computing," which was written in September 1997, submitted to the 1998 International Conference on Software Engineering (not accepted for publication), and posted on the FCE CyberDesk website in September or October 1997; (e) "CyberDesk: The Use of Perception in Context-Aware Computing," which was submitted to the Perceptual User Interfaces (PUI) Workshop, was published in October 1997 as part of the proceedings of the

1997 PUI Workshop, placed on the FCE CyberDesk website around that same time (*see* Dey Dep. at 126:19-129:2); and (f) "Context-Aware Computing: The CyberDesk Project," which described the second instantiation of the CyberDesk System, was published in the proceedings of the AAAI 1998 Spring Symposium at Stanford University, which was held from March 23-25, 1998 (*see* Dey Dep. 138:20-139:23), and posted to the CyberDesk website in December or January 1998 (Dey Dep. 143:1-6). In addition to the publications posted to the FCE CyberDesk website, the website also offered users the ability to access a stand-alone, operational version of CyberDesk starting sometime in 1998, portions of the CyberDesk System computer code, links to utilities and system components that could be used by and with the CyberDesk System (such as a contact database created by other Georgia Tech researchers), and links to additional, third party research and resources that might be of interest to users of the CyberDesk System (including, for example, a link to the Apple Data Detectors website). (Dey Dep. at 143:7-144:16.)

134. For a detailed, element-by-element analysis of how the CyberDesk System included, disclosed, and suggested the various elements of the asserted '843 patent claims, see Ex. D.

B. Apple Data Detector System

135. The Apple Data Detector ("ADD") System was created and publicly displayed at least as early as in 1996. (*See, e.g.*, Apple Data Detectors Video, ARENDI-DEFS00000001). While the various materials identified below describe certain features and certain aspects of the operation of the Apple Data Detector System, no single publication accurately captures or describes the full feature-set and the full operation of the Apple Data Detector System, as it was demonstrated for and/or used by persons of skill in the art in 1996-1998 (before the priority date and the critical date of the '843 patent). Accordingly, my understanding of the full feature-set

and full operational capability of the Apple Data Detector System in 1996-1998 comes from the combined disclosures of (a) the patent, publications, and product materials identified below (and identified in my list of Materials Considered, Ex. A), (b) videos, screenshots and descriptions of Apple Data Detector System demonstrations given at conferences in 1996 and 1997, (c) materials produced during the litigation by Apple concerning the development and operation of the Apple Data Detector System, (d) my examination of a working version of the Apple Data Detector System on a computer produced by Apple in this litigation, and (e) the full transcript of the deposition of James Miller, one of the creators of the Apple Data Detector System, and James Miller's other testimony related to the Apple Data Detector System.

136. Apple Data Detectors User Manuals and marketing materials were published and distributed by Apple in 1997. (*See* Apple Internet Address Detectors User Manual (August 28, 1997), ARENDI-DEFS00000591-606; "Apple Introduces Internet Address Detectors" (Sept. 8, 1997), ARENDI-DEFS00000623-27). These User Manuals and marketing materials generally describe certain aspects of the ADD System and its operation, though none of these publications, by itself, fully describes the operation of a computer using the ADD system in conjunction with a word processing or other computer program. In an Accelerated Examination Support Document filed to support Application No. 12/841,302, Arendi admitted that a specific 1997 ADD User Manual "discloses a method for finding (recognizing) and acting on certain types of information (p. 1). ADD starts when a user highlights text in a word processing document (p. 5, step 1). ADD then analyzes the text to determine whether it recognizes some portion of that text as, for example, an e-mail address or web address (p. 4; p.5, step 2). The user can use a contextual menu to initiate an action that is related to the recognized data (p. 5, step 2.) For example, if ADD recognizes a web address, the user may use the contextual menu to initiate viewing the

website associated with the web address (p. 5, steps 3 and 4.)." (Accelerated Pet. Appl. 12/841,302, p. 111, AHL0164806.) Similar admissions and descriptions were included by Arendi in its other Accelerated Examination Support Documents (identified above).

137. The Accelerated Examination Support Document filed in support of the application for the '356 patent maps the 1997 ADD User's Manual to claim elements that are substantially similar to those included in the asserted '843 patent claims. (Doc. In Support of Pet. Accl. Exam. Appl. No. 12/841,302, pp. 62-69, AHL0164757-64.) In this same Support Document, Applicant describes the 1997 ADD User Manual's disclosure and suggests that the described ADD did not include the following claim elements: i.) "analyzing selected textual information by the document editing program," and "providing an input device configured by the document editing program," because "ADD itself is part of the operating system and it analyzes the selected textual information and configures any input device"; ii.) "causing an electronic search in the information source . . . for the search term in order to find whether the search term is included in the information source," because "ADD simply does not disclose, suggest, or teach searching for a search term (or for second information) in an information source"; or iii.) "inserting at least part of the second information into a document." (pp. 111-12, AHL0164806-07). Importantly, in none of its patent applications did Arendi describe to the Patent Office the full range of publications and materials describing different aspects and features of the ADD System, and the Patent Office was not aware of the various public demonstrations and presentations of the ADD System offered by Apple in 1996 and 1997. Among other things, these other materials and demonstrations showed that, in fact, the ADD System did include and disclose searching for second information in an information source and inserting found

information into a document, contrary to Arendi's suggestion in its Accelerated Examination Support Documents.

138. The publication, "Collaborative, Programmable Intelligent Agents," by Bonnie A. Nardi, James R. Miller, and David J. Wright, Communications of the ACM, Vol. 41, No. 3, pp. 96–104 (ARENDI-DEFS00003329-37), ("Nardi Publication") was published in March 1998. According to the Support Document for the Accelerated Petition of Application No. 13/111,639, the Nardi Publication describes

> "the functionalities of Apple Data Detectors (ADD), which is a program that finds (recognizes) and acts on certain types of information." (p. 98-99). ADD starts when a user highlights text in a document and presses a modifier key (p. 97, Fig. 1, p. 98, "User interface" section). ADD then analyzes the highlighted text to determine whether it recognizes some portion of the text as, for example, an e-mail address or telephone number (p. 97, Fig. 1, p. 99, Fig. 4). The user can use a pop-up menu to initiate an action that is related to the recognized data (p. 97, Fig. 1.) For example, if ADD recognizes a date and time, the user may use the pop-up menu to place the date in an electronic calendar. (p. 97, Fig. 1).
>
> In another example, the Nardi reference describes a functionality wherein ADD recognizes a telephone number in the selected text and uses the telephone number to search for an associated name and address in a personal information manager program (p. 99, Fig. 4). ADD then creates a new word processor document and uses the name and address to generate word processor letterhead (p. 99, Fig. 4).

(Accl. Support Doc. Appl. No. 13/111,639, p. 105; ARENDI 142915.)

139. The Accelerated Examination Support Document filed in support of the application for the '356 patent maps the Nardi reference to claim elements that are substantially similar to those included in the asserted '843 patent claims. (Doc. In Support of Pet. Accl. Exam. Appl. No. 12/841,302, pp. 62-69, AHL0164757-64.) In this same Support Document, Applicant describes the Nardi reference's disclosure and admits that the described ADD did not include the following claim elements: i.) "analyzing selected textual information by the document editing program," because "ADD analyzes the selected textual information and ADD is separate from the document editing program"; ii.) "providing an input device configured by

the document editing program," because "it is ADD, a part of the operating system, that provides the input device"; iii.) "performing an action that 'depends at least in part on whether the search term is included in the information source,'" because "ADD carries through with an action irrespective of whether a search term is found in an information source"; and iv.) "causing insertion of at least part of the second information into the document," because "although Nardi does disclose generating word processor letterhead, the letterhead is not created in a document in which textual information is selected, as required by the claims." (pp. 113-14, AHL0164808-09). Again, however, in none of its patent applications did Arendi describe to the Patent Office the full range of publications and materials describing different aspects and features of the ADD System, and the Patent Office was not aware of the various public demonstrations and presentations of the ADD System offered by Apple in 1996 and 1997.

140. Further aspects of the ADD System are described in U.S. Patent No. 5,946,647 ("Miller"), which was filed on February 1, 1996, and issued on August 31, 1999. (ARENDI-DEFS00010281-96.) As described in Arendi's Petition for Accelerated Examination of Appl. No. 11/745,186, U.S. Patent No. 5,940,647 ("Miller") "discloses a method for detecting data in a document and performing a particular action based on the detected data (Abstract). For example, if a particular structure, such as a telephone number, is identified within the document, the telephone number within the document is mouse-sensitive (5:34-37). When a 'mouse-down' operation is recognized over the telephone number, a pop-up menu appears that enables the user to either dial the number or put the electronic number in a telephone book (Figs. 6 and 7; 5:38-50)." (Accl. Exam. of Appl. No. 11/745,186, p. 47, AHL0121116.)

141. The Accelerated Examination Support Document filed in support of the application for the '356 patent maps the Miller patent to claim elements that are substantially

similar to those included in the asserted '843 patent claims. (Doc. In Support of Pet. Accl. Exam. Appl. No. 12/841,302, pp. 27-33.) In this same Support Document, Applicant describes the Miller patent's disclosure and admits that the described ADD did not include certain claim elements. (*Id.* at pp. 103-104.) To reiterate, however, by itself, the Miller patent (like the ADD User Manual and the Nardi Publication) does not fully describe, illustrate, or suggest the full feature-set and operation of the ADD System as it was publicly demonstrated and/or used in 1996-1998.

142. For detailed, element-by-element analyses of how (1) the ADD System, and the materials describing aspects of that system, (2) Nardi, and (3) Miller disclosed and suggested the various elements of the asserted claims of the asserted patent, *see* Exs. E, F, and G.

C. LiveDoc System

143. LiveDoc Version 0.8 ("LiveDoc System") incorporates Data Detectors technology (described above) and was invented in the United States by Apple employees at least by December 5, 1995. (*See* External Requirements Specification for LiveDoc for Applications, Version 1.0a3, June 12, 1995, MILLER-APL00000025-57; "Structure Detectors/LiveDoc for Maxwell: A Starting Point, December 6, 1995," MILLER-APL00000091-95) Various aspects of the LiveDoc System are described in U.S. Patent No. 5,946,647 ("Miller"), which was filed on February 1, 1996, and in the articles "From Documents to Object: An Overview of LiveDoc" and "Drop Zones: An Extension of LiveDoc," both of which were published in the April 1998 issue of SIGCHI Bulletin. LiveDoc was demonstrated live at MacWord on or around August 7, 1996. The LiveDoc System could work in conjunction with Claris Emailer, Now Contact, and other applications.

144. While the various materials identified below describe certain features and certain aspects of the operation of the LiveDoc System, no single publication accurately captures or

describes the full feature-set and the full operation of the LiveDoc System, as it was demonstrated for and/or used by persons of skill in the art in 1996-1998 (before the priority date and the critical date of the '843 patent). Accordingly, my understanding of the full feature-set and full operational capability of the LiveDoc System in 1996-1998 comes from the combined disclosures of (a) the patent, publications, and product materials identified below (and identified in my list of Materials Considered), (b) videos, screenshots, and descriptions of LiveDoc System demonstrations given at conferences in 1996 and 1997, (c) materials produced during the litigation by Apple concerning the development and operation of the LiveDoc System, (d) my examination of a working version of the LiveDoc System on a computer produced by Apple in this litigation, and (e) the full transcript of the deposition of James Miller, one of the creators of the Live Doc System.

145. According to Arendi's own description in its Accelerated Examination Support Document for the '993 patent, the publication, "From Documents to Objects, An Overview of LiveDoc," SIGCHI Bulletin, Vol. 30 (April 1998) by James R. Miller and Thomas Bonura, describes some

> "functionalities of LiveDoc, which is a system that recognizes certain types of information in text displayed by an application program. LiveDoc starts by analyzing displayed text in an application and identifying text that represents a certain structure (e.g., telephone numbers, email addresses, and URLs) (Fig. 2.) [Fig. 2:

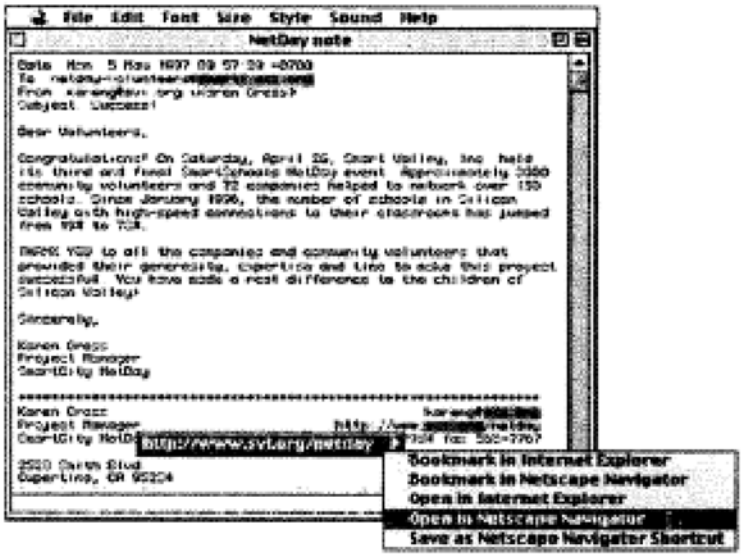


Figure 2: A sample interaction with LiveDoc. Note the highlighting of the discovered structures, the menu of actions available or the selected structure, and the nested highlighting of nested structures.

]

> When a user enters 'LiveDoc mode,' LiveDoc highlights the identified text and, if a user clicks on the text, a menu appears that allows the user to initiate an action that is related to the text (p. 4, last paragraph). For example, in Fig. 2, LiveDoc analyzes displayed text in an e-mail application and recognizes a phone number, e-mail address, and URL. LiveDoc then allows the user to open a page pointed to by the URL using Netscape Navigator (Fig. 2; p. 4, last paragraph; p. 5; p. 9, second paragraph)."

(Doc. In Support of Accl. Exam. Appl. No. 11/745,186 pp. 56-57; AHL0121125-26).

146. The LiveDoc System also was described in "Drop Zones: An Extension of LiveDoc," by Thomas Bonura and James R. Miller, April 1998, SIGCHI Bulletin, Volume 30, Number 2 ("Drop Zones"); this was published in the April 1998 issue of SIGCHI Bulletin. Drop Zones described the LiveDoc system:

> LiveDoc is an extension to the Macintosh user experience that allows documents to reveal structured information in such a way that it can be readily identified and used to achieve specific actions. Various kinds of recognizers, including context-free grammars, are used to describe the structures to be found; these structures can be made up of either a single lexical term (either a variable structure like a phone number, or a collection of static strings, like company names) or multiple terms (for instance, a meeting can be defined as a combination

> of date, time and venue structures). Small pieces of code can then be associated with each structure to instruct applications to carry out specific user actions on the discovered structures -- perhaps to tell a telephony application to 'Dial this phone number.' These actions can then be offered to users by visually highlighting the discovered structures and attaching pop-up menus to the highlights.

(Drop Zones at 1, ARENDI-DEFS00000642-46 at ARENDI-DEFS00000642.)

147. For a detailed, element-by-element analysis of how the LiveDoc System discloses and suggests the various elements of the asserted claims of the patent-in-suit, *see* Ex. H and the analysis provided in Subsection B above regarding the Apple Data Detector System, including Exs. E, F, and G.

D. Apple Newton MessagePad 2000 with Intelligent Assistant System ("Newton System")

148. The Newton handheld device was offered for sale, sold, and/or publicly used in the United States by at least March 1997. (*See* "Apple Unveils Super Newton New MessagePad More Powerful, Multifunctional," New Orleans Times Picayune (October 29, 1996); "Apple Ships $1000 MessagePad 2000," Washington Post Newsweek Interactive, (March 24, 1997); "Apple says initial MessagePad sales brisk," Reuters (April 20, 1997); "Treading Lightly in a Sea of Hand-Held Computers," New York Times on the Web: Technology | Cybertimes (Aug. 21, 1997)).

149. The Newton is a handheld computer device, which I personally used, that contains an operating system, which "is a modular set of tasks performing functions such as memory management, task management, scheduling, task to task communications, input and output, power management, and other low-level functions." (Newton Programmer's Guide: For Newton 2.0, Apple Computer, Inc. (1996) (hereinafter "Newton Guide"), at 1-1, ARENDI-DEFS00003649-4590 at ARENDI-DEFS00003697.) In order to help with mailing, faxing, printing, and calling tasks, the Newton maintains a contact database called the Name File. The

Name File stores at least three fields of information associated with the contact (i.e., name, phone number, and address). (*See* MessagePad 2000 User's Manual, Apple Computer, Inc. (1997) (hereinafter "Newton Manual") at 49 ("You can use the Name File as an address book to store information about people, companies, and groups. The Name File contains name cards that you create. Each card has information such as name, address, telephone numbers, electronic mail addresses, and notes."), ARENDI-DEFS00004995-5285 at ARENDI-DEFS00005043.)

150. An essential part of the Newton System's information architecture is the Intelligent Assistant—

> The Intelligent Assistant is a system service that attempts to complete actions for the user according to deductions it makes about the task that the user is currently performing. The Assistant is always instantly available to the user through the Assist button, yet remains nonintrusive . . . The Assistant knows how to complete several built-in tasks; they are Scheduling (adding meetings), Finding, Reminding (adding To Do items), Mailing, Faxing, Printing, Calling, and getting time information from the Time Zones map. Each of these tasks has several synonyms; for example, the user can write 'call,' 'phone,' 'ring,' or 'dial' to make a phone call.

(*See* Newton Guide at 1-8, ARENDI-DEFS00003704.) In order to invoke the Intelligent Assistant, a user taps the "Assist" button on the handheld device, which launches the Intelligent Assist feature with text from the document. For example, if a user entered the text "call bob" in the Notes application (an application that provides three different types of stationery: regular notes, checklists, and outlines) and then tapped the "Assist" button, the Intelligent Assistant pulls up the name and phone number associated with the name "bob" from the Name File, and then displays this information in the Notes application alongside the entered text. (*See* Newton Manual p. 196, ARENDI-DEFS00005190.) If a user enters a longer text string, and then invokes the Intelligent Assistant, the Assistant simply will ignore "words that do not appear in any registered template's lexicon . . . [and instead] extracts meaningful words" from the entered text. (See Newton Guide pp. 18-9, ARENDI-DEFS00004303.) For example, if a user entered the

phrase, "make a call to Bob at work," the Intelligent Assistant would extract the words "call," "Bob" and "work," in order to provide the user with additional information needed to complete a requested action. (*Id.*)

151. The Newton System uses a unified data representation so that all data stored by all applications uses a common format. Then, "[d]ata can easily be shared among different applications, with no translation necessary," which allows for seamless integration between applications. (See Newton Guide at 1-5, ARENDI-DEFS00003701.)

152. Additional aspects of the Newton System are detailed in U.S. Patent No. 5,644,735 ("Luciw"), which was filed on April 19, 1995, and issued on July 1, 1997.

153. Luciw relates to Apple's pen-based, handheld Newton device. It discloses providing user assistance based on information entered into a document such as a note areas 54a and 54b displayed by a notepad application, as shown in Fig. 2 (see below). (2:19-22, ARENDI-DEFS00009587-615 at ARENDI-DEFS00009607; 6:24-59, ARENDI-DEFS00009609.)

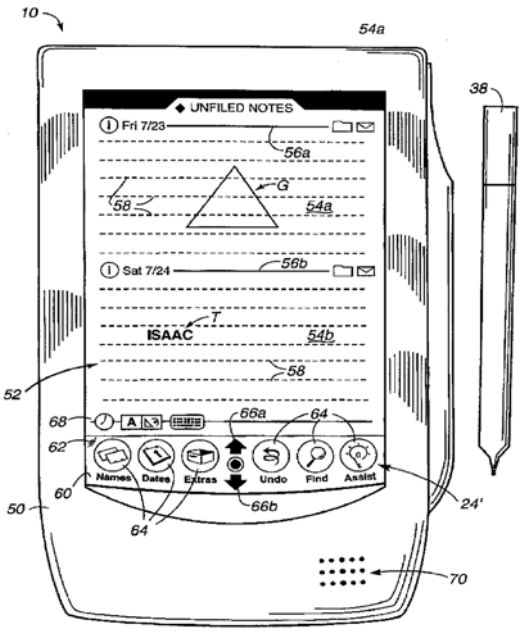


Figure 2

154. A user may select the "explicit assist" button 64 in Fig. 2, which then prompts the system to analyze the document in order to determine what type of assistance, if any is possible given the user's entry. (9:22-10:5, ARENDI-DEFS00009611; 13:52-14:4, ARENDI-DEFS00009613.) For example, if the user enters a first name, such as "Isaac," Luciw then searches a database and presents for user selection a list of names as shown in Figs. 6a and 6b (see below). (11:60-12:6, ARENDI-DEFS00009612.)

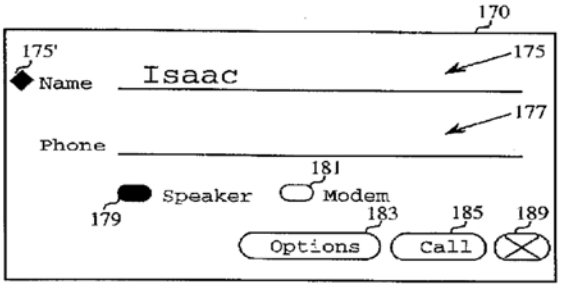


Figure 6a

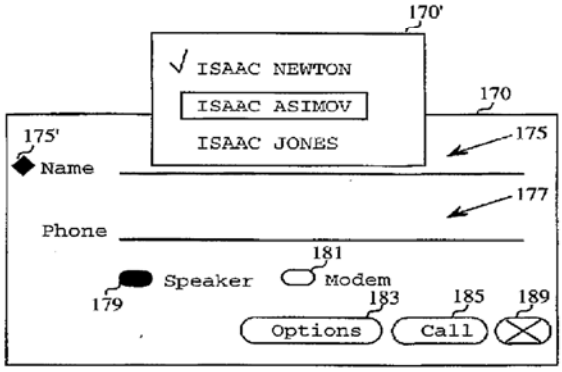


Figure 6b

155. When the user makes a selection from the list presented, information associated with the person, such as the person's full name, is inserted into the document. (*Id.*; 12:41-54.)

156. For a detailed, element-by-element analysis of how each the Newton System and Luciw discloses and suggests the various elements of the asserted claims of the patent-in-suit, *see* Exs. I and J.

E. Eudora PRO ("Eudora System")

157. Qualcomm Inc.'s Eudora PRO 3.0 was offered for sale, sold, and/or publicly used in the United States by at least February 1997. (*See* "Review: Eudora PRO 3.0" by Robert Madill (February 1997), ARENDI-DEFS00014105-11.)

158. Eudora is an e-mail management system that I personally used that was created by Qualcomm, Inc. The system has an Address Book, which stores addressee information like an e-mail address, street address, phone number, and other incidental notations. (*Id.*) A Eudora user is able to initiate an electronic communication from directly within the Address Book by using the "To, CC, and Bcc buttons." To create a new message from the Address Book, the user "select[s] the entry to which you want to address the mail . . . [t]hen click on To, CC, or Bcc. A new composition window is displayed with the selected nickname(s) inserted into the appropriate fields." (Version 3.0 for Windows User Manual – Eudora Mail Pro, June 1997, at 87, ARENDI315568-765 at ARENDI315654.) The system also has a user-configurable "toolbar" that places a series of buttons across the screen, which provides the Eudora user links to commonly used commands within the Eudora mail management system. (*See* "Review: Eudora PRO 3.0.") Another feature of Eudora is its ability to accept "embedded URLs" in the body of a message. When the user enters a URL into the body of an e-mail message, the URL automatically is colored blue and then underlined. Then, when a recipient receives the message, he/she is able to command-click the URL, which launches their preferred Web browser and automatically directs the recipient to the website. (*Id.*) Also included in the Eudora e-mail management system is a spellcheck application, "Spellswell from Working Software," which checks for grammar and spelling errors. (*Id.*)

159. For a detailed, element-by-element analysis of where Eudora discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. K.

F. Microsoft Word 97

160. Microsoft Word 1997 ("Word"), which I used, was offered for sale, sold, or publicly used in the United States at least by 1997. (*See* "Getting Results with Microsoft Office 1997," copyright 1995-1997; MS 113733-4433; *see also* "Special Edition Using Microsoft Word 97" ("Person") at 475-514, AHL0194629-67.)

161. According to the documentation filed in support of Accelerated Petition for the Examination of Application No. 13/111,639, Microsoft Word 1997 had a functionality known as "'Mail Merger.' (p. 206-214). Mail Merger allows a user to insert various types of contact information into a form letter (p. 206). The process begins when a user drafts a form letter and places merge fields, or placeholders, into the document (p. 207, 210). The user can select an address book to create a list of contact information for insertion into the form letter (p. 208). When the user selects the 'view merged data' button the contact information from the address book (names and addresses) are displayed in the form letter (p. 212)." (Support for Accl. Pet. of Appl. No. 13/111,639, p. 107; ARENDI 142917.) Person further describes the mail-merge system that was available in Microsoft Word 97, which enabled users to manually/automatically merge a mailing list with the main document to fill in blank fields.

162. For a detailed, element-by-element analysis of where each of Microsoft Word 1997 and Person discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Exs. L-M. This analysis included, among other things, my analysis of a system running Microsoft Word 97.

G. Microsoft Outlook 97

163. Microsoft Outlook 1997 ("Outlook") was offered for sale, sold, or publicly used in the United States at least by 1997. (*See* "Getting Results with Microsoft Office 1997," copyright 1995-1997, MS 113733-4433; *see also* "Special Edition Using Microsoft Outlook 97,"

by Gordon Padwick (1997); Microsoft Outlook 1997, Product Enhancements Guide, ARENDI-DEFS00000631-32.)

164. Microsoft Outlook 1997 is an e-mail management system available to Microsoft Office users. When composing e-mail messages, Outlook automatically checked the names of the addresses entered into the To, Cc, and Bcc boxes against the names in the user's Address Book. (*See* Product Enhancements Guide at 1, ARENDI-DEFS00000631.) If a contact matched an entered name, then the entered name in the To, Cc, or Bcc field is underlined. (*See id*.) On the other hand, if multiple names are found that match the user's entry, then a red line appears under the entered name, and a right click on the entered names shows other possible name matches that the user may select. (*See id.*) After a user has resolved an ambiguous contact entry, then Outlook automatically remembers and then proposes the same address-book entry the next time the same name is entered. (*See id.*)

165. For a detailed, element-by-element analysis of where Microsoft Outlook 1997 discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. N.

H. U.S. Patent No. 6,085,201 ("Tso")

166. U.S. Patent No. 6,085,201 ("Tso") was filed on June 28, 1996, and issued on July 4, 2000. (*See* Tso at 1, ARENDI-DEFS00010890-99 at ARENDI-DEFS00010890.) I understand that to mean that Tso is prior art to the asserted patent-in-suit under pre-AIA 35 U.S.C. § 103(a).

167. As described in Arendi's Petition for Accelerated Examination of Appl. No. 11/745,186, U.S. Patent No. 6,085,201 ("Tso") "discloses composing an e-mail message with the help of a context-sensitive template engine (Abstract). The method begins when a user selects a text string to be processed (4:32-35). Once a user invokes the template engine, the template

engine analyzes the text string to determine an appropriate template for the e-mail message (Abstract; 2:7-21). Tso discloses "decomposing" the text string into individual words and comparing the individual words to keywords associated with predetermined templates (5:7-17). The template with the best match between keywords and individual words is chosen as the most appropriate template and that template is presented to the user (5:7-17; 5:42-44)." (Doc. In Support of Accl. Exam. for U.S. Patent Appl. No. 11/745,186, at p. 52; AHL0121121.)

168. For a detailed, element-by-element analysis of where Tso discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. O.

I. U.S. Patent No. 6,085,206 ("Domini")

169. U.S. Patent No. 6,085,206 ("Domini") was filed on June 20, 1996, and issued on July 4, 2000. (*See* Domini at 1, ARENDI-DEFS00010900-18 at ARENDI-DEFS00010900.) I understand that to mean that Domini is prior art to the asserted patent-in-suit under pre-AIA 35 U.S.C. §§ 103(a) and 102(e).

170. Domini discloses identifying and correcting spelling errors in word processing program documents. (*See, e.g.*, *id.* at Abstract ("In an electronic word processing system environment, a system and method for verifying the accuracy of the grammatical composition of a sentence and the spelling of words within the sentence in an electronic document."), ARENDI-DEFS10900; 4:65-5:11, ARENDI-DEFS00010908-09.) In order to initialize the spell check program, the user selects the "spelling and grammar" command. (*Id.* at 16:13-16, ARENDI-DEFS00010914.) The spell check program operates without user intervention, and identifies misspelled words by changing the font of those words to a red, bold typeface. (*Id.* at 17:27-33, ARENDI-DEFS00010915; 4:12-16 ("It is determined whether any of the words in the sentence are misspelled and an indication, such as presenting the misspelled word in red, bold typeface, is provided for any misspelled words."), ARENDI-DEFS00010908.) The spell check program of

Domini displays a list of suggested corrections, which are shown in Fig. 3 (copied below), box 317. (*Id.* at 1:42-44, 10907; 12:1-5, ARENDI-DEFS00010912.) When the user selects the "Change" button, 340, the suggested correction, which is selected by the user, is inserted into the document. (*Id.* at 12:61-64, ARENDI-DEFS00010912.)

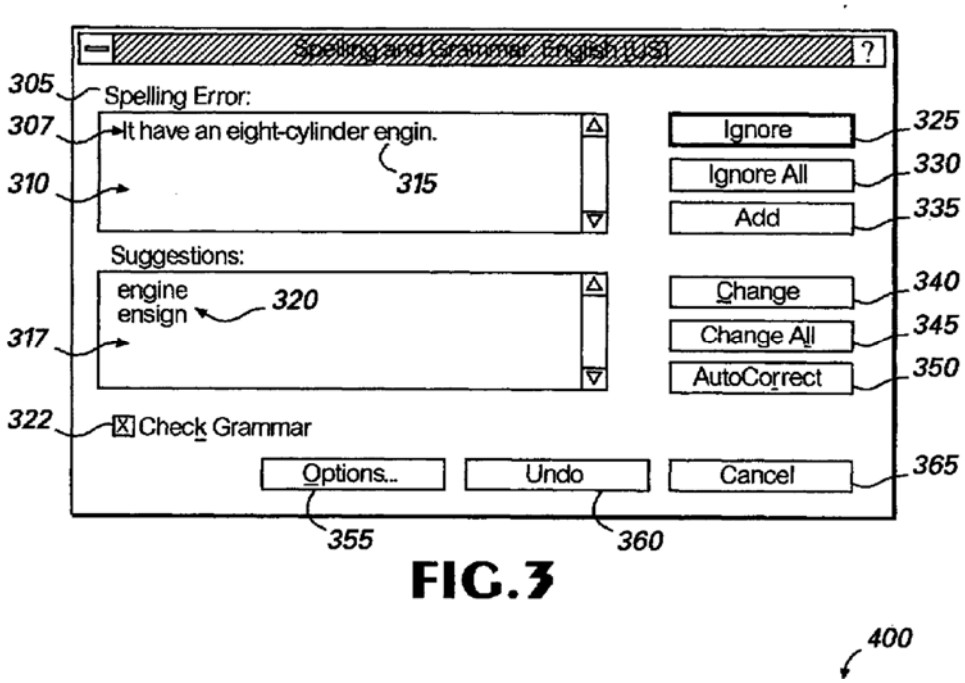


FIG. 3

171. For a detailed, element-by-element analysis of where Domini discloses and suggests the various elements of the asserted claims of the patent-in-suit, *see* Ex. P.

J. U.S. Patent No. 6,377,965 ("Hachamovitch")

172. U.S. Patent No. 6,377,965 ("Hachamovitch") was filed on November 7, 1997, and issued on April 23, 2002. (*See* Hachamovitch at 1, ARENDI-DEFS00011168-85 at ARENDI-DEFS00011168.) I understand that to mean that Hachamovitch is prior art to the asserted patent-in-suit under pre-AIA 35 U.S.C. § 103(a).

173. As described in Arendi's Petition for Accelerated Examination of Appl. No. 12/841,302, Hachamovitch "is directed to an auto-completion system (Abstract). As the user types a data entry into a document, the system searches for possible entry completions

corresponding to the partial data entry (Abstract). The system then presents the user with possible entry completions for the partial data entry (Abstract). The user can select one of the possible entry completions and the system will automatically complete his data entry (Abstract). In one particular embodiment, the system uses context to limit the number of entries presented to the user (Fig. 5, step 512; 5:38-54). For example, if the user entered information into a structured data field, then the system will only present possible completions that are relevant to that structured data field (11:14-35)." (Accelerated Examination of Appl. No. 12/841,302, p. 108, AHL0164803.)

174. For a detailed, element-by-element analysis of where Hachamovitch discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. Q.

K. U.S. Patent No. 5,392,386 ("Chalas")

175. U.S. Patent No. 5,382,386 ("Chalas") was filed on February 3, 1994, and issued on February 21, 1995. (*See* Chalas at 1, ARENDI-DEFS00009278-94 at ARENDI-DEFS00009278.) As described in Arendi's Petition for Accelerated Examination of Appl. No. 12/841,302,

> "[Chalas] discloses a system for adding functions to an existing application program (*e.g.*, Microsoft Word) (Abstract). Chalas describes clipboard that is used to transfer information between application programs (2:38-44). Chalas discloses a 'capture mechanism' that monitors communications between the operating system and an application program (5:26-30). The capture mechanism also simulates user input commands from user input devices to the operating system, and simulates commands from the operating system to the application program (5:64-68; *See also* 15:20-25). By simulating the commands and messages, Chalas is able to 1) transfer information that has been selected by the user from the application program to the clipboard, 2) modify that information, and 3) transfer the modified information back to the application program (6:1-6)." (AHL0164796, Accelerated Examination of Appl. No. 12/841,302, p. 101.) Chalas therefore "imparts add-on functionality to the document editing program (8:39-43). For example, Chalas can impart real-time spell checking (8:43-54) and translation (6:54-56). Additional functions includes detecting zip codes and providing the name of town and state (12:51-53) or detecting key phrases that designated to be replaced by a picture in the text (12:57-60)."

(Accelerated Exam. Support Doc. for Patent Appl. No. 12/841,302 at pp. 101-02, AHL0164796-97.)

176. For a detailed, element-by-element analysis of where Chalas discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. R.

L. Selection Recognition Agent/Pandit

1. Selection Recognition Agent ("SRA")

177. Intel's SRA was a computer program that was offered for sale, sold, and/or publicly used in the United States by at least March 1997. (*See* Intel News Release, "Intel Makes Exploratory Internet Technologies Available," Intel.com Pressroom (March 4, 1997), ARENDI-DEFS00019559-61.) SRA constitutes prior art under pre-AIA 35 U.S.C. §§ 102(a), 102(b), and 103(a).

178. Intel's SRA is described through the following (i) "The Selection Recognition Agent: Instant Access to Relevant Information and Operations," a publication by Pandit and Kalbag (1997) ("SRA: Instant Access"); and (ii) "The Selection Recognition Agent: Instant Access to Relevant Information and Operations," a presentation given in January 1997 at the International Conference on Intelligent User Interfaces in Orlando, Florida ("The SRA Proceedings"). The SRA is "an unobtrusive program that a user constantly runs on his PC. The SRA monitors operating system events to determine when the user has selected text in a window. It then uses fast, simple recognition processes to identify meaningful objects in the selected text. The SRA can currently recognize geographical names, dates, email addresses, Usenet newsgroup name components, world-wide web site names (URLs), and phone numbers. If the SRA recognizes one of these in the selected text, it alerts the user. The user can then use SRA to perform operations that are relevant to the recognized text object. For example, the SRA can start a web browser on a page referenced by a selected URL, or download a Usenet newsgroup's

list of Frequently Asked Questions (FAQs)." (SRA: Instant Access, p. 47, AHL0122361-66 at AHL0122361.) Additionally, the SRA "automatically turns plain text into a kind of hypertext, by quickly recognizing selected text, and then linking it to related information and applications. Thus, the SRA turns the entire desktop into a kind of hypertext document." (*Id.*)

179. For a detailed, element-by-element analysis of where SRA discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. S.

2. U.S. Patent No. 5,859,636 ("Pandit")

180. U.S. Patent No. 5,859,636 ("Pandit") was filed on December 27, 1995, and issued on January 12, 1999. (*See* Pandit at 1, ARENDI-DEFS00010090-105 at ARENDI-DEFS00010090.) I understand that to mean that Pandit is prior art to the asserted patent-in-suit under pre-AIA 35 U.S.C. § 103(a).

181. As described in Arendi's Petition for Accelerated Examination Support Document for Appl. No. 11/745,186, "Pandit discloses a method for recognizing textual data and performing a particular operation based on the recognized textual data (Abstract). For example, if accented text is recognized as an e-mail address, then a menu appears that enables the user to either send an e-mail or add the e-mail to an address book (Fig. 1d; 2:51-63)." (Accl. Exam Support Doc., p. 45; AHL0121114.)

182. For a detailed, element-by-element analysis of where Pandit discloses and suggests the various elements of the asserted claims of the patent-in-suit, see Ex. T.

M. On-site Inspection

183. On March 12, 2020, I inspected an Apple PowerBook 3400C laptop computer ("PowerBook 3400C"), an Apple PowerBook 1400CS laptop computer ("PowerBook 1400CS"), and a Newton MessagePad 2000 ("Newton Device") (collectively, "Inspected Devices") at the offices of DLA Piper, 2000 University Ave., Palo Alto, California (event collectively referred to

in the attached charts as the "Inspection" or "my Inspection"). It is my understanding that the PowerBook 3400C and PowerBook 1400CS were the same devices that were exhibits in the Miller deposition (exhibits 8 and 9, respectively). It is my understanding that the Newton Device was the same device that was Exhibit 8 in the Pagallo deposition. I spent approximately 4 hours inspecting the Newton Device, and 4 hours inspecting the PowerBook 3400C and PowerBook 1400CS. The PowerBook 3400C was running, among other things, Apple Internet Address Detectors Version 1.0.1 and LiveDoc Version 0.8. The PowerBook 1400CS was running, among other things, Apple Internet Address Detectors Version 1.0.2 and US Geographic Detectors 1.0. During my inspection of the Inspected Devices, a number of photographs were taken to document the operation of the Inspected Devices. These photographs are included and described in the attached charts for the Newton system, Apple Data Detectors system, and LiveDoc system.

IX. THE INVALIDITY OF THE ASSERTED CLAIMS OF THE '843 PATENT UNDER 35 U.S.C. §§ 102 AND/OR 103

184. It is my opinion that at least the CyberDesk System anticipates the asserted '843 Patent claims pursuant to 35 U.S.C. §§ 102(a) and 102(b). For a detailed, element-by-element analysis of how CyberDesk discloses and suggests the various elements of the asserted claims, see Ex. D.

185. For purposes of establishing obviousness under 35 U.S.C. § 103, I understand that there is no longer a rigid requirement regarding motivation to combine. In the discussion that follows, I provide various examples of combinations of prior art that I believe invalidate the asserted claims, and I generally explain the motivations for a POSITA to combine the prior art teachings.

186. It is my opinion that the following combination of references invalidate the asserted claims. Importantly, as explained in Section III.D above and as found by the PTAB, the

prior art Pandit reference discloses every element of the asserted claims of the '843 patent, with the exception of one limitation -- "performing a search using at least part of the first information as a search term in order to find the second information." ('843 Patent, col. 10, ll. 52-55, AHL0118075.) Although the Federal Circuit determined that common sense, *alone*, could not be used to satisfy this limitation, this limitation may be filled by other prior art references that a person of ordinary skill in the art would have been motivated to combine with Pandit to add the "performing a search" step. It is my opinion that at least the following combinations render the asserted claims obvious, particularly in light of the '843 patent's admissions with respect to prior art functionalities, pre-existing motivations to combine prior art functionalities, the state of the art, and common sense:

A. Pandit + CyberDesk System

As identified in both the attached CyberDesk System and Pandit charts, and in Exhibit U, additional references can also be combined with the Pandit + CyberDesk combination for certain claims and claim elements. For example, claims 15, 18 and 19 are obvious in light of Pandit + CyberDesk + Schulman; Pandit + CyberDesk + Domini; and Pandit + CyberDesk + Schabes. In fact, for each of the combinations below, additional references can also be combined with them for certain claims and elements, as identified in both their respective attached charts, and in Exhibit U which provides the bases to combine various references generally.

B. Pandit + Eudora System

C. Pandit + Apple Data Detector System

D. Pandit + LiveDoc System

E. Pandit + Newton System

F. Pandit + Microsoft Outlook 97

G. CyberDesk System + Chalas

H. CyberDesk System + Eudora System [and/or specific publications describing aspects of Eudora]

I. CyberDesk System + Apple Data Detector System [and/or specific publications describing aspects of the Apple Data Detector System]

J. **CyberDesk System + Newton System [and/or specific publications describing aspects of the Newton System]**

K. **CyberDesk System + LiveDoc System [and/or specific publications describing aspects of the LiveDoc System]**

L. **CyberDesk System + Selection Recognition Agent System [and/or specific publications describing aspects of the Selection Recognition Agent System, including Pandit]**

M. **CyberDesk System + Domini**

N. **CyberDesk System + Microsoft Word 97**

O. **Apple Data Detector System + Chalas**

P. **Apple Data Detector System + Eudora System [and/or specific publications describing aspects of the Eudora System]**

Q. **Apple Data Detector System + CyberDesk System [and/or specific publications describing aspects of the CyberDesk System]**

R. **Apple Data Detector System + Newton System [and/or specific publications describing aspects of the Newton System]**

S. **Apple Data Detector System + LiveDoc System [and/or specific publications describing aspects of the LiveDoc System]**

T. **Apple Data Detector System + Selection Recognition Agent System [and/or specific publications describing aspects of the Selection Recognition Agent System, including Pandit]**

U. **Apple Data Detector System + Domini**

V. **Apple Data Detector System + Microsoft Word 97**

W. **Apple Data Detector System + Microsoft Outlook 97**

X. **Eudora System + CyberDesk System [and/or specific publications describing aspects of the CyberDesk System]**

Y. **Eudora System + Apple Data Detector System [and/or specific publications describing aspects of the Apple Data Detector System]**

Z. **Eudora System + Newton System [and/or specific publications describing aspects of the Newton System]**

AA. **Eudora System + LiveDoc System [and/or specific publications describing aspects of the LiveDoc System]**

BB. Eudora System + Selection Recognition Agent System [and/or specific publications describing aspects of the Selection Recognition Agent System, including Pandit]

CC. Chalas + CyberDesk System

DD. Chalas + Apple Data Detector System

EE. Chalas + LiveDoc System

FF. Chalas + NewtonSystem

GG. Chalas + Selection Recognition Agent System

For a detailed, element-by-element analysis of how these combinations disclose and render obvious the various elements of the asserted claims, see Exs. D-U.

187. Admitted Prior Art: the '843 patent makes a number of admissions and representations regarding the prior art. For example:

1. Field of the Invention

This invention relates to a method, system and computer readable medium for name and address handling (hereinafter called "address handling"), and more particularly to a touch screen, keyboard button, icon, menu, voice command device, etc. (hereinafter called "button") provided in a computer program, such as word processing program, spreadsheet program, etc., and coupled to an information management source for providing address handling within a document created by the computer program.

2. Discussion of the Background

In recent years, with the advent of programs, such as word processors, spreadsheets, etc. (hereinafter called "word processors") users may require retrieval of information, such as name and address information, etc., for insertion into a document, such a letter, fax, etc., created with the word processor. Typically, the information is retrieved by the user from an information management source external to the word processor, such as a database

> program, contact management program, etc., or from the word processor itself, for insertion into the document. Examples of such word processors are WORD™, NOTEPAD™, EXCEL™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™, etc., and examples of such information management sources are ACCESS™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™, etc.

('843 patent at 1:17-42, AHL0118071.)

188. Thus, the '843 patent itself admits prior art functionalities and pre-existing motivations to combine prior art functionalities such as inserting retrieved textual information from databases into word processing documents (e.g., retrieving address information from a contact database and inserting it into a letter). As such, this section of the '843 patent is incorporated into all obviousness combinations listed above for both the admitted prior art functionalities and motivations to combine prior art functionalities. For example, this section of the '843 patent is particularly relevant to all obviousness combinations listed above with respect to claims 18 and 19 of the '843 patent.

189. To the extent that any prior art reference listed above does not teach a claim limitation either expressly or inherently, then the claim limitation is obvious to a POSITA based on the state of the art (see also Section V), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, in light of the teachings of, at least, each prior art system and/or reference listed above.

190. There are multiple reasons that a person of ordinary skill at the art at the time of invention would have been motivated to combine specific prior art references as suggested above. These reasons include, but are not limited to, the nature of the problems being addressed (e.g., attempting to make cross-referencing functionality as easy and integrated as possible,

attempting to give a user working in a text document quick access to information in a separate program/database, etc.); the express teachings and suggestions of the prior art references, themselves (individually, and as a whole); the direction of research and industry exploration at the time (e.g., the move to further extend hypertext concepts that had been studied and deployed since the late 1980s, further pushed by the emergence of the WWW, toward automating the connection of text string occurrences); common knowledge; common sense; predictability of certain programming and/or user actions; expectations of programmers and/or users; industry trends; design incentives and/or needs; market forces; and the obviousness of trying certain steps, features, and combinations. In particular, it would have been obvious to try combining the above-described prior art references because there were only a finite number of predictable solutions to the problems being addressed and/or because known work in the relevant field was encouraging variations based on predictable design incentives and/or market forces.

191. For example, to the extent that in any primary prior art reference identified above the "computer program" that is displaying a document is deemed separate or different from the computer program that is performing other claimed functionality (e.g., analysis of text to identify "first information," or configuration of the "input device" to receive a user command to utilize the identified "first information"), it would have been obvious that one design approach is to include the display and other functions as part of a single program, in accord with the teachings of Chalas, Eudora, Microsoft Office 97 (specifically use of Word macros), etc. In fact, Mr. Hedloy, himself, (a) used existing macro capabilities within Microsoft Word 97 to create his Postnummer and OneButton Contact Manager products that admittedly embodied the claimed invention, and (b) acknowledged the potential similarity of his invention to spell check functionality incorporated directly into Microsoft Word. *(See, e.g.*, Hedloy 30(b)(1) Dep. Oct.

29, 2019 at 32:4-9; Hedloy 30(b)(1) Dep. Oct. 30, 2019 at 464:13-23.) A POSITA would understand from these examples of Mr. Hedloy's invention that references to "computer program," which is construed as "a self-contained set of instructions, as opposed to a routine or library, intended to be executed on a computer so as to perform some task," assume that the underlying software of such computer programs must make use of kernel calls, interprocess communication schemes, or similar schemes familiar to those with skill in the art, to effect the coupling of programs, and the retrieval between programs, according to the varied system architectures and methods to manage programs, processes, threads, and other computer infrastructures. Further, there were in 1996-1998, and there remains now, essentially two basic approaches for how to implement intelligent agent functionality like that disclosed in CyberDesk, Apple Data Detector, LiveDoc, Selection Recognition Agent, etc.: (1) as a stand-alone program/utility that can be invoked and utilized by many applications, or (2) as a built-in feature of an individual application. As confirmed in references such as the Chalas patent that discusses alternatives and tradeoffs in the "way to add new features to an application program" (1:28-29, and more broadly in Background of the Invention), the benefits and drawbacks of these two basic options/approaches were well understood by persons of skill in the art in 1997-1998. It would have been obvious for such a POSITA interested in developing such intelligent agent functionality to explore both approaches.

192. Similarly, to the extent any primary prior art reference identified above discloses accessing information from, using information found in, or adding information into a contact database without also arguably disclosing searching the database for "first information" identified from document text, it would have been obvious to have the prior art system/method actually conduct such a search using identified "first information" as suggested by CyberDesk,

Eudora, Apple Data Detector "Write Letter" feature, LiveDoc, etc. A person of skill in the art at the time of invention would have understood that matching or relating information in documents is a fundamental objective of searching, and that incorporating a search of the database using found first information would have been wholly consistent with, and would advance, the goal of permitting a user working in a document to use identified "first information" (like a name, or a phone number, or a zip code) in the document to find and use associated "second information" (like an email address or town information) in an efficient and helpful manner. For example, since contact databases of various forms were one of the common capabilities of PDAs and PCs, and since mail merge and letter writing with address blocks were popular applications on such systems, a POSITA would find it obvious for a word processing or spreadsheet program to use a name or other identifier to search in a contact database, and then insert the results in the word processing documents, along with related support for interacting with and updating the database. Indeed, a POSITA would have understood from common sense, the teachings of various prior art (including CyberDesk, Eudora, Apple Data Detectors, LiveDoc, Outlook 97, etc.), the nature of the problem being solved, market trends and research, and understood user behaviors and expectations, that it was obvious to try using first information to perform a search to find available second information in order to determine possible steps to take and/or actions to perform with the available second information.

193. It is also important to look at the claims as a whole to understand the operations being performed by the claimed computer programs (e.g., the first and second computer programs). Little is said in the '843 specification about software infrastructure related to "programs," so the Court's construction of "computer program" guides this understanding. The

occurrences of the word "process"[4] in the '843 specification indicate a very broad interpretation. The words "first" and "second" are NEVER used in the '843 specification, only in the claims. While a prior art reference may disclose a particular operation as being performed by one "computer program" versus another "computer program," it would be understood by a POSITA that the operation could be performed by either computer program. For example, to the extent any primary prior art reference identified above discloses displaying a document using a computer program that is not the same computer program that configures an input device that allows a user to initiate an operation, it would have been obvious to have the prior art system/method display the document using either computer program, or a service of the operating system invoked by one of the computer programs. This is nothing more than a routine software design choice. For example, to the extent the CyberDesk computer program (e.g., Locator, IntelliButton, and ActOn Button Bar) in the CyberDesk system does not display the document, it would have been obvious to have the CyberDesk computer program display the document. Whether to directly incorporate the display of a document into the CyberDesk computer program or to separate that functionality into a separate computer program is nothing more than a design choice that is easily implemented either way. So too would be design choices like in the Newton System, with its specialized operating system, where computer programs "intended to be executed on a computer so as to perform some task" like the Intelligent Assistant, which "knows how to complete several built-in tasks" (Newton Guide at 1-8, ARENDI-DEFS00003704), and the "contact database" called the "Name File" (Newton Manual

[4] "Processor," "processing," and "processes" do occur; regarding the latter, see for example 9:31-45. Occurrences like "word processing," "word processor" are highly focused, as are phrases like "sound processing," "image processing," and "signal processing" (9:20-23). Occurrences like "performing the processes of the present invention" (3:28-29) relate to senses of the term referring to methods. Occurrences of the single word "processor" relate to hardware.

at 49, ARENDI-DEFS00005043), fit slightly differently in the system architecture. Indeed, a POSITA would have understood from common sense, the teachings of various prior art (including CyberDesk, Eudora, Apple Data Detectors, LiveDoc, Outlook 97, etc.), the nature of the problem being solved, market trends and research, and understood user behaviors and expectations, that it was obvious to implement a particular software operation either as part of a single computer program or as a separate computer program. For example, a POSITA may decide to implement a particular software operation as a separate computer program merely to make it easier to reuse that particular software operation in the future, either on its own or with other computer programs. But, the functionality is the same whether packaged as a single computer program or packaged as two computer programs. Thus, one should not lose the forest through the trees when assessing which "computer program" performs a particular software operation in the prior art references identified above.

194. Further, to the extent that any primary prior art reference identified above discloses performing an operation or action, but arguably does not disclose the operation or action "being of a type depending at least in part on the type or types of the first information," it would it would have been obvious to have the prior art system/method alter the types of actions offered to the user, and performed by the system/product, based on the types of first information identified, just as suggested by CyberDesk, Eudora, the Apple Data Detector "Write Letter" feature, LiveDoc, Selection Recognition Agent, etc. It also would have been obvious for a POSITA to modify the primary prior art reference to assure that an action was only taken if "searching finds any second information." A POSITA would be driven by considerations of efficiency to consider the type of information, so as to reduce the scope of analysis to be closely related to the first information, thus implementing such dependency. For example, it would

make sense to a POSITA that a primary prior art reference like Chalas (which, by Arendi's own admission in its Accelerated Examination Support Document for the '356 patent, discloses detecting a zip code, searching for the zip code in an information source external to the document, and taking action using found second information associated with the zip code) could and should be modified so that the type of actions available to a user would change if (a) there was no information relating to zip code found in the relevant database, and/or (b) the identified first information were a telephone number or an email address, rather than a zip code. A POSITA would know of basic concepts from database management and information retrieval that searching may require, and certainly will benefit from, being constrained by "field" or "column" or "metadata element" or other notion of "type of information." A POSITA would have understood from common sense, the teachings of various prior art (including CyberDesk, Eudora, Apple Data Detectors, LiveDoc, Selection Recognition Agent, etc.), the nature of the problem being solved, market trends and research, and understood user behaviors and expectations, that it was obvious to try these modifications of the identified prior art references.

195. The combinations of the prior art references identified above also would have been obvious because the combinations represent the known potential options with a reasonable expectation of success. For example, as mentioned above, and as confirmed in publications like Chalas, there were in 1996-1998, and there remains now, essentially two basic approaches for how to implement intelligent agent functionality like that disclosed in the CyberDesk, Apple Data Detector, Live Doc, Selection Recognition Agent, etc. references: (1) as a stand-alone program/utility that can be invoked and utilized by multiple applications, or (2) as a built-in feature of an individual application. The benefits and drawbacks of these two basic options/approaches were well understood by persons of skill in the art in 1996-1998; indeed,

such benefits and drawbacks are suggested in at least the Chalas patent. It would have been obvious for such a POSITA interested in developing such intelligent agent functionality to explore both of these well-known and well-understood approaches.

196. Furthermore, one of ordinary skill in the art would have been motivated to combine any of the references using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the reference or in the prior art generally. Indeed, it should be noted that the CyberDesk publications cite directly to Apple Data Detector and Selection Recognition Agent art (and teach that such art is instructive and comparable). The Miller publications similarly cite to the CyberDesk and Selection Recognition Agent art (and suggest that such art is instructive and comparable). In short, in 1996-1998, there was considerable activity around the use of text recognition and intelligent agents (including cross-referencing of art in publications, conferences, demos, and active sharing of ideas). Any POSITA at the time would have been motivated to explore and combine in different permutations the various approaches that were being taught and tested at the time.

197. Because the references above include overlapping teachings relating to the operations and advantages of particular intelligent agent implementations, the problems and issues addressed in the ‘843 patent specification, and the elements of the asserted claims, I reserve the right to present other combinations of the above-identified references at trial, as well as additional explanation of the motivations to combine particular references. Additional combinations and/or additional explanations regarding motivations to combine could become more relevant and/or necessary based on additional rulings by the Court, new claim construction

or application positions taken by Arendi, and/or opinions or positions offered by Arendi in its expert reports or in summary judgment papers or at trial.

X. THE ASSERTED CLAIMS LACK SUPPORT IN THE PATENT'S WRITTEN DESCRIPTION

198. Asserted claims 1, 8, 23, and 30 of the '843 patent (and all asserted claims dependent therefrom) are also invalid under 35 U.S.C. § 112. I have provided my understanding of the requirements of 35 U.S.C. § 112 in Section II.E.

199. As previously noted, Plaintiff's two sets of Infringement Contentions, which were served in December 2013 and February 2019, lack proper and complete disclosure as to each Defendant's accused products and functionalities. Moreover, Plaintiff's deficient and inadequate Infringement Contentions do not permit me to ascertain or understand how Plaintiff is interpreting and applying the various claim elements and/or how far Plaintiff is attempting to extend the scope of the asserted patent claims; and Plaintiff has not proposed any claim constructions to this point. Accordingly, I presently cannot fully assess the extent to which Plaintiff's interpretations and/or attempted applications of its asserted claims will raise or affect Section 112 issues. I thus reserve the right to further supplement or modify my report, including my identification of Section 112 issues.

200. As I presently understand the claim language as applied in Plaintiff's deficient Infringement Contentions, the asserted claims of the patent-in-suit fail to meet one or more of the requirements of 35 U.S.C. § 112, and are therefore invalid, for at least the following reasons:

201. Claims 1 and 23 of the '843 patent (and all asserted claims dependent therefrom) do not meet the written description and/or enablement requirements of 35 U.S.C. § 112. There is no support in the patent specification for the limitation "providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information . . . ."

This limitation requires a user command to initiate a search operation using the first information identified in the analyzing step. Thus, the searching step must occur after the analyzing step. The specification does not disclose analyzing and searching steps initiated by separate user commands.[5] The specification discloses only a single execute command that initiates both the analyzing and the searching steps. To the extent that the claims are construed to require that the user command occurs after the analyzing step, a person of ordinary skill in the art would not understand how to practice the claimed invention as applied by Plaintiff, nor would they understand the alleged inventor to have been in possession of the purported invention.

202. To the extent that Arendi contends that the '843 claims can be read to cover analysis and/or identification of non-contact information, such as flight numbers, package tracking numbers, etc., the claims would not meet the written description and/or enablement requirements of 35 U.S.C. § 112. There is no support in the patent specification for such claim scope. While the specification describes searching a database for additional information (e.g., second information) based on a "name" (e.g., a type of information), it does not describe searching for second information based on anything other than a name. A person of ordinary skill in the art would not understand how to practice the claimed invention as applied by Plaintiff, nor would they understand the alleged inventor to have been in possession of the purported invention.

203. Claims 1 and 23 of the '843 patent (and all asserted claims dependent therefrom) do not meet the written description and/or enablement requirements of 35 U.S.C. § 112. There is no support in the patent specification for the limitation "while the document is being displayed,

[5] The lack of disclosure of this in the '843 patent contrasts with the plain disclosure of it in the published prior art. *See, e.g.*, Tso at 4:48-51, 5:51-53 (describing this in the context of "template engine 5").

analyzing in a computer process . . . ." The specification discloses that the analysis of the document is initiated when a "user hits button in word processor" (*see* Figs. 1 and 2, AHL0118057-58); "OneButton" (*see* Figs. 3-5, AHL0118059-61); "in a word processor, the button is added and a user types information, such as an addressee's name, or a part of the name, etc. in a document created with the word processor, such as a letter, fax, etc., and then clicks, selects, commands, etc. the button via the appropriate input device . . . [a] program then executes and retrieves the typed information from the document . . ." (*see* 3:42-49, AHL0118072). The specification does not disclose the analysis of a document automatically—without a user hitting a "button."[6] Moreover, the specification does not disclose any structure or algorithm relating to "analyzing, in a computer process, first information from the document . . ." as required by Section 112. A person of ordinary skill in the art would not understand how to practice the claimed invention as applied by Plaintiff, nor would they understand the alleged inventor to have been in possession of the purported invention.

204. To the extent that Arendi contends that the '843 claims can be read to cover implementations and/or functionalities that utilize code, routines, or libraries that are included in, or part of, the operating system, rather than part of the code that defines a "first computer program," the claims would not meet the written description and/or enablement requirements of 35 U.S.C. § 112, as the patent specification does not adequately describe or enable the full scope of the claims as applied in such a manner. Indeed, Arendi added disclosures regarding inclusion of the described functionalities in operating system libraries/code to the specification of the '993

[6] The lack of disclosure of this in the '843 patent also contrasts with the plain disclosure of it in the published prior art. *See, e.g.*, Hachamovitch at 6:39-52, 10:18-37 (describing this in the context of an auto-complete system).

Patent (which Arendi concedes was only entitled to a later priority date than the '843 Patent) precisely because such disclosures are absent from the '843 Patent specification.

205. All asserted claims of the '843 patent, as applied by Arendi, do not meet the written description and/or enablement requirements of 35 U.S.C. § 112, as the patent specification does not adequately describe or enable the full scope of the claims as alleged in Arendi's Infringement Contentions. The '843 patent has a priority date of late 1998, and discloses the alleged software inventions being implemented using then-standard word processing documents, which were run on desktop computers with then-standard desktop operating systems and then-standard hard-wire Internet connections. (*See* Section V, above.) The complex mobile software systems that Plaintiff currently accuses of infringement are not described or enabled in the specifications (or the prosecution histories). Nor would it have been an obvious, trivial, or even a possible task for a person of ordinary skill in the art in 1998 to create the complex mobile software systems now accused.

206. Initially, the desktop computer systems disclosed by the '843 patent were not designed for mobile use over wireless cellular networks. Indeed, the '843 patent fails to disclose how an operating system would be implemented on a mobile device, such as a cellular phone. More specifically, the interoperability of multiple standalone applications in a mobile device environment is not disclosed, including opening and passing information to a second application from a first application. Nor does the '843 patent disclose how a mobile phone operating system or application would interface with other applications and databases on remote servers, including making requests to and receiving/processing/displaying data from third party applications and servers. As another example, the '843 patent fails to disclose how a mobile device operating system would recognize and process user inputs and actions such as touch gestures on a touch

screen, including how those touch gestures would interact directly with the standalone applications.

207. Moreover, open source mobile operating systems were unknown in 1998. Existing operating systems in 1998 running on hand-held devices, like PalmOS, were proprietary, and typically did not allow for cellular operation. Data synchronization on such devices was achieved by way of a physical interconnection between the hand-held device and a PC. The patent specification provides no usable guidance or examples that would allow a person of ordinary skill in the art in 1998 to practice the claimed inventions using separate mobile applications interacting with a mobile operating system and/or a remote server or database using a cellular network (or even Wi-Fi).

208. In fact, the high-speed cellular network connections available today, and which are required to operate complex software systems that Arendi currently accuses of infringement, were not available in 1998. While cellular networks existed in 1998, the available wireless bandwidth was inadequate to permit data-intensive communications between a cellular phone and servers on different networks around the world in real-time, such as in the accused software systems. The available limited cellular data networks available at the time were low speed and directed to use by specific non-commercial applications transmitting discrete data sets, such as CDPD's (Cellular Digital Packet Data) used in police vehicles to transmit license plate data information to laptops mounted in the police vehicles. Other low data rate applications included pagers and SMS text messaging, which utilized bandwidth in voice channels to transmit data.

209. Thus, the specification of the '843 patent does not disclose the software code or instructions that would enable a person of ordinary skill in the art in 1998 to implement the alleged inventions as currently interpreted by Arendi to encompass mobile devices (such as the

Apple iPhone, the Google Pixel 3, LG Rebel 4, or the Samsung S9), including, at least, how to launch a second application program, pass information between two application programs, query databases (local or external), insert information into databases (local or external), create pop-up windows or menus, parse/analyze textual information in a document, insert information from external sources into a document, or display information from external sources. Moreover, the specification provides no detail on how such complex operations can be performed on a handheld mobile device operating over a cellular connection.

XI. THE VALUE OF THE ASSERTED CLAIMS WITH RESPECT TO THE PRIOR ART

210. I have been informed by counsel that my opinions regarding the prior art as compared to the asserted claims are relevant not only to the issue of the validity of the '843 patent, but also to the analysis of the value of the patent should it ultimately be deemed valid. Counsel has advised that one way of determining the value of a reasonable royalty for a purportedly patented invention is to consider certain characteristics of the patent and the prior art, known as the Georgia-Pacific factors. Specifically, I have been informed that Georgia-Pacific factors 9, 10, and 11 are the following:

> 9. The utility and advantages of the patent property over the old modes or devices, if any, that had been used for working out similar results;
>
> 10. The nature of the patented invention; the character of the commercial embodiment of it as owned and produced by the licensor; and the benefits to those who have used the invention; and
>
> 11. The extent to which the infringer has made use of the invention; and any evidence probative of the value of that use.

211. Thus, I have been asked by counsel to be prepared, and I am prepared, to discuss at trial consistent with this report: (1) the differences between the asserted claims of the '843 patent and the prior art discussed in this report, (2) the utility and advantages (or lack thereof) of the purportedly patented technology as compared to old methods of finding data related to the

contents of a document that existed before the priority date of the '843 patent, (3) the similarity of the results obtained using the purportedly patented technology as compared to the prior art, (4) the nature of the patented invention, including the comparison of the nature of the patented invention with that of the prior art and similar methods of finding data related to the contents of a document that came before the patent; and (5) to the extent they should come to light, commercialized versions of the purported invention, and any value purporting to be related to such commercialization.

212. For example, the asserted claims of the '843 patent all require finding data related to the contents of a document using a first and second computer program. *See, e.g.,* claim 1: "A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer [and] . . . . using a second computer program." There is relatively little, if any, increased value in a software process that performs the well-known software operations of text parsing, searching, and insertion (as admitted by the '843 patent itself) as a single computer program versus spread across multiple computer programs. While there may be value to the overall development platform, e.g., operating system, in compartmentalizing certain functionality and services such that they can be utilized repeatedly and independently by individual and distinct applications, the application itself will provide the same functionality regardless of whether that functionality is built into the application or is a part of some distinct framework. Indeed, the fact that commercial organizations like Microsoft, Apple, and Intel, as well as research institutions like Georgia Tech, published their methods and released successful products, as reported above, using various software frameworks that implemented the claimed invention, makes clear the low value of the '843 specification.

213. As another example, the prior art Pandit reference discloses every element of the asserted claims of the '843 patent, with the exception of one limitation -- "performing a search using at least part of the first information as a search term in order to find the second information." As discussed in this report, multiple other prior art references disclose the basic concept of using one piece of information to search for another piece of information. The ubiquity of this simple concept in the prior art suggests its relative low value. Notably, the '843 patent presents no explanation (unique or otherwise) as to how such a search should be performed, further testifying to its low value and well known status.

214. I also again note that the patentee's purported commercial embodiment of the '843 patent, Postnummer, and OneButton Contact Manager, were admittedly not commercial successes. Thus, assuming these products were commercial embodiments of the '843 patent, the '843 patent could not have utility and advantages over the old modes or devices embodied in or described by the prior art discussed in my report.

I certify that, to the best of my knowledge and belief:

- The statements of fact in this report are true and correct;
- The reported analyses, opinions, and conclusions are limited only by the reported assumptions and are my personal, unbiased, and professional analyses, opinions, and conclusions;
- I have no personal interest or bias with respect to the parties involved; and
- My compensation is not contingent on an action or event resulting from the analysis, conclusions, or opinions of this report.

Dated: August 7, 2020

Edward A. Fox Ph.D.

Edward Fox, Ph.D.

EXHIBIT LIST

Ex. A: List of Materials Considered
Ex. B: *Curriculum Vitae* of Edward A. Fox
Ex. C: Consulting Summary List of Edward A. Fox
Ex. D: CyberDesk Invalidity Chart
Ex. E: Apple Data Detectors Invalidity Chart
Ex. F: Nardi Invalidity Chart
Ex. G: Miller 647 Invalidity Chart
Ex. H: LiveDoc Invalidity Chart
Ex. I: Newton Invalidity Chart
Ex. J: Luciw 735 Invalidity Chart
Ex. K: Eudora Invalidity Chart
Ex. L: Word 97 Invalidity Chart
Ex. M: Person Invalidity Chart
Ex. N: Outlook Invalidity Chart
Ex. O: Tso 201 Invalidity Chart
Ex. P: Domini Invalidity Chart
Ex. Q: Hachamovitch Invalidity Chart
Ex. R: Chalas Invalidity Chart
Ex. S: Selection Recognition Agent Invalidity Chart
Ex. T: Pandit 636 Invalidity Chart
Ex. U: Obviousness Combinations

Exhibit A

Patents & File Histories

U.S. Patent No. 7,921,356
U.S. Patent No. 7,921,356 File History
U.S. Patent No. 7,917,843
U.S. Patent No. 7,917,843 File History
U.S. Patent No. 7,496,854
U.S. Patent No. 7,496,854 File History
U.S. Patent No. 8,306,993
U.S. Patent No. 8,306,993 File History
U.S. Patent No. 6,323,853
U.S. Patent Pub. No. 2011/0214052 A1

IPR Documents

0452 IPR '853 Patent

2015-08-18 Final Written Decision
2018-02-20 Federal Circuit Decision

0450 IPR '356 Patent

2014-02-20 Allison Declaration ISO Petition for IPR
2014-02-20 Petition for Inter Partes Review
2014-05-23 Arendi's Preliminary Response to Petition for IPR
2014-08-20 Decision - Institution of IPR
2014-11-04 Arendi's Response to Petition for IPR
2014-11-04 Levy Declaration re Petition for IPR
2015-02-03 Google's Reply to Arendi's Response to IPR
2015-07-28 Record of Oral Hearing held April 21, 2015
2015-08-19 Final Written Decision
Exhibit 1001 - U.S. Patent No. 7,921,356
Exhibit 1003 - Dennis Allison CV
Exhibit 1004 - US5859636
Exhibit 1005 - US5644735
Exhibit 1006 - Bonura & Miller, Drop Zones, An Extension to LiveDoc
Exhibit 1007 - US6377965
Exhibit 1008 - US5923848
Exhibit 1009 - US6085201
Exhibit 1010 - Magnanelli et al., Academia - An Agent-Maintained Database Based on Information Extracted from Web Documents
Exhibit 1011 - US5754306
Exhibit 1012 - US5790532
Exhibit 1013 - Claim Language Comparison, '356 Patent
Exhibit 1014 - Amendment, Dec. 18, 2000
Exhibit 1015 - Notice of Allowability
Exhibit 1017 - Deposition of John Levy taken Jan. 8, 2015
Exhibit 2001 - Declaration of John V. Levy
Exhibit 3001 - Collins English Dictionary excerpts

Exhibit A

0208 IPR '843 Patent
2013-12-02 Petition for IPR
2014-03-12 Arendi's Preliminary Response
2014-06-11 Decision - Institution of IPR
2014-06-25 Petitioner's Request for Rehearing
2014-06-27 Decision - Request for Rehearing
2014-11-11 Petitioners' Reply to Patent Owner's Response
2015-03-03 Record of Oral Hearing held Feb. 4, 2015
2015-06-09 Final Written Decision
Exhibit 1001 - US7917843
Exhibit 1002 - Declaration of Daniel A. Menasce
Exhibit 1003 - Amendment, Jan. 24, 2008
Exhibit 1004 - Office Action, Oct. 28, 2010
Exhibit 1005 - Amendment, Dec. 8, 2010
Exhibit 1006 - Miller, an Overview of the ATG Intelligent Systems Program
Exhibit 1007 - US5946647
Exhibit 1008 - US5644735
Exhibit 1009 - US5859636
Exhibit 1010 - Miller & Bonura, From Documents to Objects - An Overview of LiveDoc
Exhibit 1013 - Deposition of Daniel A. Menasce taken Aug. 7, 2014
Exhibit 1014 - Apple, Google & Motorola's Presentation slides
Exhibit 2001 - definition of configure
Exhibit 2002 - Declaration of John V. Levy
Exhibit 2003 - Arendi's Presentation slides
2016-08-10 IPR2014-0208 Federal Circuit Opinion
2016-11-08 - Petition for Writ of Certiorari
2017-03-20 - Supreme Court denial of Petition for Writ of Certiorari

0206 IPR '854 Patent
2013-12-02 Petition for IPR
2014-03-12 Arendi's Preliminary Response
2014-06-11 Decision - Institution of IPR
2014-08-26 Arendi's Response
2014-11-11 Petitioners' Reply to Patent Owner Response
2015-03-03 Record of Oral Hearing held Feb. 4, 2015
2015-06-09 Final Written Decision
Exhibit 1001 - US7496854
Exhibit 1002 - Declaration of Daniel A. Menasce
Exhibit 1003 - Amendment, Jan. 24, 2008
Exhibit 1004 - Amendment, Apr. 18, 2007
Exhibit 1005 - Miller, an Overview of the ATG Intelligent Systems Program
Exhibit 1006 - US6085206
Exhibit 1007 - US5946647
Exhibit 1008 - US5644735

Exhibit A

Exhibit 1009 - US5963964
Exhibit 1011 - Deposition of John V. Levy taken Oct. 22, 2014
Exhibit 1012 - Deposition of Daniel A. Menasce taken Aug. 7, 2014
Exhibit 1013 - Apple, Google & Motorola's Presentation slides
Exhibit 2001 - The American Heritage College dictionary - dictionary
Exhibit 2002 - The American Heritage College Dictionary 3rd Ed – designate
Exhibit 2003 - Declaration of John V. Levy
Exhibit 2005 - Arendi's Presentation slides
2016-07-11 IPR2014-00206 & IPR2014-00207 Federal Circuit Opinion

0207 IPR '854 Patent
2013-12-02 Petition for IPR
2014-03-12 Arendi's Preliminary Response
2014-06-11 Decision - Institution of IPR
2014-08-26 Arendi's Response
2014-11-11 Petitioners' Reply to Patent Owner Response
2015-02-04 Record of Oral Hearing held Feb. 4, 2015
2015-06-09 Final Written Decision
Exhibit 1001 - US7496854
Exhibit 1002 - Declaration of Daniel A. Menasce
Exhibit 1003 - Amendment, Jan. 24, 2008
Exhibit 1004 - Amendment, Apr. 18, 2007
Exhibit 1005 - Miller, an Overview of the ATG Intelligent Systems Program
Exhibit 1006 - US5577239
Exhibit 1007 - US6085206
Exhibit 1008 - US6377965
Exhibit 1009 - US5644735
Exhibit 1010 - Miller & Bonura, From Documents to Objects - An Overview of LiveDoc
Exhibit 1013 - Deposition of John V. Levy taken Oct. 22, 2014
Exhibit 1014 - Deposition of Daniel A. Menasce taken Aug. 7, 2014
Exhibit 2001 - The American Heritage College dictionary – dictionary
Exhibit 2003 - Declaration of John V. Levy Exhibit 2004 - Arendi's Presentation slides for IPR2014-00206 & IPR2014-00207
2016.07.11 IPR2014-00206, IPR2014-00207 Federal Circuit Opinion

0203 IPR '993 Patent
2013-12-02 Petition for IPR
2014-03-12 Arendi's Preliminary Response
2014-06-05 Decision Denying Institution of IPR
2014-07-07 Petitioners' Request for Rehearing
2014-07-25 Decision on Request for Rehearing
Exhibit 1001 - US8306993
Exhibit 1002 - Declaration of Dennis R. Allison
Exhibit 1003 - US5644735
Exhibit 1004 - US6247043

Exhibit A

Exhibit 1005 - US6870828
Exhibit 1006 - Bonura & Miller, Drop Zones an Extension to LiveDoc
Exhibit 1007 - Magnanelli et al., Academia - An Agent-Maintained Database Based on Information Extraction from Web Documents
Exhibit 1008 - Dennis Allison CV
Exhibit 1009 - US5859636
Exhibit 1010 - US5644735
Exhibit 1011 - US5754306
Exhibit 1012 - US5790532
Exhibit 1013 - Claim Language Comparison
Exhibit 1014 - Technote 1005 - The Complete Guide to Simple Text
Exhibit 1015 - 14th European Mtg on Cybermetics & Systems Research
Exhibit 1016 - ETH Zurich - Computer Science - Publications
Exhibit 1017 - 14th European Meeting Cybermetics & Systems Research
Exhibit 2001 - Request for Continued Examination - Appl. No. 11-745186
Exhibit 2002 - Miller & Bonura, From Documents to Objects - An Overview of LiveDoc

Arendi Deps & Exhibits

Atle Hedloy 30(b)(6)
November 5, 2019 Transcript & Exhibits 37-77
November 6, 2019 Transcript & Exhibits 78-102
November 7, 2019 Transcript & Exhibits 103-118

Atle Hedloy 30(b)(1)
October 29, 2019 Transcript & Exhibits 1-25
October 30, 2019 Transcript & Exhibits 26-36

Third Party Deps & Exhibits

Anind Dey
Dep. transcript, Exhibits 1-32, & Video

Jim Miller
Dep. transcript, Exhibits 1-23, & Video

Pagallo, Guilia
Dep. Transcript, Exhibits 1-7, & Video

Infringement Contentions
2013.12.06 Arendi Claim Charts
2019.02.13 Arendi Motorola (12-1601) Claim Charts & Exhibits
2019.02.13 Arendi Google (13-919) Claim Charts & Exhibits

Invalidity Contentions
2019.03.27 Defendants' Joint Amended Invalidity Contentions & Exhibits

Exhibit A

Pleadings
2018.12.18 Amended Complaint
2019.01.11 Google's Answer to Arendi's Amended Complaint
2019.07.02 Defendants' Motion for Judgment on the Pleadings
2019.07.02 Opening Brief ISO Defendant's Motion for Judgment on the Pleadings
2019.08.13 Arendi's Answering Brief in Opposition to Defendants' Motion for Judgment on Pleadings - 101 Motion Dkt #137
2019.08.13 Arendi's Answering Brief in Opposition to Defendants' Motion for Judgment on Pleadings - 101 Motion Dkt #139
The Parties' claim construction briefs, exhibits, and submitted materials
2019.08.19 Claim Construction Opinion
2019.08.19 Claim Construction Order
2019.09.05 Reply Brief ISO Defendants' Motion for Judgment on the Pleadings
2020.01.02 Order on Motion for Judgment on the Pleadings
2020.01.07 Letter Regarding Remaining Asserted Claims

Discovery
2013.10.23 Arendi's Responses to Defendants' First Set of Interrogatories
2018.11.16 Para. 4(a) Identification of Accused Products for Motorola
2019.02.06 Para. 4(a) Identification of Accused Products for Google
2019.09.09 Google's Supplemental Response to Interrogatory No. 3
2019.10.30 Motorola's Supplemental Response to Interrogatory No. 3

Accelerated Examination Support Documents
Accelerated Examination Support Document for '950 Patent
Accelerated Examination Support Document for '356 Patent
Accelerated Examination Petition and Support Document for '993 Patent
Accelerated Examination Support Document for '639 Patent App./Patent Pub. 2011/0214052

Other Produced Documents
AHL0001774-78
AHL0121553-63
AHL0122361-66
AHL0125942-6049
AHL0132649-67
AHL0132730-62
AHL0154945-5143
AHL0194629-67

ARENDI 5498-514
ARENDI 7846-73
ARENDI 147660-907
ARENDI 162876-3008
ARENDI 214286-91
ARENDI 315568-765

Exhibit A

ARENDI 321256-91
ARENDI 330487-95

ARENDI-DEFS00000001
ARENDI-DEFS00000003-103
ARENDI-DEFS00000568-74
ARENDI-DEFS00000591-606
ARENDI-DEFS00000623-27
ARENDI-DEFS00000628-30
ARENDI-DEFS00000631-32
ARENDI-DEFS00000642-46
ARENDI-DEFS00000647-711
ARENDI-DEFS00000778-79
ARENDI-DEFS00000780-81
ARENDI-DEFS00001151-60
ARENDI-DEFS00001186-262
ARENDI-DEFS00001285-94
ARENDI-DEFS00003323-28
ARENDI-DEFS00003329-37
ARENDI-DEFS00003338-644
ARENDI-DEFS00003645-48
ARENDI-DEFS00003649-4590
ARENDI-DEFS00004591-978
ARENDI-DEFS00004979-89
ARENDI-DEFS00004990
ARENDI-DEFS00004991-92
ARENDI-DEFS00004993-94
ARENDI-DEFS00004995-5285
ARENDI-DEFS00008330-429
ARENDI-DEFS00008492-622
ARENDI-DEFS00009002-30
ARENDI-DEFS00009168-90
ARENDI-DEFS00009278-94
ARENDI-DEFS00009411-40
ARENDI-DEFS00009441-60
ARENDI-DEFS00009479-94
ARENDI-DEFS00009587-615
ARENDI-DEFS00009616-26
ARENDI-DEFS00009653-71
ARENDI-DEFS00009800-13
ARENDI-DEFS00009814-25
ARENDI-DEFS00009850-68
ARENDI-DEFS00009869-79
ARENDI-DEFS00010090-105
ARENDI-DEFS00010157-73

Exhibit A

ARENDI-DEFS00010281-96
ARENDI-DEFS00010297-305
ARENDI-DEFS00010315-38
ARENDI-DEFS00010339-54
ARENDI-DEFS00010482-609
ARENDI-DEFS00010666-89
ARENDI-DEFS00010740-801
ARENDI-DEFS00010802-23
ARENDI-DEFS00010890-99
ARENDI-DEFS00010900-18
ARENDI-DEFS00010919-33
ARENDI-DEFS00010953-84
ARENDI-DEFS00010985-1004
ARENDI-DEFS00011045-65
ARENDI-DEFS00011088-98
ARENDI-DEFS00011168-85
ARENDI-DEFS00011204-41
ARENDI-DEFS00011349-57
ARENDI-DEFS00011358-70
ARENDI-DEFS00011371-82
ARENDI-DEFS00011410-20
ARENDI-DEFS00011455-56
ARENDI-DEFS00012596-631
ARENDI-DEFS00013019-617
ARENDI-DEFS00013637-57
ARENDI-DEFS00013665-75
ARENDI-DEFS00013693-888
ARENDI-DEFS00013889-905
ARENDI-DEFS00014053-64
ARENDI-DEFS00014065-104
ARENDI-DEFS00014105-11
ARENDI-DEFS00014112-19
ARENDI-DEFS00014250-374
ARENDI-DEFS00014671-706
ARENDI-DEFS00014879-87
ARENDI-DEFS00015016-28
ARENDI-DEFS00015295-6684
ARENDI-DEFS00017320-37
ARENDI-DEFS00017353-57
ARENDI-DEFS00017358-69
ARENDI-DEFS00017370-77
ARENDI-DEFS00017913-22
ARENDI-DEFS00018213-18
ARENDI-DEFS00018239-92
ARENDI-DEFS00019559-61

Exhibit A

ARENDI-DEFS00020646-66
ARENDI-DEFS00020872-969
ARENDI-DEFS00020970-1044
ARENDI-DEFS00021045-50
ARENDI-DEFS00021056-58
ARENDI-DEFS00021059-63
ARENDI-DEFS00021064-70
ARENDI-DEFS00021071-77
ARENDI-DEFS00021078-79
ARENDI-DEFS00021080-81
ARENDI-DEFS00021315
ARENDI-DEFS00022052-53
ARENDI-DEFS00022054
ARENDI-DEFS00022055
ARENDI-DEFS00022056-22064
ARENDI-DEFS00022065-22069
ARENDI-DEFS00022070-22077
ARENDI-DEFS00022078-22081

FOX_0000001-13090

HTC007286210-498

MILLER-APL00000025-57
MILLER-APL00000091-95
MILLER-APL00000096-103
MILLER-APL00000144-47
MILLER-APL00000271-73

MS 113733-4433

EDWARD A. FOX

PERSONAL INFORMATION:

- **HOME:** 203 Craig Dr., Blacksburg, VA 24060, +1-540-552-8667
- **WORK:** Email: fox@vt.edu; 114 McBryde Hall, Dept. of Computer Science, M/C 0106, Virginia Tech, Blacksburg, VA 24061 USA; +1-540-231-5113 [direct], x6931 [dept.], x6075 [FAX]; Digital Library Research Lab, x3615 (2030 Torgersen Hall)
- **CITIZENSHIP:** USA
- Homepage: http://fox.cs.vt.edu; Skype: edwardafox; Twitter: edwardafox; LinkedIn: foxedward; Facebook: edward.a.fox
- Google Scholar Edward Fox; ORCiD: 0000-0003-1447-6870; Github: edfox0714
- Public key for download from keys.openpgp.org for personal email edfox0714@gmail.com

EDUCATION:

- 8/83 Ph.D. Computer Science, Cornell University
  Area: information storage and retrieval
  Minor: linguistics; Advisor: Gerard Salton
  Title: Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types
- 1/81 M.S. Computer Science, Cornell University
- 2/72 B.S. Electrical Engineering (Computer Science Option), M.I.T.
  Advisor: J.C.R. Licklider (Director of Project MAC)
  Thesis advisor: Michael M. Kessler (Asst. Director of Library)

RECENT EMPLOYMENT:

- 12/2018- Chief Technology Officer, Mayfair Group LLC
- 2/2016- Professor (by courtesy), Dept. of Electrical and Computer Engineering, Virginia Tech
- 4/95- Professor, Dept. of Computer Science, Virginia Tech (VPI&SU)
- 1/98- Director, Digital Library Research Laboratory (Virginia Tech)
- 6/90-6/2014 Associate Director for Research, VPI&SU Computing Center; evolving into position of Faculty Advisor to Information Technology
- 5/88-4/95 Associate Professor, Dept. of Computer Science, VPI&SU
- 9/83-5/88 Assistant Professor, Dept. of Computer Science, Virginia Polytechnic Institute & State Univ. (Virginia Tech) Blacksburg VA 24061-0106
- 8/82-4/83 Manager of Information Systems, International Inst. of Tropical Agriculture, P.M.B. 5320, Ibadan, Nigeria
- 9/78-8/82 Instructor, Research Assistant, Teaching Assistant, Dept. of Computer Science, Cornell Univ., Ithaca NY 14853
- 9/72-8/78 Data Processing Manager, Vulcraft, Div. of NUCOR Corporation, Florence, SC 29501

- 9/71-6/72 Data Processing Instructor, Florence Darlington Technical College, Florence, SC 29501

MEMBERSHIP IN PROFESSIONAL SOCIETIES:

- Association for Information Science & Technology (ASIS&T) and SIGDL
- ACM (Fellow starting 2017, Member since 1967) and SIGIR, SIGCSE
- IEEE (Fellow starting 2017, Senior Member since 2004), IEEE-CS, and Technical Committee on Multimedia Computing
- Sigma Xi (Assoc. Member 7/1/1972, Full Member 7/1/1985)
- Upsilon Pi Epsilon

HONORS AND AWARDS:

- While at VPI&SU
  - Inaugural inductee ACM SIGIR Academy, class of 2020, announcement: "to honor and recognize individuals who have made significant, cumulative contributions to the development of the field of information retrieval (IR). Inductees to the SIGIR Academy are the principal leaders in IR, whose efforts have shaped the discipline and/or industry through significant research, innovation, and/or service."
  - 2017 ACM Fellow, cited for contributions in information retrieval and digital libraries
  - 2017 IEEE Fellow, cited for leadership in digital libraries and information retrieval
  - Albert Nelson Marquis Lifetime Achievement Award by Marquis Who's Who, October 2017
  - XCaliber Award 2016 "for extraordinary contributions to technology-enriched learning activities" for project "Enhanced problem-based learning connecting big data research with classes", with students: Mohamed Farag, Richard Gruss, Tarek Kanan, Sunshin Lee, Xuan Zhang
  - Research Impact in Human-Computer Interaction, conferred by VT Center for HCI, October 16, 2015
  - Honeywell (2nd) Best Student Paper Award at IEEE BTAS 2012 for Robust Feature Extraction in Fingerprint Images Using Ridge Model Tracking: Nathaniel Short, Lynn Abbott, Michael Hsiao, Edward Fox
  - Virginia Tech Scholar of the Week, recognized by the Office of the Vice President for Research, for weeks of Dec. 10, 17, and 24, 2007
  - College of Engineering Dean's Award for Excellence in Service, April 2005
  - 1st Annual NDLTD Leadership Award, May 2004
  - Best Student Paper Award (as supervisor and 2nd author) at JCDL 2004: M. A. Goncalves, E. A. Fox, A. Krowne, P. Calado, A. Laender, A. S. da Silva, and B. Ribeiro-Neto. The Effectiveness of Automatically Structured Queries in Digital Libraries. pp. 98-107
  - First Prize Winner for Track 3, for presentation "Open Digital Libraries", $1500 prize, based in part on doctoral work of Hussein Suleman, AOL-CIT Finding Common Ground University Research Day, Nov. 6, 2002, Herndon, VA
  - Honorable Mention, Poster, MARIAN - Next Generation Digital Library System Built on 5S, $100 prize, with M. Goncalves, W. Richardson, and M. Luo, AOL-CIT Finding Common Ground University Research Day, Nov. 6, 2002, Herndon, VA
  - ACM Recognition of Service Award, 2018, for Program Co-chair, ACM/IEEE Joint Conference on Digital Libraries 2018
  - ACM Recognition of Service Award, 2012, for Associate Editor, ACM J. of Computing and Cultural Heritage (JOCCH) 2006-2012
  - ACM Recognition of Service Award, 1999, for Program Chair Digital Libraries '99
  - ACM Recognition of Service Award, 1996, for Program Chair Digital Libraries '96
  - ACM Recognition of Service Award, 1995, for SIGIR Chair 1991-1995

    - ACM Recognition of Service Award, 1995, for Program Chair SIGIR '95
    - ACM Recognition of Service Award, 1991, for Editor-in-Chief, ACM Press Database and Electronic Products
    - Awards for GeoSim: A GIS-Based Simulation Laboratory for Introductory Geography, L.W. Carstensen, Jr., C.A. Shaffer, R.W. Morrill, E.A. Fox:
        1. Computational Science award, Ames Lab., U.S. Dept. of Energy, 9/16/94;
        2. Best Article Related to Teaching in a University or College in National Council for Geographic Education's J. of Geography, May '92 - April '94.
    - Choice (Current Reviews for College Libraries) selection as one of 1989-90 Outstanding Academic Books and Nonprint Materials for "Hypertext on Hypertext" which I proposed, coordinated, and served as editor for ACM
    - Election to Phi Beta Delta (Honor Society for International Scholars), March 1999, http://www.vt.edu:10021/international/PhiBetaDelta/index.html
    - Election to Upsilon Pi Epsilon computing sciences honor society 3/17/98
    - Election to Sigma Xi (full member)
    - International Directory of Distinguished Leadership, Who's Who in the South and Southwest, Who's Who of Emerging Leaders in America, Who's Who in America, Who's Who in Science and Engineering, Who's Who in American Education, Young Community Leaders of America
- While at M.I.T.
    - William Stewart Award (founding and serving as first president of the ACM student chapter, founding and serving as first chairman of the Student Information Processing Board (still operating; see https://sipb.mit.edu/), serving as instructor and head of the engineering department of the student-run Saturday program for high school students)
    - Election to Sigma Xi (associate member)

PROFESSIONAL SERVICE:

CURRENT - major responsibilities:

1. 1996- Executive Director, Chairman of the Board, and Founder, Networked Digital Library of Theses and Dissertations (NDLTD, incorporated 5/2003), http://www.ndltd.org

CURRENT - Editor / Editorial Board Member:

1. 2014- PeerJ Computer Science, launched 2/3/15, https://peerj.com/computer-science/
2. 2012- Editorial Advisory Board, Thailand's National Institute of Development Administration (NIDA) Development Journal
3. 2012- Editorial Advisory Board, Thailand's National Institute of Development Administration (NIDA) International Journal of Development Administration
4. 2008- Editorial board, Journal of Intelligent Information Systems (JIIS)
5. 2004- Editorial board, International Journal on Digital Libraries (IJDL, www.dljournal.org)
6. 1994- Foundation Editor, The Journal for Universal Computer Science (Springer)

CURRENT - conferences/workshops (selected):

1. Co-chair, Web Archiving and Digital Libraries (WADL 2020), virtual workshop at virtual conference, held with JCDL 2020 (Aug. 1-5, hosted by Wuhan University), with chair Zhiwu Xie and co-chair Martin Klein
2. Member, Steering Committee, Joint Conference on Digital Libraries (JCDL), member since 2003, chair from 2010 to 9/2014
3. Member, International Advisory Committee, International Conference of Asian Digital Libraries (ICADL)
4. Senior PC Member, TPDL 2020 Papers; PC Member, posters and demos
5. Senior PC Member, JCDL 2020 Full and Short Papers
6. Senior PC Member, SIGIR 2020 Full Papers; PC Member, Short Papers
7. Member, Program Committee, CIKM 2020, Short paper track
8. Member, Program Committee, EEKE2020, 1st Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents, at JCDL 2020
9. Member, Program Committee, BIR2020 (10th International Workshop on Bibliometric-enhanced Information Retrieval)
10. Member, Program Committee, BIRDS 2020, workshop at SIGIR 2020, Xi'an, PRC
11. Member, Program Committee, International Conference on eDemocracy & eGovernment (ICEDEG) 2020
12. Member, Program Committee, ICADL 2020
13. Member, Program Committee, WSDM 2021, Jerusalem, Israel, March 2021

CURRENT - other (selected):

1. Member, 2020 IEEE CS Fellows Evaluation Committee
2. Group manager of Web 2.0 services: LinkedIn groups on: Digital Libraries, VT Computer Science, ...

CONFERENCES/WORKSHOPS CHAIRED:

1. Co-chair, Web Archiving and Digital Libraries (WADL 2019), held with JCDL 2019, with co-chair Zhiwu Xie and chair Martin Klein
2. Program Chair, ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2018), Fort Worth, Texas, with co-chairs Vivien Petras and Min-Yen Kan
3. Co-chair, Web Archiving and Digital Libraries (WADL 2018), held with JCDL 2018, with co-chair Zhiwu Xie and chair Martin Klein
4. Chair, Web Archiving and Digital Libraries (WADL 2017), held with JCDL 2017, with co-chairs Zhiwu Xie and Martin Klein
5. Chair, Web Archiving and Digital Libraries (WADL 2016), held with JCDL, Rutgers Univ., Newark, NJ, June 19-23, 2016, with co-chairs Zhiwu Xie and Martin Klein
6. Chair, Web Archiving and Digital Libraries (WADL 2015), held with JCDL, June 24-24, 2015, Knoxville, TN
7. Chair, Web Archiving and Digital Libraries (WADL 2013), workshop at JCDL 2013, Indianapolis, July 25-26
8. Co-chair, Webinar on Emergency Informatics and Digital Libraries, July 24, 2012, http://www.ctrnet.net/webinars (with S. Sheetz as Chair)
9. Chair, Digital Libraries and Education, 1/2 day workshop at JCDL/ICADL 2010, June 21-25, Gold Coast, Australia
10. Chair, Development and Adoption of Computational Thinking Curricular Guidelines Across Disciplines, Living In the KnowlEdge Society (LIKES) Workshop 6 (http://www.likes.org.vt.edu/?q=node/203), Feb. 24-26, 2010, Durham, NC, workshop for the NSF-funded LIKES project, http://www.livingknowledgesociety.org

11. Chair, Curricular Guidelines Connecting Computing with Other Disciplines, Living In the KnowlEdge Society (LIKES) Workshop 5 (http://www.likes.org.vt.edu/?q=node/196), Nov. 12-13, 2009, Virginia Tech, Blacksburg, VA, workshop for the NSF-funded LIKES project, http://www.livingknowledgesociety.org
12. Co-Chairs Edward A. Fox, Seungwon Yang, Steve Sheetz, Christopher Zobel, and Patrick Fan. Coordinating FDI Track E: About LIKES - Living In the KnowlEdge Society, July 21-23, 2009, Virginia Tech
13. PC chair of the Information Retrieval track, ACM CIKM 2006 (http://sa1.sice.umkc.edu/cikm2006/), Arlington, VA, Nov. 6-11, 2006
14. Co-chair, Program Committee, 8th International Conference of Asian Digital Libraries (ICADL 2005), Bangkok, Thailand, Dec. 12-15, 2005
15. Co-Chairs Steve Edwards, Dwight Barnette, and Edward Fox: JETT (JAVA Engagement for Teacher Training) Workshop, supported by IBM ECLIPSE, Virginia Tech, Blacksburg, VA, July 21-22, 2004
16. Chair: Digital Libraries: Global Reach and Impact, JCDL 2004 workshop, June 10-11, 2004, Tucson, AZ, co-chair: Ching-chih Chen
17. Co-organizers/coordinators Manuel A. Perez-Quinones, Lillian Cassel, Edward Fox, John Impagliazzo, J.A.N. Lee, C. Lee Giles: Using the NSF Digital Library to Enhance Your Teaching, workshop 8 at ACM SIGCSE 2004, Norfolk, VA, March 3-7, p. 503
18. Chair: Indo-US Workshop on Open Digital Libraries and Interoperability, Arlington, VA, June 23-25, 2003, 39 participants from the United States and India, sponsored by Indo-US Science and Technology Forum and CIT, http://fox.cs.vt.edu/IndoUSdl
19. Chair: Open Source Roadshow, Case Study: Mozilla Project, Virginia Tech, October 24, 2002, assisted by Netscape/AOL/Mozilla and CIT, http://fox.cs.vt.edu/opensource.htm
20. Co-organizer: Project Kaleidoscope (PKAL) 2002 Summer Institute Cluster II (DL and CS), http://www.pkal.org/siplanning/index.html, June 2-5, 2002, Williamsburg, VA
21. Chair: NDLTD and OAI: A Case Study of a Worldwide Community Sharing (Multilingual, Multimedia) Electronic Theses and Dissertations through the Open Archives Initiative, CNI Spring Meeting session, 4/15/2002, Washington, D.C.
22. Chair: Open Archives: Communities, Interoperability, and Services; Workshop for 24th Annual International SIGIR Conference on Research and Development in Information Retrieval, New Orleans, ULA, Sept. 13, 2001, Proceedings (w. H. Suleman): http://purl.org/net/oaisept01
23. Chair: First ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL'2001, Hotel Roanoke and Conference Center, Roanoke, VA, June 24-28, 2001, www.jcdl.org (and helped run PC mtg at UCLA, March 2-5, 2001)
24. Chair: Electronic Theses and Dissertations Standards Meeting. OCLC, Dublin, Ohio, Jan. 9-10, 2001.
25. Chair: Expanding the Information and Data Management (IDM) Research and Education Community, NSF-funded workshop, Hotel Roanoke and Conf. Center, Roanoke, VA, Oct. 2-4, 2000. http://fox.cs.vt.edu/IDM/
26. Chair: Extending Interoperability of Digital Libraries: Building on the Open Archives Initiative. ECDL Workshop, Lisbon, Portugal, Sept. 21, 2000. http://purl.org/net/oaisept00
27. Chair: US-Korea Joint Workshop on Digital Libraries: Removing Barriers to International Collaboration on Research and Education through Digital Libraries, Aug. 10-11, 2000, San Diego Supercomputer Center, San Diego, CA. http://fox.cs.vt.edu/UKJWDL/
28. Chair: Extending Interoperability of Digital Libraries: Building on the Open Archives Initiative. Workshop for ACM Hypertext'2000 and ACM Digital Libraries'2000. San Antonio, TX, June 3, 2000 http://purl.org/net/oaijune00
29. Chair, Steering Committee meetings of NDLTD on 3/15/2000, 9/15/2000, 3/21/2001, 10/26/2001
30. Program Chair: Virginia Internet Technology Week (VIT'99), Virginia Tech, Blacksburg, VA, September 13-17, 1999 (http://manta.cs.vt.edu/vit/)
31. Program Chair: ACM Digital Libraries '99, Berkeley, CA, Aug. 11-14, 1999
32. Co-Chair: 1999 NSF Information and Data Management Workshop: Research Agenda for the 21st Century - IDM'99, UCLA, March 7-9, 1999, http://www.cs.ucla.edu/csd/IDM99/index.html
33. Chair: Multimedia Education, workshop at ACM Multimedia'98, Bristol, England, Sept. 12, 1998 (with R. Heller, GWU).
34. Co-Chair: ETD/NTLTD Workshop, MECCA Conf., Memphis, TN, June 10-11, 1998, http://www.mecca.org/mecca-itec-conf/conftopics.html

35. Chair: IBM Renaissance Meeting, Oct. 20-22, 1997, Virginia Tech, Blacksburg, VA
36. Chair: Education and Curriculum Development for Multimedia, Hypertext, and Information Access: Focus on Digital Libraries and Information Retrieval workshop, sponsored by both ACM Digital Libraries'97 and ACM SIGIR'97, July 23, 1997, Philadelphia.
37. Chair: Interactive Learning with a Digital Library in Computer Science workshop, June 15-21, 1997, Virginia Tech, Blacksburg, VA funded in part by NSF (as supplement to EI grant 1993-97)
38. Chair: Courseware, Education and Curriculum in Multimedia workshop, held with ICMCS'97, IEEE International Conference on Multimedia Computing and Systems, June 6, 1997, Ottawa, Canada
39. Chair: Joining the National Digital Library of Theses and Dissertations, half-day pre-conference seminar for CAUSE/CNI regional conference, Univ. of Del., May 21, 1997
40. Chair: Courseware, Training and Curriculum in Multimedia workshop, held with ACM Multimedia'96, Boston, MA, Nov. 19, 1996
41. Chair: Courseware, Training and Curriculum in Information Retrieval workshop, held after ACM SIGIR'96, Zurich, Switzerland, Aug. 22, 1996.
42. Chair: Workshop on Electronic Theses and Dissertations in the Southeast, sponsored by SURA, hosted by UNC Charlotte, Charlotte, NC, Aug. 1-2, 1996.
43. Program Chair: Digital Libraries'96, First ACM International Conference on Digital Libraries, March 20-23, 1996, Bethesda, MD
44. Chair: Workshop for Working Group on Theses, Technical Reports, and Dissertations, in Monticello Electronic Library Initiative, sponsored by SURA and SOLINET, Aug. 12-13, 1994, Virginia Tech, Blacksburg, VA
45. Conference chair: Workshop on Information Retrieval and Genomics, sponsored by ACM SIGIR and SIGBIO, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD, May 2-4, 1994
46. Chair: Workshop on Information Access and the Networks, IANET'93, held in conjunction with ACM SIGIR '93, July 1, 1993
47. Program chair: 18th Intern'l Conf. on R & D in Information Retrieval, SIGIR '95, July, 1995, Seattle, WA

EARLIER (major responsibilities):

1. 1990-2019 Editorial board, Journal of Educational Multimedia and Hypermedia (AACE)
2. 1994-2019 Editorial board, Multimedia Tools and Applications (Kluwer)
3. 2017-2019 Member, IEEE Thesaurus Editorial Board
4. 2018-2019 Member, ACM Publications Board, and Co-chair, Digital Library Committee
5. 2013-2016 Editor, ACM Book Series section on Information Retrieval and Digital Libraries (with Morgan and Claypool Publishers)
6. 1987-2014 Editorial board, Information Processing & Management (Elsevier)
7. 2010-2013 Member, CRA Board, and Member, CRA-E (Computing Research Association Committee on Education); Member, Review Committee, Outstanding Undergraduate Research Awards 2011; Member, Scholarly Publications Group; Co-chair (July 1, 2011 to June 30, 2013) of Membership Committee
8. 2002-2011 (and 1989-1993) Editorial board, ACM Transactions on Information Systems (TOIS)
9. 2005-2009 Member, Advisory Group for Programmes on Digital Repositories and on Digital Asset Management and Preservation (assisting JISC, the NSF-equivalent in UK, including attending London meeting on 25 July 2006)
10. 2004-2008 Chairman, IEEE-CS Technical Committee on Digital Libraries
11. 2004-2008 Member, Advisory Board, DELOS Network of Excellence on Digital Libraries (http://www.delos.info/), funded by the European Union's 6th Framework Programme, first meeting December 9, 2004 in Athens, second meeting December 7-8, 2005 in Sophia-Antipolis, France, third meeting January 14-15, 2007 in Rome, fourth meeting Dec. 7, 2007 in Italy

12. 1995-2008 Editor, Morgan Kaufmann Publishers, Inc. Series on Multimedia Information and Systems. Books in series include:
    1. How to Build a Digital Library - by Witten and Bainbridge
    2. Digital Watermarking - by Cox, Miller, and Bloom
    3. Readings in Multimedia Computing and Networking - edited by Jeffay and Zhang
    4. Introduction to Data Compression, Second Edition - by Sayood
    5. Multimedia Servers: Applications, Environments, and Design - by Sitaram and Dan
    6. Managing Gigabytes: Compressing and Indexing Documents and Images, Second Edition - by Witten, Moffat, and Bell
    7. Digital Compression for Multimedia: Principles and Standards - by Gibson, Berger, Lookabaugh, Lindbergh, and Baker
    8. Practical Digital Libraries: Books, Bytes, and Bucks - by Lesk
    9. Readings in Information Retrieval - by Jones and Willett
13. 2005-2007 Member, Computational and Informational Sciences Directorate Review Committee, Pacific Northwest National Laboratory (PNNL), Richland, WA
14. 2001-2006 Director, Computing and Information Technology Interactive Digital Educational Library (CITIDEL), www.citidel.org
15. 2000-2006 Co-editor-in-chief, ACM Journal of Education Resources In Computing (JERIC - and lead proposer 1998-2000), http://www.acm.org/pubs/jeric/
16. 2000-2006 Member, ACM SIG Multimedia Executive Board
17. 1999-2006 Member, Steering Committee, Open Archives Initiative http://www.openarchives.org
18. 2004-2005 Member, Advisory Board, Development of a Distributed Digital Library Training Certificate Program, University of North Carolina at Chapel Hill
19. 2002-2005 Member, Policy Committee, NSDL, National Science Digital Library, old long name: National STEM (Science, Technology, Engineering, and Mathematics) education Digital Library, www.nsdl.org (the steering committee of an initiative with over $150M NSF funding, to transform education throughout the USA at all levels in STEM areas), chair 1/2002 - 5/2003
20. 2001-2005 Advisory board, IMEJ of Computer-Enhanced Learning (http://imej.wfu.edu)
21. 1997-2005 Editor, Computer Science Teaching Center (CSTC, www.cst.org)
22. 1998-2004 Member, Governing Board, Internet Technology Innovation Center http://www.internettic.org
23. 2000-2003 Executive Editor, TICtalk Newletter, a monthly publication of Internet Technology Innovation Center
24. 1997-2003 Editorial board, IEEE Multimedia (IEEE-CS)
25. 1992-2001 Editorial board, Multimedia Systems (ACM/Springer-Verlag)
26. 1995-2001 Member, NCSTRL (Networked CS Tech. Report Library) working group (part of D-Lib working group program)
27. 1997-2000 Member, Steering Committee, ACM Preprint Database (and CoRR, Computing Research Repository, http://xxx.lanl.gov/archive/cs/intro.html)
28. 1995-2000 Co-chair, ACM SIGMM Education Committee
29. 1994-99 Editorial board, Electronic Publishing - Origination, Dissemination and Design Journal (Wiley)
30. 1996-98 Chair, Steering Committee, ACM Digital Libraries conf. series
31. 1994-95 Past Chair, member ACM Multimedia Conferences Steering/Advisory Committee
32. 1991-95 Chairman, ACM SIGIR (Special Interest Group on Information Retrieval)
33. 1992-94 Chair and Founder, ACM Multimedia Conferences Steering/Advisory Committee
34. 1989-93 Associate editor, ACM Transactions on Information Systems
35. 1988-92 Member ACM Publications Board
36. 1988-91 Editor-in-chief ACM Press Database and Electronic Products. Responsible for electronic publishing plans, products, and services. Publications include "Hypertext on Hypertext;" the "Resources in Computing" book series; CD-ROMs in the DALibrary and Computer Archive

products; a videotape documentary, etc.
37. 1987-91 Vice chairman ACM SIGIR (Special Interest Group on Information Retrieval)
38. 1987-91 Editor ACM Press Books Database Products
39. 1985-89 Editor, mailing list maintainer, and founder of IRList, an electronic digest on information retrieval and related topics.
40. 1985-87 Co-editor for ACM SIGIR Forum, the newsletter for SIGIR

EARLIER (other):

1. Member, International Advisory Committee, 12th International CALIBER 2019 (Convention on Automation of Libraries in Education and Research Institutions), 28-30 November 2019
2. Senior Member, Program Committee, TPDL 2019
3. Member, Program Committee, ICADL 2019
4. Member, Program Committee, BIRNDL 2019
5. Member, Program Committee, CIKM 2019
6. Member, Program Committee, SIGIR 2019 (The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval)
7. Senior Member, Program Committee, JCDL 2019, Urbana-Champaign, IL, June 2-6
8. Member, Program Committee, ICEDEG 2019, Sixth International Conference on eDemocracy & eGovernment, 24-26 April 2019, Quito-Ecuador
9. Member, Program Committee, 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019), co-located with the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), Paris, France, July 25, 2019
10. Member, Program Committee, BIR-AT-ECIR2019 (8th International Workshop on Bibliometric-enhanced Information Retrieval), Cologne, GE, April 14
11. Member, Program Committee, BIRDS 2019, with ASIST, Melbourne
12. Member, 2018 IEEE CS Fellows Evaluation Committee
13. Senior Member, Program Committee, TPDL 2018, 22nd International Conference on Theory and Practice of Digital Libraries, Porto, Portugal, 10-13 Sept.
14. Member, Program Committee, BIR@ECIR 2018, Grenoble, France, March 26-29
15. Member, Program Committee, 5th International Conference on eDemocracy & eGovernment (ICEDEG), 4-6 April 2018, Ambato-Ecuador
16. Member, Program Committee, CIKM 2018, International Conference on Information and Knowledge Management, Lingotto, Turin, Italy, 22-26 Oct.
17. Member, Program Committee, MEDA 2018, Workshop on Curative Power of MEdical DAta, held with JCDL 2018, https://profs.info.uaic.ro/~meda/
18. Member, Program Committee, SIGIR 2018 (human factors and interfaces track), Ann Arbor, Michigan, 8-12 July
19. Member, Program Committee, ICADL 2018, The 20th International Conference on Asia-Pacific Digital Libraries, Waikato, NZ, Nov. 19-22
20. 1999-2017 Member, Advisory Board for D-Lib Forum, www.dlib.org
21. Reviewer, NSF, 2017, and review panels in spring and fall 2017
22. Member, Program Committee, ICADL 2017 (19th International Conference on Asia-Pacific Digital Libraries), Chulalongkorn University, Bangkok, Thailand, November 13-15, 2017

23. Member, Program Committee, CIKM 2017, ACM Conference on Information and Knowledge Management, Singapore, November 6-10, 2017
24. Member, Program Committee, BIRNDL 2017, 2nd Joint Workshop on Bibliometric-enhanced IR and NLP for Digital Libraries, with SIGIR 2017, Tokyo, Japan
25. Meta-reviewer, Senior Program Committee, ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, Canada, 19-23 June 2017
26. Member, Program Committee, TPDL 2017 (21st International Conference on Theory and Practice of Digital Libraries, Thessaloniki, Greece, September 18-21, 2017)
27. Member, Program Committee, BIR 2017 (5th International Workshop on Bibliometric-enhanced Information Retrieval, at ECIR 2017, 39th European Conference on Information Retrieval, Aberdeen, Scotland, 8-13 April 2017)
28. Member, NSF review panel, fall 2016
29. Member, Nominating Committee for IEEE TCDL Chair for term 2016-2017
30. Member, Program Committee, BIR 2016 (3rd International Workshop on Bibliometric-enhanced Information Retrieval), with ECIR 2016
31. Meta-reviewer, Program Committee, ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, June 19-23, 2016
32. Member, Program Committee, BIRNDL 2016, Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries, a workshop federating the 4th Bibliometric-enhanced Information Retrieval (BIR) and the 2nd Workshop on text and citation analysis for scholarly digital libraries (NLPIR4DL), at JCDL, Rutgers Univ., Newark, NJ, June 19-23, 2016
33. Member, Program Committee, TPDL 2016, 20th International Conference on Theory and Practice of Digital Libraries, September 5-9, 2016, Hannover, Germany
34. Member, Program Committee, CERI 2016 Teaching IR Track, 4th Spanish Conf. in Information Retrieval, http://www.ugr.es/~ceri16/ss.php
35. Chair for plenary session 3, Open Access, 19th International Symposium on Electronic Theses and Dissertations, ETD 2016 "Data and Dissertations", 11-13 July 2016, Lille (France)
36. Editorial board, IEEE-TCDL Special Issue on Data Curation, V. 12, No. 1, May 2016, http://www.ieee-tcdl.org/Bulletin/v12n1/
37. Meta-reviewer, Program Committee Member, 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Sept. 14-18, 2015, Poznan, Poland
38. Member, Program Committee, 2015 NSF Workshop on Curricular Development for Computing in Context, CIC15, May 13, 2015, Stetson U., Deland FL, http://cbia.stetson.edu/cic
39. Meta-reviewer, Program Committee Member, JCDL 2015, June 21-25, 2015, Knoxville, TN
40. Member, Program Committee, Knowledge Maps and Information Retrieval (KMIR), workshop at JCDL2015, June 24-25, 2015, Knoxville, TN
41. Meta Reviewer, Digital Libraries 2014, conjoined conference for Joint Conference on Digital Libraries (JCDL) and Theory and Practice of Digital Libraries (TPDL) Conference >Reviewer, NSF, 2017
42. Member, Program Committee, Knowledge Maps and Information Retrieval (KMIR), workshop at DL 2014, London, 11 Sept. 2014
43. Member, Program Committee, ECIR 2015 workshop "Bibliometric-enhanced Information Retrieval", part of the 37th European Conference on Information Retrieval (ECIR), March 29, 2015, Vienna, http://www.gesis.org/en/events/events-archive/conferences/ecirworkshop2015/
44. 2013 Editorial board, The Open Information Science Journal (TOISJ)
45. Senior Program Committee Member, TPDL 2013, and TPDL Workshops Program Committee Member
46. Senior Program Committee Member, JCDL 2013
47. Member, Organizing Committee, Contemplative Practices for a Technological Society, April 11-13, 2013, Blacksburg, VA
48. Invited Attendee, Scholarly Communications Workshop, U. Pittsburgh, Jan. 13-15, 2013
49. Reviewer for ISCRAM 2013

50. 2006-2012 Editorial board, ACM Journal on Computers and Cultural Heritage (JOCCH), and serving 2008-2009 on search committee for next editor-in-chief
51. 2011 Editorial board, ACM Computers in Entertainment (CiE)
52. Member, Program Committee, ICADL2012, International Conference on Asia-Pacific Digital Libraries, Taipei, Taiwan, 12-15 Nov., http://www.icadl2012.org/
53. Member, Program Committee, 16th International Conference on the Theory and Practice of Digital Libraries (TPDL 2012 -- http://www.tpdl2012.org/), Cyprus, September 23-27, 2012
54. Member, Workshop Committee, 16th International Conference on the Theory and Practice of Digital Libraries (TPDL 2012 -- http://www.tpdl2012.org/), Cyprus, September 23-27, 2012
55. Senior Member, Program Committee, JCDL 2012, Washington, D.C., June 10-14
56. Member, Steering Committee, NSF-funded CCC Vision Workshop on Computing for Disaster Management, Washington, D.C., April 24-25, 2012 (including work on report to NSF due in June)
57. Member, Program Committee, 34th European Conference on Information Retrieval (ECIR 2012), Barcelona, Spain, 1-5 April 2012 (http://ecir2012.upf.edu)
58. Member, Program Committee, ACM SIGCSE 2012, Feb. 29 - March 3
59. Member, Program Committee, Personal Information Management 2012, workshop at CSCW, Seattle, Feb. 11-12, 2012
60. 2011-2 Member, CS2013 Committee for Knowledge Areas: Computational Science, Information Management
61. Member, Program Committee, ACM Multimedia 2011, Scottsdale, AZ, Nov. 28 - Dec. 1, 2011
62. Member, Program Committee, ICADL2011, International Conference on Asia-Pacific Digital Libraries, Beijing, China, 24-27 Oct., http://www.icadl2011.org/
63. Member, Program Committee, Information and Data Management Track, 26th Brazilian Symposium on Databases, SBBD, Florianopolis, Brazil, Oct. 3-6, 2011
64. Senior Program Committee Member, International Conference on Theory and Practice of Digital Libraries, TPDL, Berlin, Germany, September 25-29, 2011 (http://www.tpdl2011.org/)
65. Area Chair, Program Committee, 34th ACM International Conference on Research and Development in Information Retrieval (SIGIR 2011 -- http://www.sigir2011.org/) Beijing, China, July 24-28, 2011
66. Invited Attendee, Microsoft Research Faculty Summit, Redmond, WA, July 18-20, 2011
67. Member, Program Committee, 19th International Symposium on Methodologies for Intelligent Systems, ISMIS 2011, June 28-30, Warsaw, Poland, http://ismis2011.ii.pw.edu.pl/
68. Member, Program Committee (and attending PC meeting), Joint Conference on Digital Libraries (JCDL 2011), June 13-17, Ottawa, www.jdcl2011.org.
69. Member, Programme Committee, European Conference on Information Retrieval (ECIR) 2011, 18-21 April, Dublin, Ireland
70. 2009-2011, Member, DL.org (EU funded organization for digital libraries) Working Group on Functionality, meeting in Greece June 2009 and in Rome May 2010, along with periodic online meetings
71. 2009-2010, Chair, Working Group on Interdisciplinary Computing Curricula, at the 2009 Rebooting Computing Summit and following up into the future
72. Member, Program Committee, Information Retrieval track, 25th Brazilian Symposium on Databases (SBBD 2010), Belo Horizonte, Brazil, Oct. 5-8
73. Senior Program Committee Member, ECDL 2010, 14th European Conference on Digital Libraries, Glasgow, September 6-10, 2010, http://www.ecdl2010.org/

74. Member, Program Committee, Fourth Annual Workshop on Human Computer Interaction and Information Retrieval (HCIR 2010), New Brunswick, NJ, August 22, 2010
75. Senior Program Committee Member and Area Chair (Applications), SIGIR 2010, 33rd ACM International Conference on Research and Development in Information Retrieval, Geneva, Switzerland, July 18-23, 2010
76. Member, Program Committee, CRA Biennial Conference at Snowbird, 2010, Cliff Lodge, Snowbird Resort, Utah, July 18-20
77. Attendee, Media Computation Workshop, Virginia Tech, July 7-9, 2010
78. Member, Program Committee, ITiCSE 2010, The 15th Annual Conf. on Innovation and Technology in Computer Science Education, Bilkent, Ankara, Turkey, June 26-30, 2010, http://iticse2010.bilkent.edu.tr/
79. Member, Program Committee, JCDL 2010, Joint Conference on Digital Libraries, Gold Coast, Australia, June 21-25, 2010
80. Member, Program Committee, ICADL2010, International Conference on Asia-Pacific Digital Libraries, Brisbane, Australia, 21-25 June 2010, http://www.jcdl-icadl2010.org/
81. Member, Program Committee, Doctoral Consortium, JCDL/ICADL 2010, June 21-25, Gold Coast, Australia
82. Member, Program Committee, ETD 2010, June 16-18, Austin, TX
83. Member, Paper Review Committee, 4th International Symposium on Information Technology 2010, ITSIM 2010, Kuala Lumpur, June 15-17, http://www.itsim.org/
84. Member, Programme Committee, 32nd European Conference on Information Retrieval (ECIR 2010), The Open University in Milton Keynes, UK, March 28-31, http://kmi.open.ac.uk/events/ecir2010/
85. Member, Program Committee, ACM SIGCSE 2010, Milwaukee, March 10-13
86. 2008-2009, Co-editor of ACM TOMCCAP special issue, led by Christoph Rensing, http://tomccap.acm.org/special2009.html
87. Member, Program Committee, ECDL 2009, Sep. 27 - Oct. 2, 2009, Corfu
88. Member, Program Committee, ISMIS'09, Prague, Czech Republic, September 14-17, 2009
89. Area Chair (AC, Senior Program Committee member) for IR Applications, 32nd ACM International Conference on Research and Development in Information Retrieval, SIGIR 2009, http://www.sigir2009.org, Boston, July 19-23, 2009
90. Mentor for SIGIR 2009
91. Member, Program Committee, ITiCSE 2009, July 6-8, 2009, Paris
92. Member, Program Committee, JCDL 2009, Austin, TX, June 15-19, 2009
93. Member, Program Committee, 12th Intl. Symposium on Electronic Theses and Dissertations (ETD 2009), Pittsburgh, PA, June 10-13
94. Reviewer, DigCCurr2009: Digital Curation Practice, Promise, and Prospects, Chapel Hill, NC, April 1-3, 2009
95. Member, Program Committee, ECIR'09: 31st European Conference on Information Retrieval, April 2009
96. 2001-2008 Member, NCSTRL (Networked Computer Science Technical Reference Library, www.ncstrl.org) Committee
97. Member, Program Committee, 11th International Conference on Asian Digital Libraries, ICADL 2008, Bali, Indonesia, Dec. 2-5, 2008
98. Member, Steering Committee, 2008 NSF CISE Pathways to Revitalized Undergraduate Computing Education, CPATH 2008, Arlington, VA, Nov. 12-14, 2008
99. Member, Program Committee, SPIRE 2008, 15th String Processing and Information Retrieval Symposium, Melbourne Australia, November 10-12, 2008
100. Member, Program Committee, HCIR 2008, Workshop on Human-Computer Interaction and Information Retrieval, October 23, Microsoft Research, Redmond, Washington
101. Member, Program Committee, Brazilian Symposium on Databases (SBBD), Campinas, Brazil, October 13-15, 2008
102. Member, Program Committee, European Conference on Research and Advanced Technology for Digital Libraries: Towards the European Digital Library, ECDL2008, Aarhus, Denmark, September 14 - 19, 2008
103. Senior Member, Program Committee, SIGIR 2008, July 20-24, 2008, Singapore

104. Member, Program Committee, JCDL 2008, June 15-20, 2008, Pittsburgh
105. Member, Program Committee, ISMIS 2008, May 20-23, 2008, Toronto
106. 2007 Editorial board, The Open Information Systems Journal (TOIJ)
107. Co-chair, Panels, ECDL 2007, Budapest, September 2007
108. Senior PC member, ACM SIGIR 2007, attending PC meeting March 29-30, 2007 in Delft, Netherlands, 30th Annual International ACM SIGIR Conference, 23-27 July 2007, Amsterdam
109. Co-chair (with Donatella Castelli), JCDL 2007 Workshop 2: 1st Workshop on Digital Library Foundations, with 2007 Joint Conference on Digital Libraries, June 23, 2007, Vancouver
110. Co-chair (with Geneva Henry) standing in for Chair Nadia Caidi who could not attend, JCDL 2007 Doctoral Consortium, with 2007 Joint Conference on Digital Libraries, June 19, 2007, Vancouver
111. Member (of 6), Organizing Committee, JCDL 2007 Workshop 1: Developing a Digital Libraries Education Program, with 2007 Joint Conference on Digital Libraries, June 18, 2007, Vancouver
112. Member, Program Committee, JCDL 2007, 2007 Joint Conference on Digital Libraries, June 17-23, 2007, Vancouver
113. Member, Program Committee, ETD 2007, 10th International Symposium on Electronic Theses and Dissertations, Uppsala, Sweden, June 13-16, 2007, http://epc.ub.uu.se/etd2007/
114. Member, Program Committee, MUE 2007, 26-28 April 2007, Seoul, Korea
115. Member, Program Committee, 29th European Conference on Information Retrieval (ECIR 2007), Rome, April 2-5, http://ecir2007.fub.it
116. Invited IMA workshop attendee, The Evolution of Mathematical Communication in the Age of Digital Libraries, Dec. 8-9, 2006, hosted by the Institute for Mathematics and its Applications (IMA), Minneapolis, http://www.ima.umn.edu/2006-2007/SW12.8-9.06/index.html
117. Member, Program Committee, ICADL 2006, Kyoto, Japan, Nov. 2006, http://www.icadl.org/
118. Member, Program Committee, SPIRE 2006, 13th ed. Symposium on String Processing and Information Retrieval (http://www.cis.strath.ac.uk/external/spire06/), 11-13 Oct. 2006, Glasgow
119. Member, Program Committee, ECDL 2006, Alicante, Spain, 17-22 Sept. 2006, http://www.ecdl2006.org/, and Accompanying Professor, Doctoral Consortium
120. Co-organizer, ETANA Workshop, Vanderbilt University, Nashville, Sept. 8-9, 2006
121. Member, Program Committee, SIGIR 2006 Workshop "Evaluating Exploratory Search Systems" (http://www.sigir2006.org/), Aug. 6-11, Seattle; Area Coordinator (AC) for ACM SIGIR 2006, for Web IR and Digital Libraries, coordinating the work of the reviewers in that area
122. Member, JCDL 2006 program committee, June 11-15, 2006, Chapel Hill, NC
123. Accompanying Professor, Doctoral Consortium, JCDL 2006, June 11-15, 2006, Chapel Hill, NC, http://www.jcdl2006.org/
124. Member, Program Committee, ETD 2006, 9th International Symposium on Electronic Theses & Dissertations, June 7-10, 2006, Quebec City, http://www6.bibl.ulaval.ca:8080/etd2006/
125. Member, Program Committee, Southeast Workshop on Data and Information Management, Raleigh, NC, March 30-31, 2006
126. Member, Program Committee, 10th International Conference on Extending DataBase Technology (EDBT 2006), Munich, GE, March 27-31, 2006
127. Member, Program Committee, ACM Conference on Information and Knowledge Management (CIKM) 2005 (http://www.tzi.de/CIKM2005/), Bremen, Germany, October 31 - November 5
128. Member, Program Committee, ETD 2005, 8th International Symposium on Electronic Theses & Dissertations, Sydney, Australia, Sept. 28-30, 2005
129. Member, Organizing Committee and Co-chair for Panels, European Conference on Digital Libraries, ECDL 2005, Vienna, September 18-23, 2005, http://www.ecdl2005.org
130. Reviewer, SIGIR 2005, 28th Annual International ACM SIGIR Conference, Salvador, Brazil, August 15-19, 2005

131. Member, Program Committee, Fifth Joint Conference on Digital Libraries, JCDL 2005, June 7-11, 2005, Denver, Colorado
132. Member, Program Committee, 8th International Workshop of the DELOS Network of Excellence on Digital Libraries on Future Digital Library Management Systems (System Architecture & Information Access), Schloss Dagstuhl, Germany, March 29 - April 1st, 2005
133. 2001-2005 Member, Institutional Asset Submission Programme Steering Group (assisting JISC, the NSF-equivalent in UK, for FAIR program), including March 14, 2002 teleconference review panel
134. 2003-2004 Member, Energy Sciences & Technology Directorate Review Committee, Pacific Northwest National Laboratory (PNNL), Richland, WA
135. 2001-2004, Member, National Visiting Committee, "Web-Network Technology Curriculum Development" NSF grant to Erie Community College, SUNY Buffalo, ...; attending meetings in Buffalo, NY on 11/7/2001; June 11-13, 2002; June 18-19, 2003; June 15-16, 2004
136. Co-chair, Program Committee, ICADL 2004, 7th International Conference of Asian Digital Libraries, Shanghai, China, Dec. 13-17, 2004
137. Member, Program Committee, CIKM 2004, 13th Conference on Information and Knowledge Management, Nov. 8-13, 2004, Washington, D.C.
138. Member, Program Committee, SCCC'04, XXIV International Conference of the Chilean Computer Science Society, Nov. 11-12, 2004, Arica, Chile
139. Member, Program Committee, AIRS 2004, Asia Information Retrieval Symposium, Beijing, Oct. 14-16, 2004
140. Moderator, Discussion group on Information Integration, NSF IDM PI Workshop, Oct. 10-12, 2004, Cambridge, MA
141. Member, Program Committee, SPIRE 2004, String Processing and Information Retrieval Symposium, Padova, Italy, Oct. 5-8
142. Member, Program Committee, SIGIR 2004, 27th Annual International ACM SIGIR Conference, University of Sheffield, UK, July 25-29, 2004
143. Member, Program Committee, ELPUB 2004, ICCC 8th International Conference on Electronic Publishing, University of Brasilia, Brazil, June 23 - 26, 2004
144. Member, Program Committee, ISI-2004, 2nd Symposium on Intelligence and Security Informatics, Tucson, Arizona, June 10-11, 2004
145. Member, Program Committee, JCDL 2004, ACM/IEEE-CS Joint Conference on Digital Libraries, Tucson, Arizona, June 7-11, 2004
146. Member, Program Committee, ETD 2004, The 7th International Symposium on Electronic Theses & Dissertations: Distributing knowledge worldwide through better scholarly communication, Lexington, KY, June 3-5, 2004
147. Member, Program Committee, DLKC'04, International Symposium on Digital Libraries and Knowledge Communities in Networked Information Society, March 2-5, 2004, University of Tsukuba, Japan
148. Member, Program Committee, CIKM'2003, New Orleans, LA, Nov. 2-8, 2003
149. Member, Program Committee, SPIRE-2003, String Processing and Information Retrieval Symposium, Manaus, Brazil, October 8-10, 2003
150. Member, Program Committee, ENC 2003, Encuentro, biannual conf. of Mexican Computer Society (SMCC), Sept. 8-12, Apizaco, Mexico
151. Member, Program Committee, ECDL'2003, Trondheim, Norway, Aug. 17-22, 2003
152. Member, Program Committee, SIGIR'2003, Toronto, July 28-Aug.1, 2003
153. Member, Program Committee, Elpub 2003, 7th International Conference on Electronic Publishing, June 25-28, 2003, Guimaraes, Portugal
154. Member, Program Committee, The First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI 2003), June 2-3, 2003, Tucson, AZ, http://ecom.arizona.edu/ISI/
155. Member, Program Committee, JCDL'2003, Houston, May 27-31, 2003
156. 2001-2 Program Co-Chair, ICADL'2002, Singapore, Dec. 17-20, 2002, http://www.cais.ntu.edu.sg:8000/icadl02
157. Member, Program Committee, ECDL'2002 (and co-chair, demos and posters), Rome, Sep. 16-18, 2002
158. Member, Program Committee, SPIRE'2002, Lisbon, Portugal, Sep. 11-13
159. Member, Program Committee, JCDL'2002, Portland, July 14-18, 2002 (attending PC meeting March 8-10, 2002, Chapel Hill, NC)
160. Member, Int'l Program Committee, 2002 Workshop on Multimedia Information Retrieval and Management, June 5, 2002, Hong Kong Poly. U.
161. Member, Organizing Committee, OCKHAM (Open Community Knowledge Hypermedia Administration and Metadata), May 3, 2002, Atlanta, GA (sponsored by Digital Library Federation)

162. 2002 Co-organizer: Deborah Knox, Lillian Cassel, John Impagliazzo, Edward Fox, JAN Lee, Manuel Perez, and C. Lee Giles. "Submission, Evaluation, and Refinement of Resources to CITIDEL". Full day workshop (scheduled but cancelled due to attendance,attended only by co-PIs) at ACM SIGCSE Symposium 2002, http://www.cs.cofc.edu/sigcse2002/, in Proc. of the 33rd SIGCSE Technical Symposium on Computer Science Education, Feb. 27 - March 3, 2002, Covington, KY, p. 421
163. 2002 Invited participant, NIMD'2002 Workshop, Jan. 7-9, 2002, Arlington, VA (Novel Intelligence from Massive Data program of ARDA)
164. Member, Program Committee, ICADL 2001, Dec. 10-12, 2001, Bangalore, India, 4th Int'l Conf. of Asian Digital Libraries
165. Member, Program Committee and Area Coordinator, 24th Annual International SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, Sept. 2001 (and attending PC meeting 3/23-25/2000)
166. Member, Program Committee, ECDL 2001, 5th European Conference on Research and Advanced Technology for Digital Libraries, September 4-8 2001, Darmstadt, Germany, http://www.ecdl2001.org
167. Member, Program Committee, SPIRE'2001 - String Processing and Information REtrieval, Laguna de San Rafael, Chile, Nov. 13-15,2001
168. 2001 Invited participant, SMETE.ORG workshop, PKAL Summer Institute, July 15-18, 2001, Snowbird, Utah
169. 2001 Invited participant, brainstorming meeting on "Digital Libraries: Future Directions for a European Research Programme", organized by the DELOS Network of Excellence on Digital Libraries, in cooperation with the Cultural Heritage Applications Unit of the 5th FP IST programme, 13-15 June 2001, S. Cassiano, Italy, Brainstorming Report ERCIM-02-W02, http://delos-noe.org/
170. 2001 Member, Steering Committee, NSF IDM'2001, April 29 - May 1, 2001, Fort Worth, Texas, http://itlab.uta.edu/idm01/
171. 2001 Member, Review team for UMBC CS program, April 16, 2001, Baltimore
172. 2000-01 Member, Program Committee, WWW10 Conference, 10th International World Wide Web Conference, Hong Kong, June 2001, http://www.www10.org
173. 1999-01 Member, ACM Content Steering Committee (w. John White, charge at http://www.cs.arizona.edu/people/rts/Portal/hidden/steering.charge)
174. 1998-01 Member, Curriculum 2001 (CC2001) Review Committee (and member of focus group on Information Management)
175. 1998-01 Member, Ad Hoc Committee on Being a Scholar in the Digital Age, that helped to prepare "Issues for Science and Engineering Researchers in the Digital Age", National Research Council, ISBN 0-309-07417-7, Washington, D.C.: National Academy Press, 2001, 57 pages
176. 2000 Member, Board of Experts for NCLIS
177. 2000 Member, Program Committee, MMM 2000, November, 13-15, 2000, Nagano Japan, http://www.phys.waseda.ac.jp/shalab/mmm2000/
178. 2000 Member, Program Committee for The Ninth International Conference on Information and Knowledge Management (CIKM'2000), Nov. 6-11, 2000, Washington, DC, http://www.csee.umbc.edu/cikm/2000/
179. 2000 Member, Organizing Committee (OC) for the International Conference on Information Society in the 21 Century: Emerging Technologies and New Challenges (IS2000), November 5-8, 2000, The University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan
180. 2000 Member, review panels, NSF, Jan. 12-13, 2000 and May 22-25, 2000
181. 1999-2000 Member, Committee for National Research Council to prepare "On Being a Researcher in the Digital Age", for booklet published by NRC/NAS
182. 1999-2000 Member, Program Committee, ACM Multimedia 2000, Los Angeles, Oct. 30 - Nov. 3, 2000, http://www.acm.org/sigs/sigmm/MM2000/
183. 1999-2000 Member of the International Program Committee, SIGIR 2000, Athens, July 25-28, 2000
184. 1999-2000 Member, Coordination Group (led by R. Steinmetz, with W. Effelsberg and N. Georganas), "Multimedia for Multimedia: Learning and Teaching in the Next Decade", Dagstuhl seminar, Castle Dagstuhl, Germany, June 11-16, 2000
185. 1999-2000 Member, Program Committee, ACM Digital Libraries '2000, San Antonio, TX, June 2-7, 2000, http://www.dl00.org/
186. 1999-2000 Member, Program Committee, Web9 Conference, 9th International World Wide Web Conference, The Web: The Next Generation, Amsterdam, May 15-19, 2000, http://www.www9.org

187. 1999-2000 Member, Steering Committee, NSF IDM'2000, Chicago, March 5-7, 2000, http://boston.cwru.edu/idm00/
188. 1999-2000 Member, Program Committee, ICDE'2000, 16th International Conf. on Data Engineering, San Diego, CA, Feb. 28-March 3, 2000 http://research.microsoft.com/icde2000/
189. 1999 Member, Program Committee, 5th International Computer Science Conference (ICSC '99), Hong Kong, December 15-17, 1999
190. 1999 Participant, 7th Dublin Core Metadata Workshop, Die Deutsche Bibliothek, Frankfurt, October 25-27, 1999 (and presenter at Discussion Group - Dissertation Online - Oct. 27)
191. 1999 Invited participant and presenter, Workshop on an international project of electronic dissemination of theses and dissertations, UNESCO, Paris, Sep.27-28, 1999
192. 1999 Chair, Steering Committee meeting of NDLTD, 9/24/99, Washington
193. 1999 Member, Program Committee, 5th Eurographics Workshop on Multimedia Media Convergence: Models, Technologies and Applications, EG Multimedia'99, Milan, Italy, Sep. 7-8, 1999 http://egmm99.di.fct.unl.pt
194. 1999 Invited expert, by NSF, for Digital Library for Education (DLE) meeting, Washington, D.C., June 10, 1999
195. 1998- Member, Steering Committee, ACM Digital Libraries conf. series
196. 1998-2000 Member, Program Committee of 4th European Conf. for Digital Libraries, ECDL'2000, Lisbon, Sept. 18-21, 2000, http://www.bn.pt/org/agenda/ecdl2000/
197. 1998-99 Member, Program Committee for The Eighth International Conference on Information and Knowledge Management (CIKM'99), Kansas City, Missouri, Nov. 2-6, 1999, http://www.cs.umbc.edu/cikm
198. 1998-99 Member, Program Committee, Fifth International Workshop on Multimedia Information Systems (MIS'99), Palm Springs Desert Resorts area, CA, Oct. 21-23, 1999
199. 1998-99 Member, Program Committee, ICON'99, International IEEE Conf. on Computer Networks, Brisbane, Australia, Sept. 9-10, 1999
200. 1998-99 Member, Review Committee for ACM SIGIR'99, Aug.15-19, Berkeley,CA
201. 1997-99 Associate Program Chair, ACM Multimedia'99, 7th ACM Int'l Multi- media Conf., Nov. 1999, Orlando, http://www.acm.org/sigmm/MM99/
202. 1997-99 Member, Program Committee for CoLIS 3, Dubrovnik, May 1999
203. 1999 Chair of Technical Committee meeting, for Networked Digital Library of Theses and Dissertations, held before Coalition for Networked Information Spring Meeting, Renaissance Washington D.C. Hotel, April 26-27, 1999
204. 1999 Participant, NSF Digital Library Workshop on undergraduate education, Arlington, VA, Jan. 4-6, http://www.dlib.org/smete/
205. 1998-99 Member, ASIS 1999 Mid-Year Meeting program committee
206. 1996-99 Member, OCLC's Research Advisory Council
207. 1995-99 Member, Columbia U. Digital Library Advisory Committee
208. 1998 Participant, University of Missouri-Columbia's Delphi Study of Digital Libraries. For details of the study and report see http://www.coe.missouri.edu/~is334/projects/Delphi_DL/Default.htm
209. 1998 Participant, NSF Workshop on Digital National Library for Undergraduate Science, Mathematics, Engineering, and Technology Education, Arlington, VA, July 21-23, http://www.dlib.org/smete/
210. 1998 Chair of Steering Committee meetings, for Networked Digital Library of Theses and Dissertations, held at Coalition for Networked Information offices, Washington, D.C, March 27, 1998; Sept.25,1998
211. 1998 Chair of Technical Committee meeting, for Networked Digital Library of Theses and Dissertations, held before Coalition for Networked Information Spring Meeting, Arlington, VA, April 14, 1998
212. 1998 Participant, D-Lib Metrics Working Group Meeting, Stanford, CA, Jan. 7-8, 1998
213. 1997-98 Member, Program Committee for IEEE Multimedia Systems'98, IEEEMM98
214. 1997-98 Member, Program Committee for ACM SIGIR '98, Aug., Melbourne, Australia

215. 1997-98 Member, Program Committee for The Seventh International Conference on Information and Knowledge Management (CIKM'98), Nov. 1998
216. 1997-98 Associate Program Chair, ACM Multimedia '98, 6th ACM Int'l Multimedia Conf., 12-16 Sept. 1998, Bristol, UK http://www.acm.org/sigmm/MM98/
217. 1997-98 Member, Program Committee of 2nd European Conf. for Digital Libraries, Crete
218. 1997-98 Member, Program Committee, SICON '98, IEEE Int'l Conf. on Networks, 30 June - 3 July 1998, Singapore (and session chair)
219. 1997-98 Member, Program Committee for ACM Digital Libraries '98, Pittsburgh, PA, June 23-26, 1998, http://www.ks.com/dl98 (also, Workshop Chair, committee to select winner of the Vannevar Bush award)
220. 1997-98 Member, Program Committee for BAS98, The 3rd Symposium on Computer Networks, June 25-26, 1998, Dokuz Eylul University, Izmir, Turkey
221. 1997-98 Member, Program Committee for RIDE98, Eighth International Workshop on Research Issues in Data Engineering: Continuous-Media Databases and Applications, February 23-24, 1998, Orlando, Florida, Sponsored by IEEE Computer Society Technical Committee on Data Engineering
222. 1996-98 Member, Steering Committee, Monticello Electronic Library, a project of SURA and SOLINET (re TIAPP grant)
223. 1995-98 Member, ACM SIGIR Executive Committee (as Past Chair)
224. 1995-98 Member, ACM SIGIR Electronic Publishing Committee
225. 1993-98 Member, ACM SIGIR Education Committee
226. 1997 Chair, Nominations Committee, ACM SIGIR
227. 1997 Chair of Technical Committee meeting, for Networked Digital Library of Theses and Dissertations, held before Coalition for Networked Information Spring Meeting, Minneapolis, Oct. 26, 1997
228. 1997 US Representative, Workshop on Curricular Development on Information Management and Digital Libraries, Lisbon, Portugal, Oct. 3-4, 1997
229. 1997 Chair of Steering Committee meeting, for Networked Digital Library of Theses and Dissertations, held at Coalition for Networked Information offices, Washington, D.C, Sept. 19, 1997
230. 1997 Member, workshop on a Digital National Library for Science, Mathematics, Engineering and Technology, sponsored by National Research Council, Washington, D.C., August 7-8, 1997
231. 1996-97 Program committee member, First European Conference in Research and Advanced Technology for Digital Libraries, Pisa, Italy, 9/1-3/97
232. 1996-97 Program committee member, 20th Intern'l Conf. on R&D in Information Retrieval, SIGIR '97, July 27-31, 1997, Philadelphia; also chair of session on Image Retrieval; also on Best Paper Committee
233. 1996-97 Technical papers committee member, 2nd ACM Int'l Conf. on Digital Libraries, DL'97, July 21-23, 1997, Philadelphia
234. 1995-97 Program committee member, IEEE Multimedia Systems'97, the IEEE International Conference on Multimedia Computing and Systems (ICMCS), June 3-6, 1997, Ottawa, Canada
235. 1995-97 Program comm. member, 1997 International Conference on Software Engineering, ICSE 97, May 17-24, 1997, Boston, MA
236. 1996-97 Organizing comm. member, Workshop on R&D Opportunities in Federal Information Services, 5/13-15/97, Arlington, VA; also met 12/3-5/96
237. 1997 Chair of Technical Committee meeting, for National Digital Library of Theses and Dissertations, held before Coalition for Networked Information Spring Meeting, Crystal City, VA, 4/1-2/97
238. 1997 Invited attendee, Computing and Humanities workshop, Computer Science and Telecommunications Board, National Research Council, March 28, 1997, Washington, D.C.

239. 1997 Chair of Steering Committee meeting, for National Digital Library of Theses and Dissertations, held at Coalition for Networked Information offices, Washington, D.C, March 14, 1997
240. 1997 Invited attendee and co-chair of one of four working groups, NSF Workshop for a Research Initiative on Distributed Knowledge-Work Environments, Santa Fe, March 9-11, 1997
241. 1997 NSF Database and Expert Systems panel, Feb. 3, 1997
242. 1996-97 Program committee member, Multimedia Computing and Networking 1997 (MMCN97) Conference (sponsored by SPIE and IS&T), San Jose, Feb. 10-12, 1997.
243. 1996 Chair of Technical Committee meeting, for National Digital Library of Theses and Dissertations, held after Coalition for Networked Information Fall Meeting, San Francisco, Dec. 7, 1996
244. 1996 Program review committee member, ACM Multimedia '96 conference, November 18-22, 1996, Boston, MA, http://www.acm.org/sigmm/MM96/
245. 1996 Member, NSF DLI Site Review Team, visiting 2 institutions, May, 1996
246. 1995-96 Steering committee member, NSF-sponsored 38th Allerton Institute, Oct. 27-29, 1996, Monticello, IL; co-chaired session with G. Marchionini
247. 1995-96 Program committee member, 19th Intern'l Conf. on R&D in Information Retrieval, SIGIR '96, Aug. 18-22, 1996, Zurich, Switzerland
248. 1995-96 Technical Program Committee member, HPDC Focus Workshop on Multimedia and Collaborative Environments, part of Fifth IEEE Int. Symp. on High Performance Distributed Computing (HPDC-5), Aug. 6-9, 1996, Syracuse, NY
249. 1995-96 Program committee member for Multimedia Computing and Networking 1996 - part of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Jan. 29-31, 1996, San Jose, CA
250. 1995-96 Program committee member, Pacific Workshop on Distributed Multimedia Systems, DMS'96, June 25-28, 1996, Hong Kong
251. 1995 Steering committee member, NSF-sponsored 37th Allerton Institute, "How We Do User-Centered Design and Evaluation of Digital Libraries: A Methodological Forum," 10/29-31/95, Monticello, IL
252. 1995 Program committee member, Curriculum Development in Computer Information Science: A Framework for Developing a New Curriculum in IR, workshop following ACM SIGIR'95, July 13, 1995, Seattle, WA
253. 1995 Invited participant, ARPA sponsored HPCC/IITA DigLib Workshop, May 18-19, 1995, Tysons Corner, VA.
254. 1995 Organizing Committee, NSF hosted workshop on Computer Science Technical Reports, NSF, Arlington, VA, April 7-8, 1995
255. 1994-96 Program committee member, ISMIS'96, organized with the assistance of the Polish Academy of Sciences, 6/10-13/96, Zakopane, Poland
256. 1994-95 Senior Member of the Program Committee, member of conference executive committee, ACM Multimedia '95, Third ACM International Conference on Multimedia, Nov. 5-9, 1995, San Francisco, CA
257. 1994-95 Program committee member, CIKM 95, Fourth International Conference on Information and Knowledge Management, Nov. 28 - Dec. 2, 1995, Baltimore, MD
258. 1994-95 Organizing Committee Member, Digital Libraries '95, June 11-13, 1995, Austin, TX
259. 1994-95 Program committee member, Electronic Publishing and the Information Superhighway, DAGS'95 (Dartmouth Institute for Advanced Graduate Studies 4th Annual Summer Meeting), May 31- June 2, Boston
260. 1994-95 Program committee member, 1995 Pacific Workshop on Distributed Multimedia Systems, March 30-31, 1995, Honolulu, Hawaii
261. 1994-95 Member, IBM Digital Library Council
262. 1994 Participant, Oct. 31, 1994 NSF Workshop on World-Wide Web in support of Computer Science Research and as a Means for NSF to Disseminate Information, NSF Headquarters, Arlington, VA
263. 1994 Assoc. Program Chair, Digital Libraries '94, June 19-21, 1994, College Station, TX.

264. 1994 Session Chair on Architectures, EG-MM '94, Eurographics Symposium and Workshop on Multimedia: Multimedia/Hypermedia in Open Distributed Environments, June 6-9, 1994, Graz, Austria. Also, invited workshop participant.
265. 1994 Member, ACM Electronic Membership Directory task force
266. 1994 NSF Instrumentation and Laboratory Improvement Program panel, Jan. 19-22, 1994.
267. 1994 NSF Database and Expert Systems panel, March 29-30.
268. 1993-94 Program vice-chair, CIKM 94, Third International Conference on Information and Knowledge Management, Nov. 29-Dec. 2, 1994, NIST, Gaithersburg, MD
269. 1993-94 Program and conference executive committees member, ACM Multimedia 94, Second ACM International Conference on Multimedia, Oct. 15-20, 1994, San Francisco, CA
270. 1993-94 Program committee member, 2nd International Workshop on Advanced Teleservices and High-Speed Communication Architectures (IWACA), Sept. 26-28, 1994, Heidelberg, Germany
271. 1993-94 Member, NSF Information Engineering Task Force, initial meeting Nov. 4-6, 1993, Alexandria, VA, preparing NSF Task Force Report on Developing a Framework for Academic Programs in Informatics: Educating the Next Generation of Information Specialists
272. 1993-94 Organizing committee member, Workshop on Intelligent Access to On-line Digital Libraries, in connection with IEEE CAIA '94, San Antonio, TX, March 1, 1994
273. 1993 NSF panels on Jan. 7-8, March 19, April 15, May 7, June 21-22, 1993
274. 1993 NIH RADIX review panel, March 26, Bethesda, MD
275. 1992-94 Program committee member, 17th Intern'l Conf. on R & D in Information Retrieval, SIGIR '94, Dublin, Ireland, July, 1994
276. 1992-94 Program committee member, IS&T/SPIE Symposium on Electronic Imaging Science and Technology --- two workshops: 1) Digital Video Compression and Processing on Personal Computers: Algorithms and Technologies, 2) High-Speed Networking and Multimedia Computing (and session chair). Feb. 6-10, 1994, San Jose, CA
277. 1992-94 Member, ACM Electronic Publishing Volunteer Advisory Committee
278. 1992-93 Member, Design Team, The Capture and Storage of Electronic Theses and Dissertations, sponsored by Coalition for Networked Information (CNI), VPI&SU, Council of Graduate Schools, and University Microfilms Int., including Information Technology group leader for 11 Oct. 1992 Design Meeting, Washington, D.C., speaker at 2 sessions of the CNI 1993 Spring Meeting, March, San Francisco.
279. 1992-93 Program committee member, ACM Multimedia '93, First ACM International Conference on Multimedia, Aug. 2-6, 1993, Anaheim, CA
280. 1992-93 Editorial board, NSF funded SCID3B (Scientific Database Bulletin Board)
281. 1992 Steering committee member, Hyperbases workshop, sponsored by the National Science Foundation and Texas A&M, 15-16 Oct., 1992, Washington, D.C.
282. 1992 Invited member, NSF Workshop on Visual Information Management Systems, org. by U. of Mich. and IBM Almaden Res. Cntr., 24-24 Feb. 1992, San Francisco.
283. 1992 Review panel member, NSF CISE Education Infrastructure Program
284. 1991-95 Jury, ASIS (Amer. Soc. for Inf. Science) Award for Research in Information Science: member 1991-93, chair 1994, member 1995
285. 1991-92 Program comm. member, 1992 Symposium on Applied Computing (SAC '92)
286. 1991 Review panel member, NSF Research Initiation Awards
287. 1991 Steering committee member, "Emerging Technologies" panel chair: Directions in Text Analysis, Retrieval and Understanding workshop, sponsored by the National Science Foundation and U. Nebraska, 11-12 Oct., 1991, Chicago IL.
288. 1991 Invited workshop member, "User Interfaces for Online Public Access Catalogs" Invitational Research Workshop, sponsored by the U. of Maryland Human-Computer Interaction Lab. and the Library of Congress Information Technology Office, Nov. 1, 1991, Library of Congress, Washington, D.C.

289. 1990-93 Member, Chemical Abstracts Service Research Council
290. 1990-92 Program committee member, 15th Intern'l Conf. on R & D in Information Retrieval, SIGIR '92, June 21-24, 1992, Copenhagen, Denmark
291. 1990-92 Member, Working Group on Hypermedia, Text Encoding Initiative
292. 1990-91 Member, Editorial Board, Applied Computing Review
293. 1990-91 Member, ACM Press CD-ROM Advisory Panel
294. 1990-91 Program committee member, publicity chair, 14th Intern'l Conf. on R & D in Information Retrieval, SIGIR '91, October 13-16, 1991, Chicago
295. 1990-91 Organizing committee member, AAAI-91 Workshop on Natural Language Text Retrieval, July 15, 1991, Anaheim, CA
296. 1990-91 Program committee member, RIAO (Recherche d'Informations Assistee par Ordinateur) 91, April 2-5, 1991, University Autonoma of Barcelona
297. 1990 Invited participant at Scientific Database Workshop, sponsored by NSF and Univ. of Virginia Dept. of Comp. Sci., Charlottesville VA, Mar. 12-13, 1990.
298. 1990 Organizer and chairman of planning meeting on Electronic Submissions at ACM Computer Science Conference, Feb. 20, 1990, Washington, D.C.
299. 1989-91 Member of ad hoc committee of the ACL: ACL Data Collection Initiative
300. 1989-91 Program comm. member, International Conference on Multimedia Information Systems, Singapore, Jan. 16-18, 1991
301. 1989-91 Member, Research Committee, American Society for Information Science
302. 1988-90 Technical program comm. member, The Fifth Annual AI Systems in Gov. Conf., May, 1990, Washington, D.C.
303. 1988-90 Member Advisory Board, Lexical Research Program, Applied Information Technologies Research Center, Columbus, Ohio
304. 1988-90 Program committee member, 13th Intern'l Conf. on R & D in Information Retrieval, Sept. 5-7, 1990, Brussels, Belgium. Access Methods session chair.
305. 1988-89 Technical program comm. member, Intelligent Systems - Realizing the Payoff, AI Systems in Gov. Conf., March 27-31, 1989, Washington, D.C.
306. 1988 Invited participant in AITRC sponsored workshop on lexicons and text collections, April 20, 1988, Dublin, Ohio
307. 1988 Review panel member, NSF Science & Tech. Research Centers
308. 1987-94 Editorial board, CD-ROM Professional(formerly Laserdisc Professional)
309. 1987-89 Publicity chairman, program committee member, 1989 ACM SIGIR Intern'l Conf. on R & D in Information Retrieval, June 25-27, 1989, Cambridge, MA
310. 1987-88 Program committee, hypertext session chair: RIAO (Recherche d'Informations Assistee par Ordinateur) 88: User-Oriented Text and Image Handling, March 21-24, 1988, Cambridge, MA.
311. 1987 Invited participant in AAP/UMI Workshop on Dissertation Markup, April 15, 1987, Ann Arbor, MI
312. 1986-87 Planning committee for Workshop on Distributed Expert-Based Information Systems, March 5-7, 1987, New Brunswick, NJ
313. 1985 Conference Committee, Virginia Networking and Telecommunications Planning Conference, Oct. 15 -16, 1985, Fredericksburg, VA

UNIVERSITY SERVICE:

CURRENT:

1. 2015- Faculty Affiliate, Virginia Tech Center for Autism Research (VTCAR)

2. 2015- Affiliated Faculty Member, Global Change Center (GCC)
3. 2014- University Libraries' Dean's Advisory Committee (Library DAC)
4. 2011- Member, Discovery Analytics Center (VT DAC)
5. 1998- Director, Digital Library Research Laboratory, www.dlib.vt.edu
6. 1995- Founding Member of Center for Human-Computer Interaction (CHCI)

PRIOR:

1. 2009-2019 Member, Search Committee(s), Dept. of Computer Science
2. 2012-9 CS Mentoring Program Coordinator, Co-Coordinator
3. 2013-7 College of Engineering Faculty Organization Executive Committee, Member, and President 2015-2017
4. 2016-7 Member, College of Engineering Commencement Committee
5. 2007-2014 Faculty Advisor to Information Technology, special staff member reporting to VP for Information Technology
6. 2009-2012 Member, Personnel Committee, Dept. of Computer Science
7. 2007-2010 Member, Commission on Undergraduate Studies and Policies (CUSP)
8. 2007-2010 Member, Committee on Undergraduate Curricula (CUC)
9. 2007-2010 Member, College of Engineering Undergraduate Curricular Committee
10. 2003-2009 Chair, Undergraduate Program Committee, Dept. of Computer Science
11. 2007-2008 Chair, Search Committee for next Head of Dept. of Computer Science
12. 2005-2007 Member, Personnel Committee, Dept. of Computer Science
13. 2005 Member, Review Committee for Dean of the College of Business
14. 2003-2004 Member, College of Engineering Undergraduate Curriculum Committee
15. 2002-2004 Member, Information Sciences and Technology Advisory Committee of the Administrative Coordinating Council for Engineering, Physical Sciences, and Information Sciences and Technology (ACCEPSIST)
16. 1998-2005 Director, Internet Technology Innovation Center at Virginia Tech (a University Center), http://fox.cs.vt.edu/itic/, www.internettic.org
17. 1996-2003 Member, Electronic Thesis and Dissertation Advisory Committee
18. 2003-2004 Member, College of Engineering Executive Committee
19. 2000-2003 Member, Undergraduate Program Committee, Dept. of Computer Science
20. 2001-2002 Member, University Council
21. 2000-2001 Fellow, Institute for the Social Assessment of Information Technology (ISAIT). Virginia Tech, http://www.uap.vt.edu/classes/uap3344/ISAITInfo/
22. 1996-2002 Chair, Commission on University Support (2001-2002, Vice chair 2001, member in previous years)
23. 2001 Member, Technical Advisory Group for WPI, meeting 1/19/2001
24. 1999-00 Member, Review Committee for Center for Wireless Telecommunications (CWT)
25. 1997- Member, Personnel Committee of Dept. of Computer Science
26. 1995-99 Member, University Computing and Communications Resources Committee (chair for AY 1995-1996)
27. 1995-99 Coordinator, College of Arts and Sciences, Advanced Communication and Information Technology Center (ACITC)
28. 1997-98 Member, University Libraries Serials Collection Committee
29. 1996-98 Member, Steering Committee, University Self Study
30. 1996-98 Member, 125th Anniversary Steering Committee

31. 1996-98 Chair, Traditional Students Subcommittee, University Self Study
32. 1995-97 Member, Computing Resources Committee of Dept. of Computer Science
33. 1994-97 Member, Faculty Senate (and chair, 1995-96, of working group on distance learning)
34. 1994-95 Member, University Research Computing Advisory Group
35. 1994-95 Member, Blacksburg Electronic Village Research Advisory Group
36. 1992-95 Member, Executive Committee of Department of Computer Science
37. 1992-93 Faculty advisor, Organization for Student Computing
38. 1991-93 Chair, Search Committee, for Head of Department of Computer Science
39. 1991 Member, Search Committee for Asst. Dean for Engineering Computing
40. 1991 Member, Planning Comm. for 2nd Annual University/Industry Cooperation Conf.
41. 1991 Member, Task Group for Innovations in Teaching: The Information Resources
42. 1990-94 Member (for Arts & Sciences), University's Library Committee
43. 1990-92 Chair, Arts & Sciences' Ad Hoc College Library Committee
44. 1989-93 Member, Scholarly Publications Project subcommittee
45. 1989-92 Chair, Dept. of Computer Science's Computing Resources Committee
46. 1988-95 Member (for Arts & Sciences), University's Communications Resources Committee
47. 1988-89 Member, Dept. of Comp. Science's Graduate Program & Research Activities Comm.
48. 1987-91 Departmental representative, Sigma Xi
49. 1987-90 Member, Arts & Sciences' Ad Hoc College Library Committee
50. 1987-88 Member, PC Selection Committee (selecting CS computer system for requirement)
51. 1987-88 Faculty Counselor for University's Honor System
52. 1983-89 Member, Dept. of Computer Science's Computing Resources Committee, wrote first draft of request for proposal leading to 1985 UNIX requirement for CS students
53. 1986-87 Member, LRC Teaching Learning Grant Committee
54. 1986-87 Member, Computing and Information Services Committee of University Self Study (serving on 2 of the 4 task forces, responsible for helping prepare final report)
55. 1986 Member, review team, CNS Single System Image project
56. 1986 Member, Task Force, Library's role in relation to other units in an electronic campus
57. 1985 Member, Ad Hoc Committee on Library Futures
58. 1983-87 Technical liason for Virginia Tech to CSNET (proposed joining the Computer Science Network, and coordinated this effort that was taken over in 1987 by CNS, leading to university Internet connection)

SELECTED COURSES TAUGHT:

- **Regular:** Data Structures; Information Storage and Retrieval; Information Systems Project; Intro. to Artificial Intelligence; Intro. to Computer Science; Intro. to Data Base Management Systems, Intro. to LIKES (Living In the KnowlEdge Society), Intro. to Networked Information (team); Multimedia, Hypertext and Information Access; Multimedia Technology and Projects (team)
- **Special Topics:** Big Data Text Summarization, CD-ROM Publishing and Access, Computational Linguistics, Digital Libraries, Intelligent Multimedia, Information Systems, Interactive Accessibility, Natural Language Processing, Office Information Systems, UNIX/C
- **VTechWorks Submissions from Recent Courses (team term projects):**

- CS4624 (Multimedia, Hypertext, and Information Access): https://vtechworks.lib.vt.edu/handle/10919/18655
- CS4984 (Computational Linguistics, Big Data Text Summarization): https://vtechworks.lib.vt.edu/handle/10919/50956
- CS5604 (Information Retrieval): https://vtechworks.lib.vt.edu/handle/10919/19081
- CS6604 (Digital Libraries): https://vtechworks.lib.vt.edu/handle/10919/47780

PUBLICATIONS:

(with h-index of 60 and i10-index of 254, with 18229 citations, according to Google Scholar)

CD-ROMS (in quantity, among earliest manufactured in Virginia):

1. E. Fox, editor and project manager. Virginia Disc One. Produced by Nimbus Records, 1990. Blacksburg VA: VPI&SU Press. Contains about 600 megabytes in about 6000 files of data and DOS executable programs. ISBN 0-929900-00-6.
2. E. Fox, editor and project manager. Virginia Disc Two. Produced by Nimbus Records, 1990. Blacksburg VA: VPI&SU Press. Contains 2 searchable library catalog collections accessible with MicroVTLS and Personal Librarian. ISBN 0-929900-02-2.
3. E. Fox, editor and project manager. Virginia Disc Three. Produced by Nimbus Records, 1990. Blacksburg VA: VPI&SU Press. Contains about 300 megabytes of data and software of interest for Macintosh systems running the A/UX version of UNIX. ISBN 0-929900-01-4.

VIDEOTAPES:

1. E. Fox, executive producer, script writer. Interactive Digital Video. Five minute preview (Aug. 1989), hour final program (Jan. 1990), supplementing July 1989 Communications of the ACM, produced for ACM Press Database and Electronic Products with assistance from IBM T. J. Watson Research Center (with producer/director Guillermo Pulido).
2. E. Fox, executive producer. Advanced Naval Message Analyzer. Fifteen minute (Nov. 1990) VPI&SU videotape description of results for NSWC contract about the CODER system, producer/director Jim Ruggiero, host/script/graphics Richard Barnhart.

BOOKS:

1. General Chairs: Jiangping Chen, Marcos Andre Goncalves, Jeff M. Allen; Program Chairs: Edward A. Fox, Min-Yen Kan, Vivien Petras. 2018. JCDL '18 The 18th ACM/IEEE Joint Conference on Digital Libraries. Fort Worth, TX, USA - June 03 - 07, 2018. Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries. ACM, New York, NY, USA. ISBN 978-1-4503-5178-2. 435 pages, https://dl.acm.org/citation.cfm?id=3197026
2. Edward A. Fox and Jonathan P. Leidig, editors. Digital Library Applications: CBIR, Education, Social Networks, eScience/Simulation, and GIS. Morgan & Claypool Publishers, San Francisco, March 2014, 175 pages, ISBN paperback 9781627050326, ebook 9781627050333, http://dx.doi.org/10.2200/S00565ED1V01Y201401ICR032, Synthesis Lectures on Information Concepts, Retrieval, and Services 6 (1), 1-175
3. Edward A. Fox and Ricardo da Silva Torres, editors. Digital Library Technologies: Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security. Morgan & Claypool Publishers, San Francisco, March 2014, 205 pages, ISBN paperback 9781627050302, ebook 9781627050319, http://dx.doi.org/10.2200/S00566ED1V01Y201401ICR033

4. Rao Shen, Marcos Andre Goncalves, and Edward A. Fox. Key Issues Regarding Digital Libraries: Evaluation and Integration. Morgan & Claypool Publishers, San Francisco, Feb. 2013, 110 pages, ISBN paperback 9781608459124, ebook 9781608459131, http://dx.doi.org/10.2200/S00474ED1V01Y201301ICR026
5. Edward A. Fox, Marcos Andre Goncalves, and Rao Shen. Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach. Morgan & Claypool Publishers, San Francisco, July 2012, 180 pages, ISBN paperback 9781608459100, ebook 9781608459117, http://dx.doi.org/10.2200/S00434ED1V01Y201207ICR022, supplementary website https://sites.google.com/a/morganclaypool.com/dlibrary/
6. Edward A. Fox, editor. Digital Libraries: A 5S Approach. With its 15 chapters authored with: Monika Akbar, Pranav Angara, Yinlin Chen, Lois M. Delcambre, Noha Elsherbiny, Eric Fouh, Marcos Andre Goncalves, Nadia P. Kozievitch, Spencer Lee, Jonathan Leidig, Lin Tzy Li, Mohamed Magdy Gharib Farag, Uma Murthy, Sung Hee Park, Rao Shen, Venkat Srinivasan, Ricardo da Silva Torres, and Seungwon Yang. 527 page draft textbook, used in VT's Digital Libraries class (CS6604) in Fall 2011, released selectively for review and testing (including in a Villanova DL course), most recent version Feb. 8, 2012
7. Philip S. Yu, Vassilis Tsotras, Edward Fox, and Bing Liu (eds.). Proceedings 15th ACM Conference on Information and Knowledge Management (CIKM 2006), November 6-11, 2006, Arlington, VA, USA, ACM Press, ISBN 1-59593-433-2
8. Fox, E.A.; Neuhold, E.; Premsmit, P.; Wuwongse, V. (Eds.) Digital Libraries: Implementing Strategies and Sharing Experiences; Proceedings 8th International Conference on Asian Digital Libraries, ICADL 2005, Bangkok, Thailand, December 12-15, 2005, Lecture Notes in Computer Science, Vol. 3815, XVII, 529 p., ISBN: 3-540-30850-4, DOI: 10.1007/11599517
9. Zhaoneng Chen, Hsinchun Chen, Qihao Miao, Yuxi Fu, Edward Fox, Ee-Peng Lim, eds. Digital Libraries: International Collaboration and Cross-Fertilization; Proceedings 7th International Conference on Asian Digital Libraries, ICADL 2004, Shanghai, Dec. 2004. Springer, Lecture Notes in Computer Science 3334, ISBN 3-540-24030-6.
10. Edward A. Fox, Shahrooz Feizbadi, Joseph M. Moxley, and Christian R. Weisser, eds., Electronic Theses and Dissertations: A Sourcebook for Educators, Students, and Librarians, New York: Marcel Dekker, April 2004, ISBN 0-8247-0973-X
11. Ee-Peng Lim, Schubert Foo, Chris Khoo, Hsinchun Chen, Edward Fox, Shalini Urs, Thanos Constantino, eds. Digital Libraries: People, Knowledge, and Technology; Proceedings 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Dec. 2002. Springer, Lecture Notes in Computer Science 2555.
12. L. Kalinichenko (coordinating author), with working group (D. Atkins, E. Fox, D. Fulker, Y. Ioannidis, S. Koernig, B. Ludascher, M. Marlino, P. Savino, N. Seshagiri, T. Sumner, M. Wright). Digital Libraries in Education: Analytical Survey. UNESCO Institute for Information Technologies in Education, 2003, Moscow, 68 pages, http://www.iite.ru/img/upload/Digital_Libraris.pdf
13. Joseph M. Moxley, Diane Masiello, and E. Fox, eds., The Guide for Electronic Theses and Dissertations, over 425 pages, completed in English in 2001, translated into French and Spanish in 2002, and in process for later translation into a number of other languages, funded in part and made available by UNESCO as book and web site (www.etdguide.org), 2001-2002; in 2008 updated and converted for inclusion in Wikibooks at http://en.wikibooks.org/wiki/ETD_Guide
14. C. Borgman and E. Fox. Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL'2001, Hotel Roanoke, Roanoke, VA, June 24-28, 2001
15. E. Fox, N. Rowe, and D. Watkins, eds., Proceedings of the Fourth ACM International Conference on Digital Libraries, DL'99, Berkeley, CA, Aug. 11-14, 1999.
16. A. Cardenas, W. Chu, and E. Fox, eds. Proceedings of the NSF Information and Data Management Workshop, UCLA, Los Angeles, March 7-9, 1999.
17. E. Fox and G. Marchionini, eds., Proceedings of the First ACM International Conference on Digital Libraries, DL'96, Bethesda, MD, March 20-23, 1996.

18. E. Fox, P. Ingwersen, R. Fidel, eds., Proceedings SIGIR'95, 18th International Conference on Research and Development in Information Retrieval, New York: ACM Press, 1995.
19. E. Fox, ed. Sourcebook on Digital Libraries: Report for the National Science Foundation, TR-93-35, VPI&SU Computer Science Dept., Dec. 1993, Blacksburg, VA. Over 400 pages, http://fox.cs.vt.edu/DigitalLibrary/

JOURNAL/MAGAZINE ARTICLES:

1. Liuqing Li, Jack Geissinger, William A. Ingram, Edward A. Fox. Teaching Natural Language Processing through Big Data Text Summarization with Problem-Based Learning. Data and Information Management, ISSN:2543-9251, 4(1): 18-43, March 24, 2020, open access, https://doi.org/10.2478/dim-2020-0003, https://content.sciendo.com/downloadpdf/journals/dim/4/1/article-p18.xml
2. Niu, Shuo; McCrickard, D. Scott; Nguyen, Julia; Haqq, Derek; Kotut, Lindah; Stelter, Timothy L. and Fox, Edward A.; "Investigating Paradigms of Group Territory in Multiple Display Environments", PACM (Proceedings of the ACM) on Human-Computer Interaction, Vol. 4 (GROUP, Article 13), Jan. 2020, https://doi.org/10.1145/3375193
3. Ingram, William A., Bipasha Banerjee, and Edward A. Fox. "Summarizing ETDs with deep learning." Cadernos BAD 1 (2020): 46-52
4. Zach, Florian J., Ma, Yufeng, Fox, Edward A. (2019). A Preliminary Analysis of Images in Online Hotel Reviews. e-Review of Tourism Research (eRTR) http://ertr.tamu.edu, 16(2/3), 156-164. https://journals.tdl.org/ertr/index.php/ertr/article/view/328
5. Hamed Alhoori, Mohammed Samaka, Richard Furuta, and Edward A. Fox. Anatomy of Scholarly Information Behavior Patterns in the Wake of Academic Social Media Platforms. International Journal on Digital Libraries, 2018, https://doi.org/10.1007/s00799-018-0255-9
6. S.M.Shamimul Hasan, Edward A. Fox, Keith Bisset, Madhav V. Marathe. EpiK: A Knowledge Base for Epidemiological Modeling and Analytics of Infectious Diseases. J Healthcare Informatics Research (Dec. 2017) 1(2): 260-303. DOI: 10.1007/s41666-017-0010-9, https://doi.org/10.1007/s41666-017-0010-9
7. Andrea Kavanaugh, Steven D. Sheetz, Hamida Skandrani, and Edward A. Fox. Media use by young Tunisians during the 2011 revolution vs 2014 elections. Information Polity 22(2-3): 137-158 (Oct. 20, 2017), DOI 10.3233/IP-170412
8. Zhiwu Xie and Edward A. Fox. Advancing library cyberinfrastructure for big data sharing and reuse. Information Services & Use, vol. 37, no. 3, pp. 319-323, 7 Nov. 2017. https://content.iospress.com/articles/information-services-and-use/isu853. Also in Proc. NFAIS 2017, Alexandria, VA, USA, February 26-28, 2017. DOI: 10.3233/ISU-170853
9. Edward A. Fox, Zhiwu Xie, Martin J. Klein. Web Archiving and Digital Libraries (WADL) 2016: Highlights and Introduction to this Special Issue. Bulletin of IEEE Technical Committee on Digital Libraries, 13(1), April 2017, 3 pages, http://www.ieee-tcdl.org/Bulletin/v13n1/papers/intro.pdf
10. Yinlin Chen, Zhiwu Xie, and Edward A. Fox. A Library to Manage Web Archive Files in Cloud Storage. Bulletin of IEEE Technical Committee on Digital Libraries, 13(1), April 2017, 1 page, http://www.ieee-tcdl.org/Bulletin/v13n1/papers/chen.pdf
11. Mohamed Farag and Edward A. Fox. Which webpage should we crawl first? Social media-based webpage source importance guidance. Bulletin of IEEE Technical Committee on Digital Libraries, 13(1), April 2017, 1 page, http://www.ieee-tcdl.org/Bulletin//v13n1/papers/farag.pdf
12. Sunshin Lee and Edward A. Fox. Archiving and Analyzing Tweets and Webpages with the DLRL Hadoop Cluster. Bulletin of IEEE Technical Committee on Digital Libraries, 13(1), April 2017, 1 page, http://www.ieee-tcdl.org/Bulletin//v13n1/papers/lee.pdf
13. Edward A. Fox, Martin Klein, and Zhiwu Xie. Guest Editors' Introduction to the Special Issue on Web Archiving. International Journal on Digital Libraries, 18, 2017. DOI: 10.1007/s00799-016-0203-5; also on paper: 19(1): 1-2
14. Mohamed Magdy Gharib Farag, Sunshin Lee, Edward A. Fox. Focused Crawler for Events. International Journal on Digital Libraries, 18:1-17, 2017. DOI: 10.1007/s00799-016-0207-1

15. Warren Kurt Bickel, Amanda Quisenberry, Prashant Chandrasekar, Mikhail Nikolaas Koffarnus, Edward A. Fox, Chris Franck. The social interactome of recovery: Network topology influences social media engagement. Drug and Alcohol Dependence vol. 171, page e20, 2017. http://dx.doi.org/10.1016/j.drugalcdep.2016.08.070
16. Andrea L. Kavanaugh, Steven D. Sheetz, Rodrigo Sandoval-Almazan, John C. Tedesco, Edward A. Fox. Media use during conflicts: Information seeking and political efficacy during the 2012 Mexican elections. Government Information Quarterly, 33(3): 595-602, Feb. 2016, DOI: 10.1016/j.giq.2016.01.004, http://dx.doi.org/10.1016/j.giq.2016.01.004
17. Tarek Kanan, Raed Kanaan, Omar Al-Dabbas, Ghassan Kanaan, Ali Al-Dahoud, Edward Fox. Extracting Named Entities Using Named Entity Recognizer for Arabic News Articles. International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE) 5(11):78-84, 11/30/2016.
18. Tarek Kanan and Edward A. Fox. Automated Arabic Text Classification with P-Stemmer, Machine Learning, and a Tailored News Article Taxonomy. Journal of the Association for Information Science and Technology (JASIST), 67(11): 2667-2683, Nov. 2016, published online 23 Dec. 2015, DOI: 10.1002/asi.23609.
19. Zhiwu Xie, Edward A Fox, Tyler Walters, Pablo Tarazaga, and Jiangping Chen. “Developing Library Cyberinfrastructure (LCI) Strategy for Big Data Sharing and Reuse,” D-Lib Magazine, September/October, 2016.
20. Richard Gruss, Tarek Kanan, Xuan Zhang, Mohamed Farag, Mary C. English, and Edward A. Fox. Teaching Big Data Through Project-based Learning in Computational Linguistics and Information Retrieval. Journal of Computing Sciences in Colleges (ISSN 1937-4771) 31(2): 260-270. From Proc. 29th Annual Consortium for Computing in Small Colleges: Southeastern Conference (CCSC:SE), Roanoke College, Salem, VA, November 6-7, 2015.
21. Yinlin Chen and Edward A. Fox. Extending Ensemble - An Education Digital Library for Computer Science Education. Journal of Computing Sciences in Colleges (ISSN 1937-4771) 31(2): 201-207. From Proc. 29th Annual Consortium for Computing in Small Colleges: Southeastern Conference (CCSC:SE), Roanoke College, Salem, VA, November 6-7, 2015.
22. Edward A. Fox, Zhiwu Xie, Martin J. Klein. Introduction to the Web Archiving and Digital Libraries 2015 Workshop Issue: Web Archiving and Digital Libraries 2015 (WADL 2015) Overview. Bulletin of IEEE Technical Committee on Digital Libraries, 11(2), October 2015, 2 pages, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/intro.pdf
23. Mohamed M. G. Farag, Edward A. Fox. Building and archiving event web collections: A focused crawler approach. Bulletin of IEEE Technical Committee on Digital Libraries, 11(2), October 2015, 2 pages, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/farag.pdf
24. Tarek Kanan, Sagnik Ray Choudhury, C. Lee Giles, Prashant Chandresekar, Edward A. Fox. Digital Library and Archiving for Qatar. Bulletin of IEEE Technical Committee on Digital Libraries, 11(2), October 2015, 1 page, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/kanan.pdf
25. Zhiwu Xie, Prashant Chandresekar, Edward A. Fox. A UWS Case for 200-Style Memento Negotiations. Bulletin of IEEE Technical Committee on Digital Libraries, 11(2), October 2015, 1 page, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/xie2.pdf
26. Sanghee Oh, Seungwon Yang, Jeffrey A. Pomerantz, Barbara M. Wildemuth, and Edward A. Fox. Results of a Digital Library Curriculum Field Test. International Journal on Digital Libraries, 17(4):273-286, 2016. Published online as open access work, 20 May 2015, DOI: 10.1007/s00799-015-0151-5
27. Jonathan P. Leidig and Edward A. Fox. Intelligent Digital Libraries and Tailored Services. Journal of Intelligent Information Systems, Nov. 2014, 1-18, http://dx.doi.org/10.1007/s10844-014-0342-3
28. M. Tabet, T. Kanan, H. Alhoori, S. S. Lukesh, C. Thompson, E. A. Fox, and M. Samaka, "Digital library initiative: a project for educators in Qatar," ACM Inroads, 4(3): 70-75, 2013
29. Andrea Kavanaugh, Steven D. Sheetz, Riham Hassan, Seungwon Yang, Hicham G. Elmongui, Edward A. Fox, Mohamed Magdy, and Donald J. Shoemaker. Between a Rock and a Cell Phone: Communication and Information Technology Use during the 2011 Uprisings in Tunisia and Egypt. Special issue on social media, Int'l J. of Information Systems for Crisis Response and Management (IJISCRAM), 5(1): 1-21, January - March 2013

30. Uma Murthy, Edward A. Fox, Yinlin Chen, Eric M. Hallerman, Donald J. Orth, Ricardo da S. Torres, Lin Tzy Li, Nadia P. Kozievitch, Felipe S.P. Andrade, Tiago R.C. Falcao & Evandro Ramos. SuperIDR: A Tool for Fish Identification and Information Retrieval. Fisheries 38(2): 65-75, Feb. 2013
31. Kozievitch, N. P.; Almeida, J. G. A.; Torres, R. da S.; Santanche, A.; Leite, N. J.; Murthy, U.; Fox, E.A. Reusing a Compound-Based Infrastructure for Searching and Annotating Video Stories. International Journal of Multimedia Technology, 2(3): 89-97, Sept. 2012, http://www.ijmt.org/paperInfo.aspx?ID=57
32. Andrea L. Kavanaugh, Edward A. Fox, Steven D. Sheetz, Seungwon Yang, Lin Tzy Li, T. Whalen, Donald J. Shoemaker, Apostol Natsev, Lexing Xie. Social Media Use by Government: From the Routine to the Critical. Government Information Quarterly (GIQ) 29(4): 480-491, Oct. 2012, http://dx.doi.org/10.1016/j.giq.2012.06.002
33. Sourav Mishra, Kriti Sen Sharma, Spencer J. Lee, Edward A. Fox, and Ge Wang. SLATE: virtualizing multiscale CT training. Journal of X-Ray Science and Technology. 2012; 20(2):239-48. doi: 10.3233/XST-2012-0332.
34. Davinia Hernandez-Leo, Monika Akbar, Deborah Tatar, and Edward A. Fox. Social tools for educators: supporting the needs of specific communities. IEEE Learning Technology Newsletter 14(2), April 2012, special theme section on "Social Networks and Social Computing in Technology-Enhanced Learning", http://www.ieeetclt.org/issues/april2012/IEEE-LT-Apr2012.htm
35. N.J. Short, A. Lynn Abbott, M.S. Hsiao, E.A. Fox. Reducing descriptor measurement error through Bayesian estimation of fingerprint minutia location and direction. IET Biometrics, a journal of The Institution of Engineering and Technology, 1(1): 82-90, 2012. http://dx.doi.org/10.1049/iet-bmt.2011.0010
36. Edward A. Fox. Constructivist Learning with Modularization, Personalization, and Teams. In Pedagogy in Practice, a magazine of Virginia Tech's Center for Instructional Development and Educational Research (CIDER), Blacksburg, VA, Spring 2012, 14-15
37. Nadia P. Kozievitch, Jurandy Almeida, Ricardo da S. Torres, Neucimar A. Leite, Marcos A. Goncalves, Uma Murthy, Edward A. Fox. Towards a Formal Theory for Complex Objects and Content-Based Image Retrieval. Journal of Information and Data Management, 2(3): 321-336 (Oct. 2011), http://seer.lcc.ufmg.br/index.php/jidm/article/view/142
38. Nadia P. Kozievitch, Ricardo da S. Torres, A. Santanche, D.C.G. Pedronette, R.T. Calumby, Edward A. Fox. An Infrastructure for Searching and Harvesting Complex Image Objects. Information Interaction Intelligence Journal, 11(2): 39-68, 2011. http://www.irit.fr/journal-i3/volume11/numero02/
39. Delcambre, L. Archer, D., Price, S., Britell, S. Murthy, U., Fox, E. A., and Cassel, C. 2011. Superimposing a strand map over lectures and textbook content (in a database class). Journal of Computing Sciences in College. 27(1), 143-151.
40. Leonardo Candela, Donatella Castelli, Edward A. Fox and Yannis Ioannidis. On Digital Library Foundations. International Journal on Digital Libraries (IJDL) 11(1): 37-39, March 2010, http://www.springerlink.com/content/r467157673711028/fulltext.pdf
41. Jeffrey Pomerantz, Barbara M. Wildemuth, Sanghee Oh, Seungwon Yang, and Edward A. Fox. Evaluation of a Curriculum for Digital Libraries. Bulletin of IEEE Technical Committee on Digital Libraries (TCDL Bulletin), 5(1), Spring 2009, http://www.ieee-tcdl.org/Bulletin/v5n1/Pomerantz/pomerantz.html
42. Venkat Srinivasan, Edward A. Fox. Global Science and Technology Assessment by Analysis of Large Collections of Electronic Dissertations, in Hsinchun Chen, Ronald N. Kostoff, Chaomei Chen, Jian Zhang, Michael S. Vogeley, Katy Borner, Nianli Ma, Russell J. Duhon, Angela Zoss, Venkat Srinivasan, Edward A. Fox, Christopher C. Yang, Chih-Ping Wei, "AI and Global Science and Technology Assessment," IEEE Intelligent Systems, 24(4): 68-88, July/August, 2009 http://www.computer.org/portal/web/csdl/magazines/intelligent#4
43. Barbara L. Moreira, Marcos Andre Goncalves, Alberto H.F. Laender, and Edward A. Fox. Automatic Evaluation of Digital Libraries with 5SQual. Journal of Informetrics, 3(2): 102-123, April 2009, linked from http://www.sciencedirect.com/science/journal/17511577
44. R. S. Torres, A. Falcao, M. A. Goncalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval, Pattern Recognition, 42(2): 283-292, Feb. 2009, doi:10.1016/j.patcog.2008.04.010

45. Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, Edward A. Fox. Integration of Complex Archaeology Digital Libraries: An ETANA-DL Experience. Information Systems. 33(7-8): 699-723, Nov.-Dec. 2008, DOI: http://dx.doi.org/10.1016/j.is.2008.02.006
46. Edward A. Fox. Porque ETDs? Boletim de Teses e Dissertacoes Eletronicas 3, Numero 3, 17 Nov. 2008, http://www.maxwell.lambda.ele.puc-rio.br/index.php?msg=22
47. W. Ryan Richardson, Venkat Srinivasan, and Edward A. Fox. Knowledge Discovery in Digital Libraries of Electronic Theses and Dissertations: An NDLTD Case Study. Int. J. Digital Libraries, 9(2):163-171, Nov. 2008, http://www.springerlink.com/content/w3182840w7j17117/
48. Barbara M. Wildemuth, Jeffrey Pomerantz, Sanghee Oh, Seungwon Yang, and Edward A. Fox. A Digital Libraries Curriculum: Expert Review and Field Testing. D-Lib Magazine, 14(7/8), July/August 2008, doi:10.1045/july2008-inbrief
49. Heines, J. M., Goldman, K. J., Jeffers, J., Fox, E. A., and Beck, R. 2008. Interdisciplinary approaches to revitalizing undergraduate computing education. J. Comput. Small Coll. 23, 5 (May. 2008), 68-72.
50. Marcos Andre Goncalves, Edward A. Fox, and Layne T. Watson. Towards a Digital Library Theory: A Formal Digital Library Ontology. Int. J. Digital Libraries 8(2): 91-114, April 2008, DOI: 10.1007/s00799-008-0033-1
51. Edward A. Fox, Christopher Andrews, Weiguo Fan, Jian Jiao, Ananya Kassahun, Szu-Chia Lu, Yifei Ma, Chris North, Naren Ramakrishnan, Angela Scarpa, Bruce H. Friedman, Steven D. Sheetz, Donald Shoemaker, Venkat Srinivasan, Seungwon Yang, & Laura Boutwell, A Digital Library for Recovery, Research, and Learning from April 16, 2007 at Virginia Tech. Traumatology, 14(1): 64-84 (2008), DOI: 10.1177/1534765608315632
52. Donatella Castelli & Edward Fox. (2007). In Brief: Series of Workshops on Digital Library Foundations. D-Lib Magazine 12(9). http://www.dlib.org/
53. Marcos Andre Goncalves, Barbara L. Moreira, Edward A. Fox, and Layne T. Watson. "What is a good digital library?" - A quality model for digital libraries. Information Processing and Management, Volume 43, Issue 5, September 2007, Pages 1416-1437, http://dx.doi.org/10.1016/j.ipm.2006.11.010
54. Jeffrey Pomerantz, Sanghee Oh, Seungwon Yang, Edward A. Fox, and Barbara M. Wildemuth. The Core: Digital Library Education in Library and Information Science Programs. D-Lib Magazine, Nov. 2006 (Vol. 12 No. 11), http://www.dlib.org/dlib/november06/pomerantz/11pomerantz.html
55. Edward A. Fox, Seungwon Yang, Seonho Kim. ETDs, NDLTD, and Open Access: A 5S Perspective. Ciencia da Informacao (leading journal of library and information science in Brazil), 35(2):75-90, May/Aug. 2006, ISSN 0100-1965, Brasilia
56. Pomerantz, J., Wildemuth, B. M., Oh, S., Yang, S., & Fox, E. A. (2006). Briefing: Digital libraries curriculum development. D-Lib Magazine 12(7/8). http://www.dlib.org/dlib/july06/07inbrief.html#POMERANTZ
57. Edward A. Fox, A Forward-Looking European Digital Library? Hence 5S?. Invited article for ERCIM News, No. 66 (http://www.ercim.org/publication/Ercim_News/enw66/), July 2006, pp. 16-17
58. Byron B. Marshall, Hsinchun Chen, Rao Shen, and Edward A. Fox. Moving digital libraries into the student learning space: the GetSmart experience. ACM Journal on Educational Resources in Computing (JERIC), 6(1), 2006, http://doi.acm.org/10.1145/1217862.1217864 (March, but published late in 2006)
59. Edward A. Fox, Fernando Das Neves, Xiaoyan Yu, Rao Shen, Seonho Kim, and Weiguo Fan. Exploring the computing literature with visualization and stepping stones & pathways. Communications of the ACM (CACM), 49(4): 52-58, April 2006
60. Ricardo da S. Torres, Claudia Bauzer Medeiros, Marcos Andre Goncalves, and Edward A. Fox. A Digital Library Framework for Biodiversity Information Systems. International Journal on Digital Libraries, Special issue on Multimedia Contents and Management in Digital Libraries, 6(1): 3-17, Feb. 2006, ISSN: 1432-5012 (Paper) 1432-1300 (Online), http://dx.doi.org/10.1007/s00799-005-0124-1
61. Xiaoyan Yu, Fernando Das-Neves, and Edward A. Fox. Hard Queries can be Addressed with Query Splitting Plus Stepping Stones and Pathways. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 28(4): 29-38, December 2005, http://sites.computer.org/debull/A05dec/yu.pdf

62. Yannis Ioannidis, David Maier, Serge Abiteboul, Peter Buneman, Susan Davidson, Edward Fox, Alon Halevy, Craig Knoblock, Fausto Rabitti, Hans Schek and Gerhard Weikum. Digital library information-technology infrastructures. International Journal on Digital Libraries 5(4): 266-274, http://dx.doi.org/10.1007/s00799-004-0094-8
63. Edward A. Fox and Elisabeth Logan. An Asian Digital Libraries Perspective (intro. by guest editors to special issue with 8 other papers). Information Processing and Management, Jan. 2005, 41(1): 1-4.
64. Morgan, Eric Lease, Jeremy Frumkin, Edward A. Fox, "The OCKHAM Initiative - Building Component-based Digital Library Services and Collections", D-Lib Magazine, November 2004 (Vol. 10 No. 11). http://dlib.org/dlib/november04/11inbrief.html#FOX
65. S. Perugini, M. A. Goncalves, and E. A. Fox. Recommender Systems Research: A Connection-Centric Survey. Journal of Intelligent Information Systems, Vol. 23(2): 107-143, Sept. 2004.
66. Weiguo Fan, Edward A. Fox, Praveen Pathak, and Harris Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search, Journal of the American Society for Information Science and Technology (JASIST), 2004, 55(7): 628-636.
67. M. Goncalves, E. Fox, L. Watson, N. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. ACM Transactions on Information Systems, April 2004, 22(2): 270-312.
68. Hussein Suleman, Edward A Fox, Rohit Kelapure, Aaron Krowne, Ming Luo. Building digital libraries from simple building blocks, Online Information Review Volume 27 Number 5, Oct. 2003, 301-310, http://oberon.emeraldinsight.com/vl=1389995/cl=44/nw=1/rpsv/cgi-bin/linker?ini=emerald&reqidx=/cw/mcb/14684527/v27n5/s1/p301.
69. H. Suleman and E. Fox. Leveraging OAI Harvesting to Disseminate Theses. Library Hi Tech, 21(2): 219-227, 2003, issue table of contents: http://www.emeraldinsight.com/0737-8831.htm.
70. E. Fox, R. Moore, R. Larsen, S. Myaeng, S. Kim. Toward a Global Digital Library: Generalizing US-Korea Collaboration on Digital Libraries. D-Lib Magazine, Oct. 2002, 8(10), http://www.dlib.org/dlib/october02/fox/10fox.html
71. E. Fox, M. Goncalves, G. McMillan, J. Eaton, A. Atkins, and N. Kipp. The Networked Digital Library of Theses and Dissertations: Changes in the University Community. In special issue on "Information Technology and Educational Change" of The Journal of Computing in Higher Education, 13(2): 3-24, Spring 2002.
72. Hussein Suleman and Edward Fox. The Open Archives Initiative: Realizing Simple and Effective Digital Library Interoperability. In special issue on "Libraries and Electronic Resources: New Partnerships, New Practices, New Perspectives" of J. Library Automation, 35(1/2):125-145, 2002.
73. M. Goncalves and E. Fox. Technology and Research in a Global Networked University Digital Library (NUDL). Ciencia da Informacao (leading journal of library and information science in Brazil), 30(3): 13-23, Sep./Dec. 2001.
74. Hussein Suleman and Edward A. Fox. A Framework for Building Open Digital Libraries. D-Lib Magazine, 7(12), Dec. 2001, http://www.dlib.org/dlib/december01/suleman/12suleman.html
75. Hussein Suleman, Anthony Atkins, Marcos A. Goncalves, Robert K. France, and Edward A. Fox, Virginia Tech; Vinod Chachra and Murray Crowder, VTLS, Inc.; and Jeff Young, OCLC. Networked Digital Library of Theses and Dissertations: Bridging the Gaps for Global Access - Part 1: Mission and Progress. D-Lib Magazine, 7(9), Sept. 2001, http://www.dlib.org/dlib/september01/suleman/09suleman-pt1.html
76. Hussein Suleman, Anthony Atkins, Marcos A. Goncalves, Robert K. France, and Edward A. Fox, Virginia Tech; Vinod Chachra and Murray Crowder, VTLS, Inc.; and Jeff Young, OCLC. Networked Digital Library of Theses and Dissertations: Bridging the Gaps for Global Access - Part 2: Services and Research. D-Lib Magazine, 7(9), Sept. 2001, http://www.dlib.org/dlib/september01/suleman/09suleman-pt2.html
77. E. Fox and G. Marchionini. Digital Libraries: Extending Traditional Values. Guest Editors' Introduction to special section on Digital Libraries. Commun. of the ACM, 44(5):30-32, May 2001, http://doi.acm.org/10.1145/374308.374329
78. L. Cassel and E. Fox. Introducing the ACM Journal of Educational Resources in Computing (JERIC), editor-in-chiefs' introduction. 1(1), March 2001, http://doi.acm.org/10.1145/376697.382399
79. E. Fox. Short pieces (In Brief) in D-Lib Magazine (www.dlib.org):

- Suzie Allard and Edward A. Fox. The 4th International Conference of Asian Digital Libraries (ICADL 2001): "Digital Libraries: Dynamic Landscapes for Knowledge Creation, Access and Management", December 10-12, 2001, Bangalore, India,
- Fox, Edward A., A Sustainable, OAI-Based Implementation of NCSTRL, http://www.dlib.org/dlib/september01/09inbrief.html#FOX-IB
- Fox, Edward A., Open Archives Initiative, http://www.dlib.org/dlib/june00/06inbrief.html#FOX
- Fox, Edward A., First ACM/IEEE CS Joint Conference on Digital Libraries Announced, http://www.dlib.org/dlib/june00/06inbrief.html#FOX2
- Fox, Edward A., Fernando Adrian Das Neves, Robert France, Marcos Goncalves, and Hussein Suleman, Digital Libraries 2000, Fifth ACM Conference on Digital Libraries, http://www.dlib.org/dlib/july00/07inbrief.html#FOX

80. E. Fox. The Digital Libraries Initiative: Update and Discussion. Guest editor's introduction to Special Section. Bulletin of the American Society of Information Science, 26(1):7-11, Oct/Nov 1999
81. Rekha Kengeri, Cheryl D. Seals, Hope D. Harley, Himabindu P. Reddy, Edward A. Fox: Usability study of digital libraries: ACM, IEEE-CS, NCSTRL, NDLTD. Int J Digit Libr 2 (1999) 2/3, 157-169, http://link. springer.de/link/service/journals/00799/bibs/9002002/90020157.htm
82. E. Fox. Networked Digital Library of Theses and Dissertations, Nature Web Matters, http://helix.nature.com/webmatters/, invited short essay, 12 August 1999, http://helix.nature.com/webmatters/library/library.html
83. G. Marchionini and E. Fox. Progress toward digital libraries: augmentation through integration. Guest editors' introduction to Special Issue: Progress Toward Digital Libraries, Information Processing & Management, 1999, 35(3): 219-225.
84. E. Fox. Networked Digital Library of Theses and Dissertations, ERCIM News No. 35, October 1998, http://www.ercim.org/publication/Ercim_News/enw35/fox.html
85. B. Liu and E. Fox. Web Traffic Latency: Characteristics and Implications. Journal of Universal Computer Science, 4(9), Sept. 1998, http://www.iicm.edu/jucs_4_9/web_traffic_latency_characteristics
86. J. Powell and E. Fox. Multilingual Federated Searching Across Heterogeneous Collections, D-Lib Magazine, The Magazine of Digital Library Research, ISSN 1082-9873, Sep. 1998, http://www.dlib.org/dlib/september98/powell/09powell.html
87. E. Fox, N. Kipp, and P. Mather. How Digital Libraries Will Save Civilization. Database Programming & Design, 11(8):60-65, Aug. 1998, See also online Suggested Resources http://www.dbpd.com/foxweb.html
88. E. Fox and G. Marchionini. Toward a Worldwide Digital Library. Guest Editors' Introduction to special section (pp. 28-98) on Digital Libraries: Global Scope, Unlimited Access. Commun. of the ACM, Apr. 1998, 41(4): 28-32. http://purl.lib.vt.edu/dlib/pubs/CACM199804
89. E. Fox. Networked Digital Library of Theses and Dissertations: An International Collaboration Promoting Scholarship. ICSTI Forum, Quarterly Newsletter of the International Council for Scientific and Technical Information, No. 26: 8-9, Nov. 1997.
90. E. Fox, R. Hall, and N. Kipp. NDLTD: Preparing the Next Generation of Scholars for the Information Age. The New Review of Information Networking (NRIN), 3: 59-76, 1997.
91. E. Fox, Robert Hall, Neill A. Kipp, John L. Eaton, Gail McMillan, and Paul Mather. NDLTD: Encouraging International Collaboration in the Academy. In Special Issue on Digital Libraries, DESIDOC Bulletin of Information Technology (DBIT), 17(6): 45-56, Nov. 1997.
92. E. Fox, J. Eaton, G. McMillan, N. Kipp, P. Mather, T. McGonigle, W. Schweiker, and B. DeVane. Networked Digital Library of Theses and Dissertations: An International Effort Unlocking University Resources. D-Lib Magazine, The Magazine of Digital Library Research, ISSN 1082-9873, Sep. 1997, http://www.dlib.org/dlib/september97/theses/09fox.html
93. E. Fox, J. Eaton, G. McMillan, N. Kipp, L. Weiss, E. Arce, S. Guyer. National Digital Library of Theses and Dissertations: A Scalable and Sustainable Approach to Unlock University Resources. D-Lib Magazine, The Magazine of Digital Library Research, ISSN 1082-9873, Sep. 1996, http://www.dlib.org/dlib/september96/theses/09fox.html
94. J. N. Guidi and E. A. Fox. Information Retrieval and Genomics - An Introduction. Computers in Biology and Medicine, Pergamon, Guest Editor Intro. to Special Issue, 26(3): 179-182, May 1996.

95. S. Chen and E. Fox. Guest Editors' Introduction to Special Issue on Digital Libraries, Journal of Visual Communication and Image Representation, 7(1), March 1996, Academic Press.
96. C. Crouch, M. McGill, M. Lesk, K.S. Jones, E. Fox, D. Harman, and D. Kraft. In Memoriam - Gerard Salton, March 8, 1927 - August 28, 1995. Journal of the American Society for Information Science (JASIS), 1996, 47(2): 108-115.
97. E. Fox and L. Kieffer. Multimedia Curricula, Courses and Knowledge Modules, ACM Computing Surveys, Dec. 1995, 27(4): 549-551.
98. Edward A. Fox. Digital libraries: introduction. ACM SIGOIS Bulletin, 16(2): 8-9, December 1995, https://doi.org/10.1145/226188.226192
99. F. Can, E. Fox, C. Snavely, and R. France. Incremental Clustering for Very Large Document Databases: Initial MARIAN Experience. Information Systems, 84:101-114, 1995.
100. W. C. Dougherty and E. A. Fox. TULIP at Virginia Tech. Library Hi Tech, 13(4): 54-60, 1995.
101. E. Fox. Hypermedia Support for a Digital Library in CS. SIGLINK Newsletter, Sept. 1995, 4(2), Special Issue on Digital Libraries.
102. N. J. Belkin, P. Kantor, E. A. Fox and J. A. Shaw. Combining the Evidence of Multiple Query Representations for Information Retrieval. Information Processing & Management, 31(3), 431-448, May-June 1995.
103. E. Fox, R. Akscyn, R. Furuta, and J. Leggett. Guest Editors' Introduction to Digital Libraries. Commun. of the ACM, Apr. 1995, 38(4):22-28.
104. E. Fox. World-Wide Web and Computer Science Reports. Commun. of the ACM, Apr. 1995, 38(4):43-44.
105. L. Heath, D. Hix, L. Nowell, W. Wake, G. Averboch, and E. Fox. Envision: A User-Centered Database from the Computer Science Literature. Commun. of the ACM, Apr. 1995, 38(4):52-53.
106. J. French, E. Fox, K. Maly, and A. Selman. Wide Area Technical Report Service --- technical reports online. Commun. of the ACM, Apr. 1995, 38(4):45.
107. E. Fox. SIGIR. ACM Computing Surveys, March 1995, 27(1):130-132.
108. L. W. Carstensen, Jr., C. A. Shaffer, R. W. Morrill, E. A. Fox. GeoSim: A GIS-Based Simulation Laboratory for Introductory Geography. Journal of Geography, 1993, 92(5): 217-222. Winner of Best Article Related to Teaching in a University or College award from all articles in this National Council for Geographic Education's journal, May'92-April'94.
109. E. Fox. Digital Libraries ("hot topics" section), IEEE Computer, Nov. 1993, 26(11): 79-81.
110. E. Fox and L. Lunin. Introduction and Overview to Perspectives on Digital Libraries. Journal of the American Society for Information Science (JASIS), Sept. 1993, 44(8): 441-443. (Guest editor's introduction to special issue)
111. E. Fox, D. Hix, L. Nowell, D. Brueni, W. Wake, L. Heath, and D. Rao. Users, User Interfaces, and Objects: Envision, a Digital Library. Journal of the American Society for Information Science (JASIS), Sept. 1993, 44(8): 480-491.
112. E. A. Fox, L. S. Heath, Q.-F. Chen, and A. M. Daoud. Practical Minimal Perfect Hash Functions for Large Databases. Commun. of the ACM, Jan. 1992, 35(1): 105-121.
113. E. Fox. Advances in Interactive Digital Multimedia Systems. IEEE Computer, Oct. 1991, 24(10): 9-21. Reprinted in Readings in Groupware and Computer Supported Cooperative Work, ed. Ron Baecker, Morgan Kaufmann, San Mateo, CA, 1993, 342-354 and as Chapter 1 of Multimedia Computing: Preparing for the 21st Century, ed. Sorel Reisman, Idea Group Publishing, Harrisburg, PA, 1994, 3-33.
114. A. Siochi, E.A. Fox, D. Hix, E.E. Schwartz, A. Narasimhan, and W. Wake. The Integrator: A prototype for flexible development of interactive digital multimedia applications. Interactive Multimedia, July-Sept. 1991, 2(3): 5-26.
115. E. Fox, Q.-F. Chen, A. Daoud, and L. Heath. Order-Preserving Minimal Perfect Hash Functions and Information Retrieval. ACM Transactions on Information Systems, July 1991, 9(3), 281-308.
116. E. Fox. Standards and the Emergence of Digital Multimedia Systems. Commun. of the ACM, Apr. 1991, 34(4):26-29. Guest ed. intro., Special Section on Digital Multimedia.
117. J.T. Nutter, E.A. Fox, and M.W. Evens. Building a Lexicon from Machine- Readable Dictionaries for Improved Information Retrieval. Literary and Linguistic Computing, 1990, 5(2): 129-138.
118. E. Fox. How to Proceed Toward Electronic Archives and Publishing. Psychological Science, Nov. 1990, 1(6): 355-8.

119. E. Fox and S. Winett, Using Vector and Extended Boolean Matching in an Expert System for Selecting Foster Homes. Journal of the American Society for Information Science (JASIS), 1990, 41(1): 10-26.
120. M. Weaver, R. France, Q. Chen, E. Fox. Using a Frame-Based Language for Information Retrieval. Int. Journal of Intelligent Systems, 1989, 4(3): 223-257.
121. E. Fox. The Coming Revolution in Interactive Digital Multimedia Systems. Commun. of the ACM, 1989, 32(7): 794-801. Guest ed. intro. Spec. Section on Interactive Technologies.
122. E. Fox. Research and Development of Information Retrieval Models and their Application. Information Processing and Management (IP&M), 1989, 25(1): 1-5. Guest editor's introduction to Special Issue on Information Retrieval Models and their Application.
123. E. Fox. ACM Press Database and Electronic Products -- New Services for the Information Age. Commun. of the ACM, Aug. 1988, 31(8): 948-951.
124. E. Fox and M. Koll. Practical Enhanced Boolean Retrieval: Experiences with the SMART and SIRE Systems. IP&M, 24(3): 257-267, 1988.
125. E. Fox and M. Weaver. Highlights of CD-ROM Hardware. The Laserdisk Professional, May 1988, 1(1): 18-27.
126. E. Fox, M. Weaver, and N. Herther. Optical Laser Technology -- A Glossary of Basic Terms. Laserdisk Professional, May 1988, 1(1): 102-103.
127. E. Fox. Testing the Applicability of Intelligent Methods for Information Retrieval. Information Services and Use, 1987, 7: 119-138.
128. E. Fox. Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval. IP&M, 1987, 23(4): 341-366.
129. E. Fox and R. France. Architecture of an Expert System for Composite Document Analysis, Representation and Retrieval. International Journal of Approximate Reasoning, 1987, 1(2):151-175. Reprinted in Readings in Information Retrieval, eds. K.S. Jones and P. Willett San Francisco: Morgan Kaufmann, 1999
130. M. Tischler and E. Fox. An Expert System for Selecting Liquid Chromatographic Separation Methods. Computers and Chemistry, 1987, 11(4): 235-240.
131. N. Belkin, C. Borgman, H. Brooks, T. Bylander, W. Croft, P. Daniels, S. Deerwester, E. Fox, P. Ingwersen, R. Rada, K. Sparck Jones, R. Thompson, and D. Walker. Distributed Expert-Based Information Systems: An Interdisciplinary Approach. IP&M, 1987, 23(5): 395-409.
132. E. Fox and S. Birch. UNIX Micros for Students Majoring in Computer Science and Personal Information Retrieval. Microcomputers for Information Management, 1986, 3(1): 15-29.
133. G. Salton, E. Fox, and E. Voorhees. Advanced Feedback Methods in Information Retrieval. JASIS, 1985, 36(3): 200-210.
134. G. Salton, E. Voorhees, and E. Fox. A Comparison of Two Methods for Boolean Query Relevance Feedback. IP&M, 1984, 20(5-6): 637-651.
135. G. Salton, E. Fox, and H. Wu. Extended Boolean Information Retrieval. Communications of the ACM, 1983, 26(11): 1022-1036.
136. G. Salton, C. Buckley, and E. Fox. Automatic Query Formulations in Information Retrieval. JASIS, 1983, 34(4): 262-280.

BOOK CHAPTERS:

1. Edward A. Fox, Monika Akbar, Sherif Hanie El Meligy Abdelhamid, Noha Ibrahim Elsherbiny, Mohamed Magdy Gharib Farag, Fang Jin, Jonathan P. Leidig, Sai Tulasi Neppali. Digital Libraries. In Computing Handbook, Third Edition, Vol. 2 (Information Systems and Information Technology), Section 3, Ch. 18, ed. by Heikki Topi, Allen Tucker, Chapman & Hall/CRC Press, Taylor and Francis Group, ISBN 9781439898444, http://www.crcpress.com/product/isbn/9781439898543, May 2014
2. Edward A. Fox and Noha ElSherbiny. Security and Digital Libraries. Chapter 8 in "Digital Libraries: Methods and Applications", ed. Kuo Hung Huang, ISBN 978-953-307-203-6. InTech, Rijeka, Croatia. Free online from www.intechopen.com. 2011, pp. 151-160 (downloaded 4000 times by 11/15/2016)

3. Seungwon Yang, Sanghee Oh, Barbara M. Wildemuth, Jeffrey P. Pomerantz, and Edward A. Fox. Educational Resource Development for Information Retrieval in a Digital Libraries Context. Chapter 14 in Efthimis N. Efthimiadis, Juan Manuel Fernandez-Luna, Juan F. Huete, Andrew MacFarlane, eds., Teaching and Learning in Information Retrieval, 2011, Berlin: Springer Verlag, ISSN 1387-5264, ISBN 978-3-642-22510-9, 199-212
4. Edward Fox, Uma Murthy, Seungwon Yang, Ricardo da S. Torres, Javier Velasco-Martin, and Gary Marchionini. Pedagogical Enhancements for Information Retrieval Courses. Chapter 4 in Efthimis N. Efthimiadis, Juan Manuel Fernandez-Luna, Juan F. Huete, Andrew MacFarlane, eds., Teaching and Learning in Information Retrieval, 2011, Berlin: Springer Verlag, ISSN 1387-5264, ISBN 978-3-642-22510-9, 47-60
5. Nadia P. Kozievitch, Ricardo da Silva Torres, Edward A. Fox, Sung Hee Park, Nathan Short, Lynn Abbott, Supratik Misra, Michael Hsiao: Rethinking fingerprint evidence through integration of very large digital libraries. In: Castelli, D., Ioannidis, Y., Manghi, P., Pagano, P., Ross, S. (eds.). Proceedings of the Second DL.org Workshop on Making DLs Interoperable: Challenges & Approaches (MDLI2010) and Proceedings of the Third Workshop on Very Large Digital Libraries (VLDL2010), In conjunction with the European Conference on Digital Libraries 2010, Glasgow, Scotland (UK), 10th of September 2010, Springer LNCS
6. George Athanasopoulos, Katerina El Raheb, Edward Fox, George Kakaletris, Natalia Manola, Carlo Meghini, Andreas Rauber, and Dagobert Soergel. Framework for Digital Library Function Description, Publication, and Discovery: A prerequisite for interoperable digital libraries. In: Castelli, D., Ioannidis, Y., Manghi, P., Pagano, P., Ross, S. (eds.). Proceedings of the Second DL.org Workshop on Making DLs Interoperable: Challenges & Approaches (MDLI2010) and Proceedings of the Third Workshop on Very Large Digital Libraries (VLDL2010). In conjunction with the European Conference on Digital Libraries 2010, Glasgow, Scotland (UK), 10th of September 2010, Springer LNCS
7. Jonathan Leidig, Edward A. Fox, Madhav Marathe, and Henning Mortveit. Epidemiology Experiment and Simulation Management through Schema-Based Digital Libraries. In: Castelli, D., Ioannidis, Y., Manghi, P., Pagano, P., Ross, S. (eds.). Proceedings of the Second DL.org Workshop on Making DLs Interoperable: Challenges & Approaches (MDLI2010) and Proceedings of the Third Workshop on Very Large Digital Libraries (VLDL2010), In conjunction with the European Conference on Digital Libraries 2010, Glasgow, Scotland (UK), 10th of September 2010, Springer LNCS
8. Edward A. Fox, Gail McMillan, and Venkat Srinivasan. Electronic Theses and Dissertations: Progress, Issues, and Prospects. Chapter 8 in Timothy W. Luke and Jeremy W. Hunsinger, eds., Putting Knowledge to Work & Letting Information Play: The Center for Digital Discourse and Culture, 10th Anniversary book, Dec. 2009, ISBN 978-1-933217-00-0, Virginia Tech, Blacksburg, VA, 126-148 http://www.cddc.vt.edu/10th-book/putting_knowledge_to_work.pdf
9. Venkat Srinivasan, Seungwon Yang, and Edward A. Fox. Digital Libraries. In Encyclopedia of Database Systems. Eds. Ling Liu and M. Tamer Ozsu. Springer Berlin, 2009, ISBN 978-0-387-35544-3
10. Rao Shen and Edward A. Fox. Browsing in Digital Libraries. In Encyclopedia of Database Systems. Eds. Ling Liu and M. Tamer Ozsu. Springer Berlin, 2009, ISBN 978-0-387-35544-3
11. Xiaoyan Yu, Manas Tungare, Weiguo Fan, Manuel Perez-Quinones, Edward A. Fox, William Cameron, Lillian Cassel: Automatic Genre-specific Text Classification. In Encyclopedia of Data Warehousing and Mining (2nd edition), edited by John Wang, Idea Group Inc., 2008
12. Xiaoyan Yu, Manas Tungare, Weiguo Fan, Yubo Yuan, Manuel Perez-Quinones, Edward A. Fox, William Cameron, Lillian Cassel: Automatic Syllabus Classification using Support Vector Machines. In Handbook of Research on Text and Web Mining Technologies, edited by Min Song, Yi-Fang Wu, Idea Group Inc., 2008
13. Seungwon Yang, Barbara M. Wildemuth, Jeffrey P. Pomerantz, Sanghee Oh and Edward A. Fox. Core Topics in Digital Library Education. In Handbook of Research on Digital Library: Design, Development, and Impact (ISBN 978-1-59904-879-6), eds. Ying-Leng Theng, Schubert Foo, Dion Hoe-Lian Goh, and Jin-Cheon Na, Publisher: IGI Global, 2009, 493-505
14. Baoping Zhang, Weiguo Fan, Yuxin Chen, Edward A. Fox, Marcos Andre Goncalves, Marco Cristo and Pavel Calado. A Genetic Programming Approach for Combining Structural and Citation-Based Evidence for Text Classification in Web Digital Libraries. In Soft Computing in Web

Information Retrieval: Models and Applications, in series: Studies in Fuzziness and Soft Computing, Vol. 197, Herrera-Viedma, Enrique; Pasi, Gabriella; Crestani, Fabio (Eds.), Jan. 2006, pp. 65-83, ISBN: 3-540-31588-8, http://dx.doi.org/10.1007/3-540-31590-X_4.

15. Edward A. Fox, Hussein Suleman, Ramesh C. Gaur, Devika P. Madali. Design Architecture: An Introduction and Overview. Chapter II (pages 22-37) in Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, eds. Yin-Leng Theng and Schubert Foo, Hersey, PA: Idea Group Inc., 2005
16. Edward A. Fox, Marcos Andre Goncalves, and Rao Shen. The Role of Digital Libraries in Moving Toward Knowledge Environments. Invited chapter in "From Integrated Publication and Information Systems to Information and Knowledge Environments: Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday", eds. Matthias Hemmje, Claudia Niederee, and Thomas Risse, Lecture Notes in Computer Science, Volume 3379, Jan. 2005, pp. 96-106, Springer-Verlag GmbH, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3379&spage=96
17. Fox, E., Suleman, H., Madalli, D., Cassel, L. Digital Libraries. Invited chapter 4 for "Practical Handbook of Internet Computing", in Munindar P. Singh ed., Chapman Hall & CRC Press, Baton Rouge, 2004, ISBN 1-584-88381-2
18. Fox, E., McMillan, G., Suleman, H., Goncalves, M., and Luo, M., Networked Digital Library of Theses and Dissertations (NDLTD). In "Digital Libraries: Policy, Planning and Practice", eds. Judith Andrews and Derek Law, Hants, England: Ashgate Publishing, ISBN 0-7546-3448-5, 2004, Chapter 11, pp. 167-187
19. Lee JAN, Impagliazzo J, Cassel L, Fox E, Giles L, Knox D, Perez-Quinones M. Enhancing distance learning using quality digital libraries and CITIDEL. In Quality Education (at) a Distance, eds. Davies G, Stacey E. Norwell, MA: Kluwer, pages 61-71, 2003. ISBN: 1-4020-7568-5. Working Conference on Quality Education (AT) a Distance, Geelong, AU, Feb. 3-6. Int Federat Informat Proc 131, TC3, WG3 6
20. Edward A. Fox, Marcos Andre Goncalves, Ming Luo, Yuxin Chen, Aaron Krowne, Baoping Zhang, Kate McDevitt, Manuel A. Perez-Quinones, Ryan Richardson, Lillian N. Cassel: Harvesting: Broadening the Field of Distributed Information Retrieval. In Distributed Multimedia Information Retrieval, Springer-Verlag, SIGIR 2003 Workshop on Distributed Information Retrieval, Toronto, Canada, August 1, 2003, Revised Selected and Invited Papers; Lecture Notes in Computer Science, Vol. 2924; Callan, Jamie; Crestani, Fabio; Sanderson, Mark (Eds.), 2004, ISBN: 3-540-20875-5, pages 1-20, http://www.springerlink.com/index/92W9V810P75U2H41
21. Edward A. Fox and Paul Mather. Scalable Storage for Digital Libraries. Chapter 12 in Multimedia Information Retrieval and Management: Technological Fundamentals and Applications, eds. D. Feng, W.C. Siu and H.J. Zhang, Berlin: Springer-Verlag, 2003, http://www.springer.de/cgi/svcat/search_book.pl?isbn=3-540-00244-8, pp. 265-288
22. Edward A. Fox and Paul Mather. Object Repositories for Digital Libraries. Chapter 13 in Multimedia Information Retrieval and Management, eds. D. Feng, W.C. Siu and H.J. Zhang, Berlin: Springer-Verlag, 2003, http://www.springer.de/cgi/svcat/search_book.pl?isbn=3-540-00244-8, pp. 289-306
23. Rao Shen, Jun Wang, and Edward A. Fox. A Lightweight Protocol between Digital Libraries and Visualization Systems. In Proceedings of JCDL Workshop on Visual Interfaces to Digital Libraries, July 18, 2002, Portland, eds. Katy Borner & Chaomei Chen, Springer Verlag, LNCS Series, Vol 2539, 2002, pp. 217-225, http://portal.acm.org/citation.cfm?id=659274
24. Advancing Education through Digital Libraries: NSDL, CITIDEL, and NDLTD. In the Procedings of Digital Library: IT Opportunities and Challenges in the New Millennium, ed. Sun Jiazheng, Beijing, China: Beijing Library Press, July 9-11, 2002, pp. 107-117.
25. E. Fox. Forward, for How to Build a Digital Library, by Ian Witten and David Bainbridge, San Francisco: Morgan Kaufmann, 2002
26. Edward A. Fox and Shalini R. Urs. Digital Libraries. Annual Review of Information Science and Technology, ed. Blaise Cronin, Vol. 36, Ch. 12, pp. 503-589, 2002.
27. E. Fox. Overview of Digital Library Components and Developments. Chapter 4 in Digital Libraries and Virtual Workplaces: Important initiatives for Latin America in the Information Age, edited by Johann van Reenen, Washington: Organization of American States, 2002, ISBN 0-8270-4377-5, pp. 105-123.

28. E. Fox. Overview of a Guide for Electronic Theses and Dissertations. Chapter 5 in Digital Libraries and Virtual Workplaces: Important initiatives for Latin America in the Information Age, edited by Johann van Reenen, Washington: Organization of American States, 2002, ISBN 0-8270-4377-5, pp. 125-156.
29. Edward A. Fox, Marcos A. Goncalves and Neill A. Kipp. Digital Libraries. In Handbook on Information Technologies for Education and Training, in Springer series "International Handbook on Information Systems", ISBN 3-540-67803-4, eds. Heimo H. Adelsberger, Betty Collis, and Jan Pawlowski, 2002, Berlin, 623-641.
30. E. Fox. Forward for Readings in Multimedia Computing, eds. Kevin Jeffay and Hong-Jiang Zhang, San Francisco: Morgan Kaufmann, 2001
31. Edward A. Fox and Ohm Sornil. Digital Libraries. Encyclopedia of Computer Science, 4th edition, edited by Anthony Ralston, Edwin D. Reilly, and David Hemmendinger, Nature Publishing Group, London, 2000, pages 576-581.
32. Edward A. Fox. Update on the Networked Digital Library of Theses and Dissertations. In Successes & Failures of Digital Libraries, ed. Susan Harum and Michael Twidale, Urbana-Champaign: U. Illinois, ISBN 0-8745-107-2, 2000, pages 12-20
33. Edward A. Fox and Ohm Sornil. Digital Libraries. Chapter 15 in Modern Information Retrieval, ACM Press / Addison-Wesley-Longman England, 1999: Ricardo Baeza-Yates and Berthier Ribeiro-Neto, eds., 415-432
34. E. Fox, ed. (chair of Subcommittee on "Traditional" Undergraduates). Chapter 3: Traditional Undergraduate Learners. In Transforming Virginia Tech For the Information Age, Reaffirmation of Accreditation Self-Study Report to the Commission on Colleges of the Southern Assoc. of Colleges and Schools, Dec. 1997, Virginia Tech, Blacksburg, VA. http://www.vt.edu:10021/admin/provost/selfstudy/report/index.html
35. W. Wake, B. D. Wake, and E. Fox. Improving Responsiveness in Interactive Applications Using Queues. In Part 8 of Pattern Languages of Program Design 2, edited by Vlissides, Coplien, and Kerth, Reading, MA: Addison-Wesley, 1996, 563-573. (Previously published in Proc. PLOP '95.)
36. E. Fox, J.T. Nutter, and M.W. Evens. A Lexicon Server using Lexical Relations for Information Retrieval. Ch. 11, Machine-Tractable Dictionaries: Design and Construction, Cheng-ming Guo, ed., Ablex Publishing Corp. (Norwood, NJ), 1995, 89-92.
37. E. Fox. How to make intelligent digital libraries. In Methodologies for Intelligent Systems, Proceedings of the 8th International Symposium, ISMIS'94, Charlotte, NC, Oct. 1994. Lecture Notes in Artificial Intelligence 869, Springer-Verlag, Berlin, 27-38.
38. E. Fox. Advances in Interactive Digital Multimedia Systems. Reprinted in Readings in Groupware and Computer Supported Cooperative Work, ed. Ron Baecker, Morgan Kaufmann, San Mateo, CA, 1993, 342-354. Also published as Chapter 1 of Multimedia Computing: Preparing for the 21st Century, ed. Sorel Reisman, Idea Group Publishing, Harrisburg, PA, 1994, 3-33. Originally in IEEE Computer, Oct. 1991, 24(10): 9-21.
39. S. Patel, G. Abdulla, M. Abrams, and E. Fox. NMFS: Network Multimedia File System Protocol. In P. Venkat Rangan, Ed., Network and Operating System Support for Digital Audio and Video, Lecture Notes in Computer Science 712, Springer-Verlag, Berlin Heidelberg, 1993, ISBN 3-540-57183-3 and 0-387-57183-3, 328-333. Reprinted from conf. proceedings of Third International Workshop on Network and Operating System Support for Digital Audio and Video, Nov. 12-13, 1992, La Jolla, CA.
40. D. Harman, E. Fox, R. Baeza-Yates, and W. Lee. Inverted Files. In Information Retrieval: Data Structures & Algorithms, editors W. Frakes & R. Baeza-Yates, Prentice-Hall, 1992, 28-43. Also on CD-ROM published by Dr. Dobb's Journal.
41. S. Wartik, E. Fox, L. Heath, and Q. Chen. Hashing Algorithms. In Information Retrieval: Data Structures & Algorithms, eds. W. Frakes & R. Baeza-Yates, Prentice-Hall, 1992, 293-362. Also on CD-ROM published by Dr. Dobb's Journal.
42. E. Fox, S. Betrabet, M. Koushik, and W. Lee. Extended Boolean Models. In Information Retrieval: Data Structures & Algorithms, eds. W. Frakes & R. Baeza-Yates, Prentice-Hall, 1992, 393-418. Also on CD-ROM published by Dr. Dobb's Journal.
43. E. Fox, Q.-F. Chen, and R. K. France. Integrating Search and Retrieval with Hypertext. In Hypertext/ Hypermedia Handbook, ed. E. Berk and J. Devlin, McGraw-Hill, New York, 1991, 329-355.
44. E. Fox, B. Rous, and G. Marchionini. ACM's Hypertext and Hypermedia Publishing Projects. In Hypertext/ Hypermedia Handbook, ed. E. Berk and J. Devlin, McGraw-Hill, New York, 1991, 465-467.
45. E. Fox and M. Weaver. Highlights of CD-ROM Hardware. In Practical Tips & Techniques for Using CD-ROM Systems, 1990, 62-71.

46. E. Fox, M. Weaver, and N. Herther. Optical/Laser Technology: A Glossary of Basic Terms. In Practical Tips & Techniques for Using CD-ROM Systems, 1990, 92-93.
47. E. Fox. Improved Retrieval Using a Relational Thesaurus for Automatic Expansion of Boolean Logic Queries. In Relational Models of the Lexicon: Representing Knowledge in Semantic Networks, Martha Walton Evens (ed.), Cambridge Univ. Press, Cambridge, UK, 1988, 199-210.
48. E. Fox. Optical Disks and CD-ROM: Publishing and Access. In Annual Review of Information Science and Technology, Martha E. Williams (ed.), ASIS / Elsevier Science Publishers B.V., Amsterdam, 1988, Vol. 23, 85-124.
49. E. Fox. Information Retrieval: Research into New Capabilities. In CD-ROM: The New Papyrus, Steve Lambert and Suzanne Ropiequet (eds.), Microsoft Press, Redmond, WA, 1986, 143-174.

REFEREED CONFERENCE/WORKSHOP PAPERS:

1. Eman Abdelrahman, Fatimah Alotaibi, Edward A. Fox, and Osman Balci. Otrouha: A Corpus of Arabic ETDs and a Framework for Automatic Subject Classification. Proc. Electronic Theses and Dissertations, ETD 2020: Discovery, Data, and Dates, virtual conference, Nov. 16-18, 2020, UAE, 8 pages, to be presented
2. Nesreen Ahmed, Richard Alo, Catherine Amelink, Young Yun Baek, Aashish Chudhary, Kristy Collins, Albert Esterline, Edward Fox, Geoffrey Fox, Aric Hagberg, Ron Kenyon, Chris Kuhlman, Jure Leskovec, Dustin Machi, Madhav Marathe, Nataragan Meghanathan, Yasuo Miyasaki, Judy Qiu, Naren Ramakrishnan, S. S. Ravi, Ryan Rossi, Roc Sosic and Gregor von Laszewski. net.science: A Cyberinfrastructure for Sustained Innovation in Network Science and Engineering. Proc. Gateways 2020, short paper, in press
3. Torfi, Amirsina, Mohammadreza Beyki, and Edward A. Fox. On the Evaluation of Generative Adversarial Networks By Discriminative Models. International Conference on Pattern Recognition. ICPR 2020. Accepted.
4. Zhiwu Xie, Martin Klein, and Edward A. Fox. Web Archiving and Digital Libraries. Proc. JCDL 2020, Aug. 1-5, Virtual Event, China, workshop, 2 pages, https://doi.org/10.1145/3383583.3398509
5. Andrea Kavanaugh, Ziqian Song, Liuqing Li, Edward A. Fox, Bethany Hsiao. Too Small to Fail: Information sharing behavior in a US Municipal Election. Proceedings 21st Annual International Conference on Digital Government Research (dg.o 2020), (online) conference, Seoul, South Korea, ACM, 14 pages
6. Liuqing Li and Edward Fox. Disaster Response Patterns across Different User Groups on Twitter: A Case Study during Hurricane Dorian. WiP/Practitioner paper in Proceedings ISCRAM 2020, May 2020, pp. 838-848, also in the ISCRAM Digital Library, http://idl.iscram.org/files/liuqingli/2020/2275_LiuqingLi+EdwardA.Fox2020.pdf, to be presented in May 2021 at ISCRAM 2021, Blacksburg, VA, 11 pages.
7. Amirsina Torfi, Edward A. Fox. COR-GAN: Correlation-Capturing Convolutional Neural Networks for Generating Synthetic Healthcare Records. AI in Healthcare Informatics Track of FLAIRS-33, The 33rd International FLAIRS Conference, The Florida Artificial Intelligence Research Society in cooperation with AAAI, North Miami Beach, FL, May 17-20, 2020, AAAI (with option to present at FLAIRS-34 due to COVID-19)
8. Saurabh Chakravarty, Maanav Mehrotra, Raja Venkata Satya Phanindra Chava, Han Liu, Matthew Krivansky, Edward A. Fox. Improving the Processing of Question Answer Based Legal Documents. In Proc. Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, Dec. 11-13, 2019. Vol. 322, p. 13, IOS Press.
9. Steven D. Sheetz, Andrea Kavanaugh, Hamida Skandrani, Edward A. Fox. Uses and Gratifications of Political Information: Student Perceptions of Information from the 2014 Tunisian Elections. In Proc. Deep Transformations and the Future of Organisations, 4th International Conference of the Interdisciplinary Laboratory of University-Business Management, University of Manouba, Tunis, Tunisia, 6-7 December 2019

10. Ingram, William A., Bipasha Banerjee, and Edward A. Fox. Summarizing ETDs with deep learning. Presented by Ingram and Fox at 22nd International Symposium on Electronic Theses and Dissertations (ETD 2019), Nov. 6-8, 2019, Porto, Portugal.
11. Ziqian Song, Wenqi Shen, Weiguo (Patrick) Fan, Edward A. Fox. Measuring the Impact of Corporate Crisis News Propagation via Twitter. Informs 11th Conference on Information Systems and Technology (CIST 2019), Oct. 19-20, Seattle.
12. Saurabh Chakravarty, Raja Venkata Satya Phanindra Chava, and Edward A. Fox. Dialog Acts Classification for Question-Answer Corpora. In Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019), part of the 19th International Conference on Artificial Intelligence and Law (ICAIL 2019), June 21, 2019, Montreal, QC, Canada.
13. Andrea L. Kavanaugh, Ziqian Song, Liuqing Li, Edward A. Fox. Communication Behavior in an Emerging Democracy: Political Expression via Tweets during the 2014 Tunisian Elections, in Proceedings of the 20th Annual International Conference on Digital Government Research (dg.o 2019). June 18-20, Dubai, United Arab Emirates, ACM, 2019, pp. 445-455, doi:10.1145/3325112.3325263.
14. Liuqing Li, Rishabh Anand, and Edward A. Fox. Users, User Roles, and Topics in School Shooting Collections of Tweets. In Proc. WADL 2019, Web Archiving and Digital Libraries Workshop at JCDL 2019, UIUC, Urbana-Champaign, Illinois, USA, June 6, 2019, http://fox.cs.vt.edu/talks/2019/20190606LiAnandFoxWADL2019.pdf.
15. Steven Sheetz, Andrea Kavanaugh, Edward Fox, Riham Hassan, Seungwon Yang, Mohamed Magdy, and Donald Shoemaker. (2019). Information Uses and Gratifications Related to Crisis: Student Perceptions since the Egyptian Uprising. In Z. Franco, J. J. Gonzalez, & J. H. Canos (Eds.), Proceedings of the 16th International Conference on Information Systems for Crisis Response And Management, ISCRAM 2019, pp. 674-690. Valencia, Spain, 19-22 May 2019: ISCRAM. http://idl.iscram.org/files/stevensheetz/2019/1862_StevenSheetz_etal2019.pdf
16. Liuqing Li and Edward A. Fox. (2019). Understanding patterns and mood changes through tweets about disasters. In Z. Franco, J. J. Gonzalez, & J. H. Canos (Eds.), Proceedings of the 16th International Conference on Information Systems for Crisis Response And Management (ISCRAM 2019). Valencia, Spain, 19-22 May 2019: ISCRAM. http://idl.iscram.org/files/liuqingli/2019/1863_LiuqingLi+EdwardA.Fox2019.pdf
17. Xuan Zhang, Zhilei Qiao, Aman Ahuja, Weiguo Fan, Edward A. Fox, and Chandan K. Reddy. 2019. Discovering Product Defects and Solutions from Online User Generated Contents. In Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13-17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313732, 20% acceptance rate.
18. Aman Ahuja, Ashish Baghudana, Wei Lu, Edward A. Fox, and Chandan K. Reddy. (2019) Spatio-Temporal Event Detection from Multiple Data Sources. In: Yang Q., Zhou ZH., Gong Z., Zhang ML., Huang SJ. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2019. Proc. 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining, 14-17 April 2019, Macau, China. Lecture Notes in Computer Science, vol 11439, pages 293-305. Springer, Cham. http://dmkd.cs.vt.edu/papers/PAKDD19.pdf, https://doi.org/10.1007/978-3-030-16148-4_23
19. Florian J. Zach, Yufeng Ma, and Edward A. Fox. A Preliminary Analysis of Images in Online Hotel Reviews. Proc. ENTER 2019: The 26th Annual eTourism Conference, Nicosia, Cyprus, 30 January - 1 February, 2019
20. Donald J. Shoemaker, Jason Callahan, Liuqing Li, Ziqian Song, and Edward Fox, Visual Comparisons of Tweets and URLs for Ten School Shootings. Roundtable presentation delivered at the annual meetings of the American Society of Criminology, Atlanta, Georgia, November 15, 2018.
21. Yinlin Chen and Edward A. Fox. Architecting a Cloud-native Data Analysis Application for ETDs. Presented at ETD 2018, the 21st International Symposium on Electronic Theses and Dissertations, Taiwan, Sept. 26-28. http://fox.cs.vt.edu/talks/2018/20180926ETD2018ChenFox.pdf, http://fox.cs.vt.edu/talks/2018/20180926ETD2018ChenFox.pptx.
22. Liuqing Li and Edward Fox. A Study of Historical Short URLs in Event Collections of Tweets. In Proc. WADL 2018, Web Archiving and Digital Libraries Workshop at JCDL 2018, Fort Worth, TX, USA.
23. Martin Klein, Zhiwu Xie, and Edward A. Fox. 2018. Web Archiving and Digital Libraries (WADL). In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL '18). ACM, New York, NY, USA, 425-426. DOI: https://doi.org/10.1145/3197026.3200209

24. Edward Fox. 2018. Introduction to Digital Libraries. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL '18). ACM, New York, NY, USA, 415-416. DOI: https://doi.org/10.1145/3197026.3201779, http://fox.cs.vt.edu/talks/2018/20180603FoxTutorialSlidesJCDL.pptx
25. Abigail Bartolome, Edward Fox and Scott McCrickard. Understanding Trail Cultures through Various Stakeholders of the Trail. In Proc. "W12: HCI Outdoors: Understanding Human-Computer Interaction in the Outdoors" Workshop at CHI 2018, 21 April 2018, Montreal, Canada
26. Abigail Bartolome, D. Scott McCrickard, and Edward A. Fox. Exploring cultural differences in the triple crown trails. In ACM GROUP 2018 workshop on Technology on the Trail., Sanibel Island, FL USA, January 2018.
27. Yufeng Ma, Tingting Jiang, Chandani Shrestha, Edward A. Fox, Jian Wu, C. Lee Giles. Scenarios for Advanced Services in an ETD Digital Library. In proceedings of ETD2017, the 20th international symposium on electronic theses and dissertations, Washington, DC, August 7-9, 2017. http://fox.cs.vt.edu/talks/2017/20170807ETD2017etdseer.pptx
28. Eduardo P.S. Castro, Saurabh Chakravarty, Eric Williamson, Denilson Alves Pereira, and Edward A. Fox. Classifying Short Unstructured Data Using the Apache Spark Platform. In Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, ON, Canada, June 19-23, 2017, pp. 1-10, acceptance rate 30%, DOI: 10.1109/JCDL.2017.7991567
29. Edward A. Fox. Introduction to Digital Libraries. In Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, Canada, June 19-23, 2017, 2 pages, DOI: 10.1109/JCDL.2017.7991620
30. Edward A. Fox, Zhiwu Xie, and Martin Klein. Web Archiving and Digital Libraries (WADL). In Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, Canada, June 19-23, 2017, 2 pages, DOI: 10.1109/JCDL.2017.7991625
31. Saurabh Chakravarty, Eric Williamson and Edward Fox. Classification of Tweets using Augmented Training. In Proc. WADL 2017, a workshop held in conjunction with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, ON, Canada, June 19-23, 2017, acceptance rate 80%
32. Prashant Chandrasekar, Islam Harb, Elsa Tai, Saurabh Chakravarty, Monika Akbar, Ann Gates, Chris Frank, Warren K. Bickel, and Edward Fox. A DL framework and case studies with linked open data. Paper and presentation at RUMOUR 2017, a workshop held in conjunction with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2017, http://2017.jcdl.org/), Toronto, Canada, June 19-23, 2017
33. Xuan Zhang, Zhilei Qiao, Lijie Tang, Weiguo Fan, Edward A. Fox, Gang Wang: Identifying Product Defects from User Complaints: A Probabilistic Defect Model. 22nd Americas Conference on Information Systems, AMCIS 2016, San Diego, CA, USA, August 11-14, 2016. Association for Information Systems 2016. http://aisel.aisnet.org/amcis2016/Decision/Presentations/14/
34. Venkat Srinivasan and Edward Fox. Progress toward automated ETD cataloging. In Proc. 19th International Symposium on Electronic Theses and Dissertations, ETD 2016 "Data and Dissertations", 11-13 July 2016, Lille (France), in Session 3.1 on 7/13 "ETD metadata and cataloging", http://etd2016.sciencesconf.org/92334
35. Zhiwu Xie, Krati Nayyar, and Edward A. Fox. Nearline Web Archiving. Paper presented at WADL 2016: Third International Workshop on Web Archiving and Digital Libraries, June 22-23, 2016. In connection with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, http://fox.cs.vt.edu/wadl2016.html
36. Mohamed Farag and Edward A. Fox. Which webpage should we crawl first? Social media-based webpage source importance guidance. Paper presented at WADL 2016: Third International Workshop on Web Archiving and Digital Libraries, June 22-23, 2016. In connection with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, http://fox.cs.vt.edu/wadl2016.html
37. Edward A. Fox. Introduction to Digital Libraries. In Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, June 19-23, 2016, 283-284, http://dx.doi.org/10.1145/2910896.2925429
38. Edward A. Fox, Zhiwu Xie, and Martin Klein. WADL 2016: Third International Workshop on Web Archiving and Digital Libraries. In Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, June 19-23, 2016, 293-294, http://dx.doi.org/10.1145/2910896.2926735

39. Zhiwu Xie, Andrej Galad, Yinlin Chen, and Edward Fox. Are Repositories Impeding Big Data Reuse? In Proc. 11th International Conference on Open Repositories. Trinity College, Dublin, Ireland, 13-16 June 2016
40. Warren K. Bickel, Amanda J. Quisenberry, Prashant Chandrasekar, Edward A. Fox, Christopher T. Franck. The Social Interactome of Recovery: Network Topology Influences Social Media Engagement. Proc. 2016 CPDD (College on Problems of Drug Dependence) Annual Meeting, selected based on abstract for oral presentation June 15, 2016
41. Andrea Kavanaugh, Steven D. Sheetz, Hamida Skandrani, John C. Tedesco, Yue Sun, and Edward A. Fox. The Use and Impact of Social Media during the 2011 Tunisian Revolution, Proc. 17th International Digital Government Research Conference (dg.o 2016), Fudan University, China, June 8-10, 2016, Yushim Kim and Monica Liu (Eds.). ACM, New York, NY, USA, 20-30, http://dx.doi.org/10.1145/2912160.2912175
42. Warren K. Bickel, Prashant Chandrasekar, Mikhail Koffarnus, Edward A. Fox, Christopher T. Franck, Sandesh Bhandari, Amanda J. Quisenberry. The Social Interactome of Recovery: Social Media Network Topology Influences Relapse. 39th Annual Scientific Meeting of the Research Society on Alcoholism, June 25-29, New Orleans, Louisiana, in Alcoholism: Clinical & Experimental Research, Vol. 40, Issue S1, p. 235A, http://dx.doi.org/10.1111/acer.13084
43. Edward A. Fox, Mohamed Farag, Sunshin Lee, Xuan Zhang, Richard Gruss. Conversation: Problem/project-based Learning with Big Data. Proc. 2016 Conference on Higher Education Pedagogy. Feb. 10-12, 2016. Blacksburg, VA, USA.
44. Tarek Kanan, Souleiman Ayoub, Eyad Saif, Ghassan Kanaan, Prashant Chandrasekar, Edward Fox. Extracting Named Entities Using Named Entity Recognizer and Generating Topics Using Latent Dirichlet Allocation Algorithm for Arabic News Articles. Proceedings International Computer Sciences and Informatics Conference (ICSIC 2016), Amman-Jordan.
45. Mohamed Farag and Edward Fox. Building and Archiving Event Web Collections: A focused crawler approach. 20 minute presentation and paper in Proc. Web Archiving and Digital Libraries Workshop (WADL 2015) - Joint Conference on Digital Libraries (JCDL). Knoxville, TN. 24 June 2015
46. Tarek Kanan, Xuan Zhang, Mohammed Magdy, and Edward Fox. Big Data Text Summarization for Events: a Problem Based Learning Course. Proc. of the Joint Conference on Digital Libraries (JCDL 2015). June 21-25, 2015. Knoxville, TN, pp. 87-90. http://dx.doi.org/10.1145/2756406.2756943
47. Edward A. Fox. Introduction to Digital Libraries. Tutorial overview in Proc. of the Joint Conference on Digital Libraries (JCDL 2015). June 21-25, 2015. Knoxville, TN, p. 291. http://dx.doi.org/10.1145/2756406.2756927
48. Edward A. Fox and Zhiwu Xie. Web Archiving and Digital Libraries (WADL). Workshop overview in Proc. of the Joint Conference on Digital Libraries (JCDL 2015). June 21-25, 2015. Knoxville, TN, p. 303. http://dx.doi.org/10.1145/2756406.2756934
49. Wingyan Chung, Edward C. Carr, Robert Beck, Edward A. Fox. Introduction: The Computing in Context Project and CIC15. Proc. of the 2015 NSF Workshop on Curricular Development for Computing in Context (CIC15), May 13, 2015, DeLand, Florida, ACM ISBN: 978-1-4503-3597-3.
50. Mohamed Magdy Farag and Edward A. Fox. Web Archive Content Analysis: Disaster Events Case Study. International Internet Preservation Consortium (IIPC) 2015 General Assembly (GA2015), Stanford, Palo Alto, CA, April 27 - May 1, 2015
51. Zhiwu Xie and Edward A. Fox. Archiving Transactions Towards Uninterruptible Web Service. International Internet Preservation Consortium (IIPC) 2015 General Assembly (GA2015), Stanford, Palo Alto, CA, April 27 - May 1, 2015
52. Hamed Alhoori, Sagnik Ray Choudhury, Tarek Kanan, Edward A. Fox, Richard Furuta, and C. Lee Giles. On the Relationship between Open Access and Altmetrics. Proc. iConference 2015, Newport Beach, CA, March 24-27, 2015, iSchools, issue date 15 March 2015. http://hdl.handle.net/2142/73451, https://www.ideals.illinois.edu/handle/2142/73451, https://www.conftool.com/iConference2015/index.php?page=browseSessions&form_date=2015-03-26
53. Edward Alan Fox, Hamed Alhoori, Tarek Kanan, Sagnik Raychoudhury, Mohammed Samaka, Richard Furuta, Lee Giles, Krishna Roychowdhury, Sumaya Al-Maadeed, John Impagliazzo, Susan Lukesh, Myrna Tabet, and Asad Nafees. Electronic Library Institute-SeerQ

(ELISQ). Qatar Foundation Annual Research Conference Proceedings: Vol. 1, ITOP0243. DOI: 10.5339/qfarc.2014.ITOP0243. Published online: 13 Nov 2014. http://www.qscience.com/doi/abs/10.5339/qfarc.2014.ITOP0243

54. Alhoori, Hamed; Furuta, Richard; Tabet, Myrna; Samaka, Mohammed; and Fox, Edward A. Altmetrics for Country-Level Research Assessment. In Proc. of the 16th International Conference on Asia-Pacific Digital Libraries, ICADL 2014, The Emergence of Digital Libraries - Research and Practices, Chiang Mai, Thailand, Nov. 5-7, 2014. Lecture Notes in Computer Science Vol. 8839. Pp. 59-64. Springer. Editors: Tuamsuk, Kulthida; Jatowt, Adam; Rasmussen, Edie. 2014. http://dx.doi.org/10.1007/978-3-319-12823-8_7
55. Sherif Elmeligy Abdelhamid, Maksudul Alam, Richard Alo, Shaikh Arifuzzaman, Pete Beckman, Tirtha Bhattacharjee, Hasanuzzaman Bhuiyan, Keith Bisset, Stephen Eubank, Albert Esterline, Edward A. Fox, Geoffrey Fox, S.M.Shamimul Hasan, Harshal Hayatnagarkar, Maleq Khan, Christopher Kuhlman, Madhav V. Marathe, Natarajan Meghanathan, Henning Mortveit, Judy Qiu, S. S. Ravi, Zalia Shams, Ongard Sirisaengtaksin, Samarth Swarup, Anil Vullikanti and Tak-Lon Wu. CINET 2.0: A CyberInfrastructure for Network Science. The 10th IEEE International Conference on e-Science, Guaruja, Brazil, Oct. 20-24, 2014
56. Monika Akbar, Clifford A. Shaffer, Weiguo Fan, Edward A. Fox, "Recommendation Based on Deduced Social Networks in an Educational Digital Library", International Conference on Digital Libraries (DL 2014), London, Sept. 2014 (long paper, acceptance rate < 25%). In Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '14). IEEE Press, Piscataway, NJ, USA, 29-38. http://dl.acm.org/citation.cfm?id=2740769.2740776
57. Yinlin Chen, Edward A. Fox, "Using ACM DL paper metadata as an auxiliary source for building educational collections", International Conference on Digital Libraries (DL 2014), London, Sept. 2014 (short paper, acceptance rate 32%). In Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries. IEEE Press, Piscataway, NJ, USA, 137-140. http://dl.acm.org/citation.cfm?id=2740769.2740791
58. Edward A. Fox. Improving the ETD landscape. 30 minute plenary slot at Electronic Theses and Dissertations 2014 (ETD 2014), July 23-25, 2014, Leicester, UK
59. Andrea Kavanaugh, Steven Sheetz, John Tedesco, Sandoval-Almazan Rodrigo, and Edward Fox. Media Use during Conflicts: Gratifications & Efficacy during 2012 Mexican Elections. 15th Annual International Conference on Digital Government Research (dg.o 2014), Aguascalientes City, Mexico, June 18-21, 2014
60. Thiago A. Godoi, Ricardo da S. Torres, Ariadne M.B.R. Carvalho, Marcos A. Goncalves, Anderson A. Ferreira, Weiguo Fan, and Edward A. Fox. 2013. A relevance feedback approach for the author name disambiguation problem. In Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries (JCDL '13), July 22-26. ACM, New York, NY, USA, 209-218. DOI=10.1145/2467696.2467709 http://doi.acm.org/10.1145/2467696.2467709
61. Lin Tzy Li, Otavio A.B. Penatti, Edward A. Fox, and Ricardo da Silva Torres. 2013. Domain-specific image geocoding: a case study on Virginia Tech building photos. In Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries (JCDL '13), July 22-26. ACM, New York, NY, USA, 363-366. DOI=10.1145/2467696.2467727 http://doi.acm.org/10.1145/2467696.2467727
62. S.M.Shamimul Hasan, Keith Bisset, Edward A. Fox, Kevin Hall, Jonathan P. Leidig, Madhav V. Marathe. An Extensible Digital Library Service to Support Network Science. International Conference on Computational Science, Proceedings of International Conference on Computational Science, Barcelona, Spain, June 5-7, 2013
63. Seungwon Yang, Haeyong Chung, Xiao Lin, Sunshin Lee, Liangzhe Chen, Andrew Wood, Andrea L. Kavanaugh, Steven D. Sheetz, Donald J. Shoemaker, and Edward A. Fox. (2013). PhaseVis1: What, when, where, and who in visualizing the four phases of emergency management through the lens of social media. In J. Geldermann and T. Müller S. Fortier F. F. T. Comes (Eds.), ISCRAM 2013 Conference Proceedings - 10th International Conference on Information Systems for Crisis Response and Management (pp. 912–917). KIT; Baden-Baden, Germany, May 2013: Karlsruher Institut fur Technologie. Research-in-Progress paper. http://idl.iscram.org/files/yang/2013/1122_Yang_etal2013.pdf
64. Robert E. Beck, Edward Carr, Wingyan Chung, Edward Fox, and Christine Nass. 2013. Computing in context (abstract only). In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, 734-734. DOI=10.1145/2445196.2445423 http://doi.acm.org/10.1145/2445196.2445423

65. Sherif Abdelhamid, Richard Alo, S M Arifuzzaman, Pete Beckman, Md Hasanuzzaman Bhuiyan, Keith Bisset, Edward Fox, Geoffrey Fox, Kevin Hall, S.M.Shamimul Hasan, Anurodh Joshi, Maleq Khan, Chris Kuhlman, Spencer Lee, Jonathan P. Leidig, Hemanth Makkapati, Madhav Marathe, Henning Mortveit, Judy Qiu, Ravi S. S., Zalia Shams, Ongard Sirisaengtaksin, Rajesh Subbiah, Samarth Swarup, Nick Trebon, Anil Vullikanti, and Zhao Zhao. CINET: A CyberInfrastructure for Network Science. Proceedings of 8th IEEE Conf. eScience 2012, Chicago, IL, Oct. 8-12, 2012, pp. 1-8, http://www.ci.uchicago.edu/escience2012/pdf/CINet-A_CyberInfrastructure_for_Network_Science.pdf
66. Nathaniel Short, A. Lynn Abbott, Michael Hsiao, Edward Fox. Temporal Analysis of Fingerprint Impressions. In Proc. IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2012), Washington, DC, Sept. 23-26, 2012
67. Nathaniel Short, A. Lynn Abbott, Michael Hsiao, Edward Fox. Robust Feature Extraction in Fingerprint Images using Ridge Model Tracking. In Proc. IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2012), Washington, DC, Sept. 23-26, 2012, Runner-up for Honeywell Best Student Paper Award
68. Sung Hee Park, Roger W. Ehrich, Edward A. Fox. A Hybrid Two-Stage Approach for Discipline-Independent Canonical Representation Extraction from References. Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 285-294, http://dx.doi.org/10.1145/2232817.2232871
69. Monika Akbar, Clifford A. Shaffer, Edward A. Fox. Deduced Social Networks for an Educational Digital Library. Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 43-46, http://dx.doi.org/10.1145/2232817.2232828
70. Yinlin Chen, Paul Logasa Bogen II, Haowei Hsieh, Edward Fox, Lillian Cassel. Categorization of Computing Education Resources with Utilization of Crowdsourcing. Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 121-124, http://dx.doi.org/10.1145/2232817.2232840
71. (ISCRAM 2012). Apr. 22-25. Vancouver, Canada, 10 pages Andrea L. Kavanaugh, Steven D. Sheetz, Riham Hassan, Seungwon Yang, Hicham G. Elmongui, Edward A. Fox, Mohamed Magdy, and Donald Shoemaker. (2012). Between a rock and a cell phone: Communication and information technology use during the 2011 Egyptian uprising. In Z. Franco and J. R. L. Rothkrantz (Eds.), ISCRAM 2012 Conference Proceedings - 9th International Conference on Information Systems for Crisis Response and Management. Vancouver, BC, Apr. 22-25: Simon Fraser University. 10 pages. http://idl.iscram.org/files/kavanaugh/2012/138_Kavanaugh_etal2012.pdf
72. Jonathan P. Leidig, Edward A. Fox, and Madhav Marathe. Simulation Tools for Producing Metadata Description Sets Covering Simulation-Based Content Collections. International Conference on Modeling, Simulation, and Identification (MSI). Pittsburgh, PA, Nov 7-9, 2011.
73. Nathan Short, Lynn Abbott, Edward Fox, and Michael Hsiao. Latent Fingerprint Segmentation. Proc. 4th International Conference on Imaging for Crime Detection and Prevention (ICDP-11), 3-4 Nov. 2011, London
74. Kevin Hoyle, Nathan Short, Michael Hsiao, Lynn Abbott, and Edward Fox. Minutiae + Friction Ridges = Triplet-Based Features for Determining Sufficiency in Fingerprints. Proc. 4th International Conference on Imaging for Crime Detection and Prevention (ICDP-11), 3-4 Nov. 2011, London
75. Nathaniel J. Short, A. Lynn Abbott, Michael S. Hsiao, and Edward A. Fox. A Bayesian Approach to Fingerprint Minutia Localization and Quality Assessment using Adaptable Templates, IJCB 2011, Washington, D.C., Oct. 11-13, 2011, 7 pages
76. Nadia Kozievitch, Jurandy Almeida, Ricardo Torres, Neucimar Leite, Marcos Goncalves, Uma Murthy, and Edward Fox. Towards a Formal Theory for Complex Objects and Content-Based Image Retrieval. Full paper for SBBD 2011, 3-6 Oct. 2011, Florianopolis, Brazil, 16 pages
77. Lois Delcambre, David Archer, Susan Price, Scott Britell, Uma Murthy, Edward Fox, and Lillian Cassel. Superimposing a Strand Map Over Lectures and Textbook Content (In A Database Class). In Proc. CCSC-NW, Oct. 7-8, 2011, Richland, WA
78. Sung Hee Park, Jonathan P. Leidig, Lin Tzy Li, Edward A. Fox, Nathan J. Short, Kevin E. Hoyle, A. Lynn Abbott, and Michael S. Hsiao. Experiment and Analysis Services in a Fingerprint Digital Library for Collaborative Research. In S. Gradmann et al. (Eds.): TPDL 2011, Berlin, LNCS 6966, pp. 179-191. Springer, Heidelberg (Sept. 2011), acceptance rate 19%

79. Monika Akbar, Weiguo Fan, Clifford A. Shaffer, Yinlin Chen, Lillian Cassel, Lois Delcambre, Daniel D. Garcia, Gregory W. Hislop, Frank Shipman, Richard Furuta, B. Stephen Carpenter II, Haowei Hsieh, Bob Siegfried, and Edward A. Fox. Digital Library 2.0 for Educational Resources. In S. Gradmann et al. (Eds.): TPDL 2011, Berlin, LNCS 6966, pp. 89-100. Springer, Heidelberg (Sept. 2011), acceptance rate 19%
80. Sung Hee Park and Edward A. Fox. Enriching the VT ETD-db System with References. In Proc. ETD 2011 - 14th International Symposium on Electronic Theses and Dissertations. Cape Town, South Africa, September 13-17, 2011. 8 pages
81. Sung Hee Park, Paul Mather, Kimberli Weeks, Collin Brittle, Edward A. Fox, Gail McMillan. VT ETD-db 2.0: Rewriting ETD-db System. In Proc. ETD 2011 - 14th International Symposium on Electronic Theses and Dissertations. Cape Town, South Africa. September 13-17, 2011. 8 pages
82. Srinivasan, Venkat, Magdy, Mohamed, and Fox, Edward. Enhanced Browsing System for Electronic Theses and Dissertations. In Proc. ETD 2011 - 14th International Symposium on Electronic Theses and Dissertations. Cape Town, South Africa. September 13-17, 2011.
83. Uma Murthy, Lin Tzy Li, Eric Hallerman, Edward A. Fox, Manuel A. Perez-Quinones, Lois M. Delcambre, and Ricardo da S. Torres. Use of Subimages in Fish Species Identification: A Qualitative Study. Proceedings of JCDL 2011, Ottawa, June 13-17, 185-194 (acceptance rate of 20-25%)
84. Leidig J, Fox E, Hall K, Marathe M, Mortviet H. SimDL: A Model Ontology Driven Digital Library for Simulation Systems. Short paper in Proceedings of JCDL 2011, Ottawa, June 13-17, 81-84, http://dx.doi.org/10.1145/1998076.1998091
85. Kavanaugh, A., Fox, E., Sheetz, S., Yang, S., Li, L.T., Whalen, T., Shoemaker, D., Natsev, A., Xie, L. Social Media Use by Government: From the routine to the critical. ACM 2011 Digital Government Research Conference (dg.o 2011), June 12-15, 2011 (College Park, MD). New York: ACM Press, 121-130.
86. Andrea Kavanaugh, Seungwon Yang, Steven D. Sheetz, and Edward A. Fox. Microblogging in crisis situations: Mass protests in Iran, Tunisia, Egypt. Paper for Workshop on Transnational HCI at CHI 2011, May 7-12, Vancouver, 6 pages
87. Wingyan Chung, Seungwon Yang, Edward A. Fox, and Steven D. Sheetz. Integrating Computational Thinking Into Information Systems and Other Curricula. In Proceedings AIS SIG-ED IAIM 2010 Conference, St. Louis, Missouri, Dec. 10-12, 8 pages
88. Peter Brusilovsky, Lillian Cassel, Lois Delcambre, Edward Fox, Richard Furuta, Daniel D. Garcia, Frank M. Shipman III, Paul Bogen and Michael Yudelson. Enhancing Digital Libraries with Social Navigation: The Case of Ensemble. In Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 116-123
89. Nadia P. Kozievitch, Ricardo da Silva Torres, Sung Hee Park, Edward A. Fox, Nathan Short, Lynn Abbott, Supratik Misra, Michael Hsiao. Rethinking Fingerprint Evidence Through Integration of Very Large x Digital Libraries. VLDL Workshop at 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2010), Glasgow, Sept. 6-10, 8 pages
90. Leidig J, Fox E, Marathe M, Mortveit H (2010) Epidemiology experiment and simulation management through schema-based digital libraries. In Proceedings of the 2nd DL.org workshop during the 14th European Conference on Digital Libraries. Glasglow, Scotland, September 6-10, 2010, 57-66
91. Edward A. Fox, Yinlin Chen, Monika Akbar, Clifford A. Shaffer, Stephen H. Edwards, Peter Brusilovsky, Daniel D. Garcia, Lois M.L. Delcambre, Felicia Decker, David W. Archer, Richard Furuta, Frank Shipman, Stephen Carpenter and Lillian Cassel. Ensemble PDP-8: Eight Principles for Distributed Portals. In Proc. JCDL/ICADL 2010, June 21-25, Gold Coast, Australia, 341-344. Acceptance rate 25%
92. Seungwon Yang, Haeyong Chung, Chris North, Edward A Fox. The Effect of Presenting Long Documents on a Large High-Resolution Display on Comprehension of Content and User Experience. Refereed paper for ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
93. Sung Hee Park, Nicholas Lynberg, Jesse Racer, Phil McElmurray, Edward A. Fox. HTML5 ETDs. Refereed paper for ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
94. Peter Brusilovsky, Lillian N. Cassel, Lois M. L. Delcambre, Edward A. Fox, Richard Furuta, Daniel D. Garcia, Frank M. Shipman III, Michael Yudelson: Social navigation for educational digital libraries. In Proc. International Conference on Computational Science, ICCS 2010, U. of

Amsterdam, The Netherlands, May 31 - June 2, also in Elsevier's Procedia 1(1): 2889-289, May 2010

95. Uma Murthy, Edward Fox, Naren Ramakrishnan, Andrea Kavanaugh, Steven Sheetz, Donald Shoemaker, and Venkataraghavan Srinivasan. Building an Ontology for Crisis, Tragedy, and Recovery. NKOS Workshop, ECDL 2009, 1 Oct. 2009, Corfu, Greece, http://fox.cs.vt.edu/talks/2009/20091001CTR-NKOS.ppt
96. Uma Murthy, Edward Fox, Yinlin Chen, Eric Hallerman, Ricardo Torres, Evandro Ramos and Tiago Falcao. Superimposed Image Description and Retrieval for Fish, Proc. ECDL 2009, 29 Sept. 2009, Corfu, Greece, http://fox.cs.vt.edu/talks/2009/20090929ECDL09SI.pdf (acceptance rate 22%)
97. Wingyan Chung, Edward A. Fox. LIKES: Educating the Next Generation of Knowledge Society Builders, 15th Americas Conference on Information Systems, AMCIS 2009, Aug. 6-9, San Francisco, http://www.amcis2009.org/
98. Fox, E. A., Lee, S. J., Velacso, J., Marchionini, G., Yang, S., Murthy, U. Curatorial Work and Learning in Virtual Environments: A Virtual World Project to Support the NDIIPP Community First Workshop on Innovation in Digital Preservation, JCDL 2009, June 19, 2009
99. Venkat Srinivasan and Edward Fox. Topical Categorization of Large Collections of Electronic Theses and Dissertations, Proc. ETD 2009, Pittsburgh, PA, June 10-11, 2009, http://fox.cs.vt.edu/talks/2009/20090611ETD2009ClassifyTalk.ppt
100. David W. Archer, Lois M. L. Delcambre, Fabio Corubolo, Lillian N. Cassel, Susan Price, Uma Murthy, David Maier, Edward A. Fox, Sudarshan Murthy, John McCall, Kiran Kuchibhotla, Rahul Suryavanshi: Superimposed Information Architecture for Digital Libraries. ECDL 2008: 88-99, http://www.springerlink.com/content/n0u7v25567814161/
101. Seungwon Yang, Jean Levy, Kevin Miller, Jeffrey Pomerantz, Sanghee Oh, Barbara Wildemuth, Edward A. Fox. Two Approaches to Enhance the Education for ETDs: Developing Educational Modules and Migrating the ETD Guide into a Community Wiki, refereed abstract accepted, leading to full paper for ETD 2008: The 11th International Symposium on Electronic Theses and Dissertations, June 4-6, Aberdeen Scotland
102. Jesse M. Heines, Kenneth J. Goldman, Jim Jeffers, Edward A. Fox, and Robert Beck. Interdisciplinary Approaches to Revitalizing Undergraduate Computing Education. In Proc. Consortium for Computing Sciences in Colleges - Northeastern Region, CCSCNE 2008, April 11-12, Staten Island, NY, http://localhost/~heines/academic/papers/2008ccscne/CCSCNE-2008-Panel.pdf
103. Seungwon Yang, Barbara M. Wildemuth, Seonho Kim, Uma Murthy, Jeffrey P. Pomerantz, Sanghee Oh, and Edward A. Fox. Further Development of a Digital Library Curriculum: Evaluation Approaches and New Tools. Springer-Verlag Berlin Heidelberg, LNCS4822, pp. 434-443, Proceedings of the 10th International Conference on Asian Digital Libraries, ICADL 2007, Dec. 10-13, Hanoi Vietnam, DOI: 10.1007/978-3-540-77094-7_55
104. Xiaoyan Yu, Manas Tungare, Weiguo Fan, Manuel Perez-Quinones, Edward A. Fox, William Cameron and Lillian Cassel. Using Automatic Metadata Extraction to Build a Structured Syllabus Repository. Springer-Verlag Berlin Heidelberg, LNCS4822, pp. 337-346, Proceedings of the 10th International Conference on Asian Digital Libraries, ICADL 2007, Dec. 10-13, Hanoi Vietnam, DOI: 10.1007/978-3-540-77094-7_43
105. Maristella Agosti, Nicola Ferro, Marcos Andre Goncalves, Barbara Lagoeiro Moreira. Towards a Reference Quality Model for Digital Libraries. 1st Workshop on Digital Library Foundations, ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, http://fox.cs.vt.edu/talks/2007/20070623JCDLdlfWkshpAgosti-Quality.ppt
106. Yi Ma, Edward A. Fox, and Marcos A. Goncalves. Personal Digital Library: PIM upon 5S Framework. First PhD Workshop on CIKM07 (PIKM07). Nov. 2007, Lisbon, pp. 117-124
107. Uma Murthy, Douglas Gorton, Ricardo Torres, Marcos Goncalves, Edward Fox, and Lois Delcambre. Extending the 5S Digital Library (DL) Framework: From a Minimal DL towards a DL Reference Model. 1st Workshop on Digital Library Foundations, ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, http://fox.cs.vt.edu/talks/2007/20070623JCDLdlfWkshpMurthy-5S.ppt
108. Jeffrey Pomerantz, Sanghee Oh, Barbara M. Wildemuth, Seungwon Yang, and Edward A. Fox. Digital Library Education in Computer Science Programs. ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, 177 - 178, http://doi.acm.org/10.1145/1255175.1255208

109. Ryan Richardson, Edward A. Fox. Using Bilingual ETD Collections to Mine Phrase Translations. ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, 352 - 353, http://doi.acm.org/10.1145/1255175.1255244, http://fox.cs.vt.edu/talks/2007/20070622JCDLpaperRichardsonFox.ppt
110. Xiaoyan Yu, Manas Tungare, Weiguo Fan, Manuel Perez-Quinones, Edward A. Fox, William Cameron, GuoFang Teng, and Lillian Cassel. Automatic Syllabus Classification. ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, 440 - 441, http://doi.acm.org/10.1145/1255175.1255265, http://fox.cs.vt.edu/talks/2007/20070622JCDLpaperYuSyllabi.ppt
111. Seungwon Yang, Sanghee Oh, Jeffrey P. Pomerantz, Barbara M Wildemuth, and Edward A. Fox: Improving Education and Understanding of NDLTD. In Proc. ETD 2007, Uppsala, Sweden, June 13-16, 2007, http://fox.cs.vt.edu/talks/2007/20070614ETDfoxYangDLcurric.ppt
112. Ryan Richardson and Edward A. Fox: Using Concept Maps in NDLTD as a Cross-Language Summarization Tool for Computing-Related ETDs. In Proc. ETD 2007, Uppsala, Sweden, June 13-16, 2007, http://fox.cs.vt.edu/talks/2007/20070613ETDfoxRichardsonCmaps.ppt
113. Manas Tungare, Xiaoyan Yu, William Cameron, GuoFang Teng, Manuel Perez-Quinones, Lillian Cassel, Weiguo Fan, Edward Fox. Towards a Syllabus Repository for Computer Science Courses, in Proc. ACM SIGCSE 2007, March 7-10, 2007, Covington, KY, http://doi.acm.org/10.1145/1227310.1227331
114. Seungwon Yang, Edward A. Fox, Barbara M. Wildemuth, Jeffrey Pomerantz, and Sanghee Oh. Interdisciplinary Curriculum Development for Digital Library Education. In Proc. ICADL 2006, Nov., Kyoto, Japan, LNCS 4312, Springer-Verlag Berlin Heidelberg, pp. 61-70, 2006, http://dx.doi.org/10.1007/11931584_9
115. Seonho Kim, Subodh Lele, Sreeram Ramalingam, Edward A. Fox. Visualizing User Communities and Usage Trends of Digital Libraries Based on User Tracking Information. In Proc. ICADL 2006, Nov., Kyoto, Japan, LNCS 4312, Springer-Verlag Berlin Heidelberg, 2006, pp. 111-120, http://dx.doi.org/10.1007/11931584_14
116. Edward A. Fox and N. Srinivas Vemuri, "ETANA-DL: Leveraging DL Technologies to Support Archaeology", presentation in ETANA (Electronic Tools and Ancient Near Eastern Archives) Workshop I, Nov. 17; Edward A. Fox and Linda Cantara, co-chairs, ETANA Workshop II, Nov. 18, in ASOR 2006, Washington, DC (refereed workshop)
117. Manuel A. Perez-Quinones, Edward Fox, Lillian Cassel, and Weiguo Fan. "Work in Progress: Personalizing a Course Website Using the NSDL", in Proceedings of the 36th Annual Frontiers in Education Conference (FIE 2006), Oct. 28-31, 2006, San Diego, California, pp. M3F3-M3F4
118. Barbara Lagoeiro Moreira, Marcos Andre Goncalves, Alberto H.F. Laender, and Edward A. Fox. 5SQual - A Quality Assessment Tool for Digital Libraries, in Proc. WDL 2006, 20 Oct. 2006, Florianopolis, SC, Brasil, 10 pages
119. Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, and Edward A. Fox. What is a Successful Digital Library? In Proc. ECDL 2006, Alicante, Spain, Sept. 17-21, 2006, Research and Advanced Technology for Digital Libraries, ISBN 978-3-540-44636-1, Lecture Notes in Computer Science, Volume 4172, ISSN 0302-9743, Springer Berlin / Heidelberg, 2006, pp. 208-219, http://dx.doi.org/10.1007/11863878_18, pp. 208-219, http://fox.cs.vt.edu/talks/2006/20060918ECDLsuccess.ppt
120. Uma Murthy, Ryan Richardson, Edward Fox, and Lois Delcambre. Enhancing Concept Mapping Tools Below and Above to Facilitate the Use of Superimposed Information. In Proc. CMC 2006, Costa Rica, Sept. 5-8, 2006.
121. Sudarshan Murthy, Uma Murthy, and Edward A. Fox. Using Superimposed and Context Information to Find and Re-find Sub-documents. Proc. Invitational Workshop on Personal Information Management, part of ACM SIGIR 2006, Aug. 10-11, 2006, Seattle, 4 pages
122. Deborah Duong, Ben Goertzel, Jim Venuto, Ryan Richardson, Edward Fox. Support Vector Machines to Weight Voters in a Voting System of Entity Extractors. In Proc. IEEE World Congress on Computational Intelligence (http://wcci2006.org), July 16-21, 2006, Vancouver, 4 pages
123. Lillian N. Cassel, Edward A. Fox. Sharing the wealth: publishing electronic resources. In Proceedings of the 11th annual conference on innovation and technology in computer science education (ITiCSE 2006), Bologna, Italy, June 26-28, 2006, p. 327, http://doi.acm.org/10.1145/1140124.1140234

124. Manas Tungare, Xiaoyan Yu, Manuel Perez-Quinones, Edward A. Fox, Weiguo Fan, and Lillian Cassel. Towards a Standardized Representation of Syllabi to Facilitate Sharing and Personalization of Digital Library Content. Full paper (10 pages) for Proc. of the 4th International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL@AH'06, see http://www.win.tue.nl/SW-EL/), June 21-23, Dublin, Ireland
125. Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, Ricardo da S. Torres, and Edward A. Fox. Exploring Digital Libraries: Integrating Browsing, Searching, and Visualization. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, 1-10, http://doi.acm.org/10.1145/1141753.1141755
126. Naga Srinivas Vemuri, Rao Shen, Sameer Tupe, Weiguo Fan, and Edward A. Fox. ETANA-ADD: An Interactive Tool for Integrating Archaeological DL Collections. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, 161-162, http://doi.acm.org/10.1145/1141753.1141783
127. Jeffrey Pomerantz, Barbara M. Wildemuth, Seungwon Yang, and Edward A. Fox. Curriculum Development for Digital Libraries. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, 175-184, http://doi.acm.org/10.1145/1141753.1141787
128. Edward A. Fox, Seungwon Yang, Barbara M. Wildemuth, and Jeffrey Pomerantz. Digital Library Curriculum Development: Enhancing Education and Comprehension of NDLTD. Paper accepted based on review of abstract, ETD 2006: Unlocking Access, 9th International Symposium on Electronic Theses and Dissertations, June 7-10, 2006, Quebec City, slides: http://www6.bibl.ulaval.ca:8080/etd2006/pages/papers/SP11_Ed_Fox.ppt
129. Seonho Kim, Seungwon Yang, and Edward A. Fox. Supply and Demand Analysis in NDLTD Based on Patron Specialty Survey and Contents Statistics. Paper accepted based on review of abstract, ETD 2006: Unlocking Access, 9th International Symposium on Electronic Theses and Dissertations, June 7-10, 2006, Quebec City, http://www6.bibl.ulaval.ca:8080/etd2006/pages/papers/SP11_SeonhoKim.pdf, http://www6.bibl.ulaval.ca:8080/etd2006/pages/papers/SP11_SeonhoKim.ppt
130. Kamini Santhanagopalan, Gail McMillan, and Edward A. Fox. Prototype for Preservation of International ETDs using LOCKSS and OAI-PMH. Paper accepted based on review of abstract, ETD 2006: Unlocking Access, 9th International Symposium on Electronic Theses and Dissertations, June 7-10, 2006, Quebec City
131. Edward Fox, Rao Shen, Srinivas Vemuri, Weiguo Fan, Linda Cantara. Joanne Eustis, James Flanagan. ETANA-DL: Leveraging digital library technologies to support archaeology. In Proc. CAA2006, Computer Applications and Quantitative Methods in Archaeology Annual Conference, Fargo, ND, April 18-21, 2006, http://www.caa2006.org/
132. Edward Fox and James W. Flanagan, "ETANA Digital Library: The Most Recent Developments and a Demonstration", Nov. 16-19, 2005, Philadelphia: talk on Thursday, plus running 9:00am - 4:00pm Saturday ETANA (Electronic Tools and Ancient Near Eastern Archives) Workshop
133. Baoping Zhang, Yuxin Chen, Weiguo Fan, Edward A. Fox, Marcos Andre Goncalves, Marco Cristo, Pavel Calado. Intelligent GP Fusion from Multiple Sources for Text Classification. In Proceedings of the 14th Conference on Information and Knowledge Management, CIKM 2005, 31st October - 5th November, 2005 Bremen, Germany, 477-484
134. Ricardo da Silva Torres, Alexandre X. Falcao, Baoping Zhang, Weiguo Fan, Edward A. Fox, Marcos Andre Goncalves, Pavel Calado. A new framework to combine descriptors for content-based image retrieval. In Proceedings of the 14th Conference on Information and Knowledge Management, CIKM 2005, 31st October - 5th November, 2005 Bremen, Germany, 335-336
135. Fernando A. Das Neves, Edward A. Fox, Xiaoyan Yu. Connecting topics in document collections with stepping stones and pathways. In Proceedings of the 14th Conference on Information and Knowledge Management, CIKM 2005, 31st October - 5th November, 2005 Bremen, Germany, 91-98
136. Vinod K. Lohani, Kumar Mallikarjunan, Mary Leigh Wolfe,Terry Wildman, Jeff Connor, John Muffo, Jenny Lo, Tamara W. Knott, G. V. Loganathan, Richard Goff, Mike Chang, John Cundiff, Greg Adel, Foster Agblevor, Mike Gregg, David Vaughan, Ed Fox, Hayden Griffin, Saied Mostaghimi. Work in Progress: Spiral Curriculum Approach to Reformulate Engineering Curriculum. In Proc. 35th ASEE/IEEE Frontiers in Education Conference. Indianapolis, IN, Oct. 19-22, 2005

137. Ryan Richardson, Edward A. Fox, and John Woods. Evaluating concept maps as a cross-language knowledge discovery tool for NDLTD. In Proceedings ETD2005: evolution through discovery, 8th International Symposium on Electronic Theses and Dissertations, Sydney, Sept. 28-30, 2005, http://fox.cs.vt.edu/talks/2005/20050927ETDevaluatingConceptMaps.ppt
138. Ananth Raghavan, Naga Srinivas Vemuri, Rao Shen, Marcos Andre Goncalves, Weiguo Fan, and Edward A. Fox. Incremental, Semi-automatic, Mapping-Based Integration of Heterogeneous Collections into Archaeological Digital Libraries: Megiddo Case Study. In Proceedings ECDL2005, Vienna, Sept. 18-23, 2005, 139-150, http://dx.doi.org/10.1007/11551362_13, http://fox.cs.vt.edu/talks/2005/20050919ECDLmegiddo.ppt
139. Rao Shen, Marcos Andre Goncalves, Weiguo Fan, and Edward A. Fox. Requirements Gathering and Modeling of Domain-Specific Digital Libraries with the 5S Framework: An Archaeological Case Study with ETANA. In Proceedings ECDL2005, Vienna, Sept. 18-23, 2005, 1-12, http://dx.doi.org/10.1007/11551362_1, http://fox.cs.vt.edu/talks/2005/20050919ECDLmodeling.ppt
140. Seonho Kim, Uma Murthy, Kapil Ahuja, Sandi Vasile, and Edward A. Fox. Effectiveness of Implicit Rating Data on Characterizing Users in Complex Information Systems. Short paper in Proceedings ECDL2005, Springer LNCS, Vienna, Sept. 18-23, 2005, 186-194, http://dx.doi.org/10.1007/11551362_17, http://fox.cs.vt.edu/talks/2005/20050920ECDLshk.ppt
141. Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, Dong Zhuang. SimFusion: Measuring Similarity using Unified Relationship Matrix. In Proc. SIGIR 2005, 28th Annual International ACM SIGIR Conference, Salvador, Brazil, August 15-19, 2005, 130-137, http://doi.acm.org/10.1145/1076034.1076059
142. Baoping Zhang, Yuxin Chen, Weiguo Fan, Edward A. Fox, Marcos Andre Goncalves, Marco Cristo, Pavel Calado. Intelligent fusion of structural and citation-based evidence for text classification. In Proc. SIGIR 2005, 28th Annual International ACM SIGIR Conference, Salvador, Brazil, August 15-19, 2005, 667-668 http://doi.acm.org/10.1145/1076034.1076181
143. Ryan Richardson and Edward A. Fox. Using Concept Maps as a Cross-Language Resource Discovery Tool. Short paper in Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries (JCDL2005), June 7-11, 2005, Denver, 256-257
144. Seonho Kim and Edward A. Fox. Interest-Based User Grouping Model for Collaborative Filtering in Digital Libraries. In Zhaoneng Chen, Hsinchun Chen, Qihao Miao, Yuxi Fu, Edward Fox, Ee-Peng Lim, eds. Digital Libraries: International Collaboration and Cross-Fertilization; Proceedings 7th International Conference on Asian Digital Libraries, ICADL 2004, Shanghai, Dec. 13-17, 2004. Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 3334, pp. 533-542, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3334&spage=533
145. Baoping Zhang, Marcos Andre Goncalves, Weiguo Fan, Yuxin Chen, Edward A. Fox, Pavel Calado, and Marco Cristo. Combining Structural and Citation-Based Evidence for Text Classification. In: CIKM 2004, Washington D.C. Proc. of the 13th Conference on Information and Knowledge Management, Nov. 8-13, Washington, D.C. ACM Press, 2004, 162-163
146. Odis Hayden Griffin, Edward A. Fox, Calvin J. Ribbens, Thomas D. Walker, Nathaniel J. Davis, Richard M. Goff, Jenny L. Lo, Vinod K. Lohani, Michael H. Gregg and Dwight Barnette. Work in Progress - A Freshman Course for Engineering and Computer Science Students. In Proc. Frontiers in Education 34th Annual Conf., FIE 2004, Savannah, GA, Oct. 20-23, 2004
147. Unni Ravindranathan, Rao Shen, Marcos Andre Goncalves, Weiguo Fan, Edward A. Fox, and James W. Flanagan. Prototyping Digital Libraries Handling Heterogeneous Data Sources - The ETANA-DL Case Study. Full paper in Research and Advanced Technology for Digital Libraries: Proceedings 8th European Conference, ECDL 2004, Bath, UK, September 12-17, 2004, eds. Rachel Heery and Liz Lyon, Lecture Notes in Computer Science, vol. 3232, Springer-Verlag GmbH, Berlin, 186-197, http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=3232&spage=186
148. Nithiwat Kampanya, Rao Shen, Seonho Kim, Chris North, and Edward A. Fox. Citiviz: A Visual User Interface to the CITIDEL System. Full paper in Research and Advanced Technology for Digital Libraries: Proceedings 8th European Conference, ECDL 2004, Bath, UK, September 12-17, 2004, eds. Rachel Heery and Liz Lyon, Lecture Notes in Computer Science, vol. 3232, Springer-Verlag GmbH, Berlin, 122-133, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3232&spage=122

149. Fan, W., Luo, M., Wang, L., Xi, W., and Fox, E. A. Tuning before feedback: Combining ranking discovery and blind feedback for robust retrieval. SIGIR 2004, 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, England, 25-29 July.
150. M. A. Goncalves, L. T. Watson, and E. A. Fox. Towards a Digital Library Theory: A Formal Digital Library Ontology. In Mathematical Formal Methods workshop, SIGIR 2004, July 29, 2004, Sheffield, England; updated version of 31 pages
151. S. Perugini, K. McDevitt, R. Richardson, M. Perez-Quinones, R. Shen, N. Ramakrishnan, C. Williams, and E. A. Fox. Enhancing Usability in CITIDEL: Multimodal, Multilingual, and Interactive Visualization Interfaces. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, pp. 315-324.
152. M. A. Goncalves, E. A. Fox, A. Krowne, P. Calado, A. Laender, A. S. da Silva, and B. Ribeiro-Neto. The Effectiveness of Automatically Structured Queries in Digital Libraries. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, pp. 98-107, winner of Best Student Paper award.
153. U. Ravindranathan, R. Shen, M. A. Goncalves, W. Fan, E. A. Fox, and J. W. Flanagan, ETANA-DL: A Digital Library For Integrated Handling Of Heterogeneous Archaeological Data. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, pp. 76-77.
154. Edward A. Fox, Gregory R. Crane, Stephen M. Griffin, Ronald L. Larsen, David M. Levy, David J. McArthur, and Sugimoto Shigeo. Digital libraries settling the score: 10 years hence and 10 before. In Proceedings of the 4th ACM/IEEE-CS joint conference on digital libraries: Global Reach and Diverse Impact (JCDL '04). Tucson, AZ, June 7-11, 2004. ACM, New York, NY, USA, 374-374. DOI=http://dx.doi.org/10.1145/996350.996439
155. Luo, M. and E. A. Fox. ETD search services. Proceedings of ETD 2004: Distributing knowledge worldwide through better scholarly communication, June 3-5, 2004, Lexington, KY.
156. W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan, W.Y. Ma, E.A. Fox. Link Fusion: A Unified Link Analysis Framework for Multi-type Inter-related Data Objects. In Proceedings of the Thirteenth International World Wide Web Conference, WWW2004, New York, U.S.A. 19-22 May 2004, 10 pages
157. Weiguo Fan; Gordon, M.D.; Pathak, P.; Wensi Xi; Fox, E.A.; Ranking function optimization for effective web search by genetic programming: an empirical study, in the Proceedings of 37th Hawaii International Conference on System Sciences (HICSS), 5-8 Jan. 2004, 105 - 112
158. Aaron Krowne and Edward A. Fox. An Architecture for Multischeming in Digital Libraries. In: Tengku Mohd Tengku Sembok, Halimah Badioze Zaman, Hsinchun Chen, Shalini R. Urs, Sung Hyon Myaeng, eds. Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access; Proceedings 6th International Conference on Asian Digital Libraries, ICADL 2003, Kuala Lumpur, Malaysia, Dec. 2003. Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2911, 563-577, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2911&spage=563
159. Baoping Zhang, Marcos Andre Goncalves, and Edward A. Fox. An OAI-Based Filtering Service for CITIDEL from NDLTD. In: Tengku Mohd Tengku Sembok, Halimah Badioze Zaman, Hsinchun Chen, Shalini R. Urs, Sung Hyon Myaeng, eds. Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access; Proceedings 6th International Conference on Asian Digital Libraries, ICADL 2003, Kuala Lumpur, Malaysia, Dec. 2003. Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2911, 590-601, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2911&spage=590
160. Li Wang, Weiguo Fan, Rui Yang, Wensi Xi, Ming Luo, Ye Zhou, Edward A. Fox, Ranking Function Discovery by Genetic Programming for Robust Retrieval, Text Retrieval Evaluation Conference-2003, Nov 17-23, NIST, Washington DC, 9 pages
161. R. Yang, L. Wang, W. Fan, W. Xi, M. Luo, Y. Zhou, E.A. Fox. VT at TREC-2003: The Web Track Report. In Proceedings of the Twelfth Text Retrieval Conference (TREC 2003). NIST Special Publication 500-252. Gaithersburg, Maryland, 18-21 November 2003, 3 pages.
162. Qinwei Zhu, Marcos Andre Goncalves, Rao Shen, Lillian Cassel, Edward A. Fox. Visual Semantic Modeling of Digital Libraries. Proc. 7th European Conference on Digital Libraries (ECDL 2003), 17-22 August, Trondheim, Norway, Springer-Verlag GmbH, Lecture Notes in

Computer Science, vol. 2769, 2004, 325-337, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2769&spage=325

163. Rohit Kelapure, Marcos Andre Goncalves, Edward A. Fox. Scenario-Based Generation of Digital Library Services. Proc. 7th European Conference on Digital Libraries (ECDL 2003), 17-22 August, Trondheim, Norway, Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2769, 2004, 263-275, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2769&spage=263
164. Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. Automatic Document Metadata Extraction using Support Vector Machines. Proc. JCDL'2003, Third Joint ACM / IEEE-CS Joint Conference on Digital Libraries, May 27-31, 2003, Houston, 37-48, DOI: 10.1109/JCDL.2003.1204842.
165. Pavel P. Calado, Marcos Andre Goncalves, Edward A. Fox, Berthier Ribeiro-Neto, Alberto H. F. Laender, Altigran S. da Silva, David C. Reis, Pablo A. Roberto, Monique V. Vieira, and Juliano P. Lage. The Web-DL Environment for Building Digital Libraries from the Web. Proc. JCDL'2003, Third Joint ACM / IEEE-CS Joint Conference on Digital Libraries, May 27-31, 2003, Houston, 346 - 357
166. Byron Marshall, Yiwen Zhang, Hsinchun Chen, Ann Lally, Rao Shen, Edward A. Fox, and Lillian N. Cassel. Convergence of Knowledge Management and E-Learning: the GetSmart Experience. Proc. JCDL'2003, Third Joint ACM / IEEE-CS Joint Conference on Digital Libraries, May 27-31, 2003, Houston, 135-146.
167. Marcos Andre Goncalves, Ganesh Panchanathan, Unnikrishnan Ravindranathan, Aaron Krowne, Edward A. Fox, Filip Jagodzinski, and Lillian Cassel. The XML Log Standard for Digital Libraries: Analysis, Evolution, and Deployment. Proc. JCDL'2003, Third Joint ACM / IEEE-CS Joint Conference on Digital Libraries, May 27-31, 2003, Houston, 312 - 314
168. E. Hoffman and E. Fox. Building Policy, Building Community: An Example from the US National Science, Technology, Engineering, and Mathematics Education Library (NSDL). Short paper in: Ee-Peng Lim, Schubert Foo, Chris Khoo, Hsinchun Chen, Edward Fox, Shalini Urs, Thanos Constantino, eds. Digital Libraries: People, Knowledge, and Technology; Proceedings 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Dec. 2002, 299-300. Springer, Lecture Notes in Computer Science 2555.
169. Wensi Xi, Edward A. Fox, Roy P. Tan, Jiang Shu. Machine Learning Approach for Homepage Finding Task. In Proceedings of the 9th String Processing and Information Retrieval Symposium (SPIRE 2002). Lisbon, Portugal, September, 2002.
170. Marcos Andre Goncalves, Paul Mather, Jun Wang, Ye Zhou, Ming Luo, Ryan Richardson, Rao Shen, Liang Xu, and Edward A. Fox, Java MARIAN: From an OPAC to a Modern Digital Library System. In Proceedings of the 9th String Processing and Information Retrieval Symposium (SPIRE 2002). Lisbon, Portugal, September, 2002.
171. Hussein Suleman and Edward A. Fox. Designing Protocols in Support of Digital Library Componentization. In "Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings", eds. Maristella Agosti and Constantino Thanos, LNCS 2458, Springer, pp. 568-582.
172. Wensi Xi, Ohm Sornil, Ming Luo, and Edward A. Fox. Hybrid Partition Inverted Files: Experimental Validation. In "Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings", eds. Maristella Agosti and Constantino Thanos, LNCS 2458, Springer, pp. 422-431.
173. Marcos Andre Goncalves, Ming Luo, Rao Shen, Mir Farooq Ali, and Edward A. Fox. An XML Log Standard and Tool for Digital Library Logging Analysis. In "Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings", eds. Maristella Agosti and Constantino Thanos, LNCS 2458, Springer, pp. 129-143.
174. Rao Shen, Jun Wang, Edward A. Fox. A Lightweight Protocol between Digital Libraries and Visualization Systems. JCDL Workshop on Visual Interfaces to Digital Libraries (p. 425 of Proc. JCDL 2002), July 18, 2002, Portland.
175. Ashwini Pande, Malini Kothapalli, Ryan Richardson, and Edward A. Fox Mirroring an OAI archive on the I2-DSI channel. Short paper in Proc. JCDL'2002, Second Joint ACM / IEEE-CS Joint Conference on Digital Libraries, July 14-18, 2002, Portland, pp. 293-294.
176. H. Anan, X. Liu, K. Maly, M. Nelson, M. Zubair, J. French, E. Fox, P. Shivakumar. Preservation and Transition of NCSTRL Using an OAI-Based Architecture. Short paper in Proc. JCDL'2002, Second Joint ACM / IEEE-CS Joint Conference on Digital Libraries, July 14-18, 2002,

Portland, pp. 181-182.

177. Marcos Andre Goncalves and Edward A. Fox. 5SL - A Language for Declarative Specification and Generation of Digital Libraries. Long paper in Proc. JCDL'2002, Second Joint ACM / IEEE-CS Joint Conference on Digital Libraries, July 14-18, 2002, Portland, pp. 263-272.
178. Jun Wang, Abhishek Agrawal, Anil Bazaz, Supriya Angle, Edward A. Fox, and Chris North. Enhancing the ENVISION Interface for Digital Libraries. Short paper in Proc. JCDL'2002 Second Joint ACM / IEEE-CS Joint Conference on Digital Libraries, July 14-18, 2002, Portland, pp. 275-276.
179. Wensi Xi, Ohm Sornil, and Edward A. Fox. Hybrid Partition Inverted Files for Large-Scale Digital Libraries. Proc. Digital Library: IT Opportunities and Challenges in the New Millennium, July 9-11, 2002, Beijing Library Press, Beijing, China, 404-418
180. Hussein Suleman and Edward A. Fox. Towards Universal Accessibility of ETDs: Building the NDLTD Union Archive. In Proc. ETD'2002, BYU, Provo, Utah, May 30 - June 1, 2002, preprint at http://rocky.dlib.vt.edu/~hussein/etd_2002/etd_2002_paper_final.pdf
181. Marcos Andre Goncalves, Ye Zhou, and Edward A. Fox. Providing Extended Services and Resources to the NDLTD Community. In Proc. ETD'2002, BYU, Provo, Utah, May 30 - June 1, 2002
182. Ing-Ray Chen and Edward Fox. Design and Analysis of A Set-Top Box for Video Streaming Services. In Proc. IMMCN'2002, Second International Workshop on Intelligent Multimedia Computing and Networking, March 8-12, 2002, Durham, NC.
183. Ohm Sornil and Edward A. Fox. Hybrid Partitioned Inverted Indices for Large-Scale Digital Libraries. In Proc. International Conference on Asian Digital Libraries, ICADL'2001, Bangalore, India, Dec. 10-12, 2001
184. Marcos Andre Goncalves, Robert K. France, and Edward A. Fox, MARIAN: Flexible Interoperability for Federated Digital Libraries. In Proceedings 5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'2001, September 4-8, 2001, Darmstadt, Germany, Lecture Notes in Computer Science, Springer-Verlag GmbH, vol. 2163 / 2001, pp. 173-186, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2163&spage=173
185. Marcos Andre Goncalves, Ali Zafer, Naren Ramakrishnan, and Edward A. Fox, Modeling and Building Personalized Digital Libraries with PIPE and 5SL. In Proceedings of the Joint DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries, 18-20 June 2001, Dublin, Ireland.
186. Marcos Andre Goncalves, Robert K. France, Edward A. Fox, Eberhard R. Hilf, Michael Hohlfeld, Kerstin Zimmermann, and Thomas Severiens. Flexible Interoperability in a Federated Digital Library of Theses and Dissertations. In Proceedings of the 20th World Conference on Open Learning and Distance Education, "The Future of Learning - Learning for the Future: Shaping the Transition", ICDE2001, Duesseldorf, Germany, 01 - 05 April 2001
187. Marcos Andre Goncalves, Robert K. France, Edward A. Fox, and Tamas E. Doszkocs. MARIAN: Searching and Querying Across Heterogeneous Federated Digital Libraries. In Proc. of First DELOS Workshop on "Information Seeking, Searching and Querying in Digital Libraries", 11-12 December 2000, Zurich, Switzerland. Talk: http://www.ndltd.org/talks/delosfox.ppt, paper: http://www.ercim.org/publication/ws-proceedings/DelNoe01/11_Fox.pdf
188. Canos, J. H., Fox, E.A., Goncalves, M.A. and France, R.K., NDLTD: una biblioteca digital global de tesis doctorales y de licenciatura. Actas de las I Jornadas de Bibliotecas Digitales. Valladolid, noviembre de 2000. Translation: NDLTD: a global digital library of theses and dissertations. Proceedings of the First Spanish Workshop on Digital Libraries (6-7 October, 2000, Valladolid, Spain), ISBN: 84-8448-066-6
189. Hussein Suleman, Edward A. Fox, Marc Abrams. "Building Quality into a Digital Library", http://purl.org/net/repository, short paper in Proc. ACM Digital Libraries '2000, June 4-7, 2000, San Antonio.
190. Fernando Das Neves and Edward A. Fox. "A study of user behavior in an immersive Virtual Environment for digital libraries", in Proc. ACM Digital Libraries '2000, June 4-7, 2000, San Antonio.
191. Edward A. Fox, Rachelle S. Heller, Anna Long, and David Watkins. "CRIM: Curricular Resources in Interactive Multimedia", in Proceedings ACM Multimedia '99, Orlando, FL, Nov. 1-5, 1999.

192. Constantinos Phanouriou, Neill Kipp, Ohm Sornil, Paul Mather, and Edward A. Fox, "A Digital Library for Authors: Recent Progress of the Networked Digital Library of Theses and Dissertations", in Proceedings ACM Digital Libraries '99, Berkeley, CA, Aug. 11-14, 1999.
193. E.A. Fox, G. McMillan, J.L. Eaton, "The Evolving Genre of Electronic Theses and Dissertations", in Proceedings Digital Documents Track of HICSS-32, Thirty-second Annual Hawaii International Conference on Systems Sciences (HICSS), Maui, HI - January 5-8, 1999, 20 pages, http://scholar.lib.vt.edu/theses/presentations/Hawaii/ETDgenreALL.pdf
194. Osman Balci, Cengiz Ulusarac, Poorav Shah, and Edward A. Fox (1998), "A Library of Reusable Model Components for Visual Simulation of the NCSTRL System," In Proceedings of the 1998 Winter Simulation Conference (Washington, DC, Dec. 13-16). IEEE, Piscataway, NJ, vol. 2, 1451-1459
195. Ghaleb Abdulla, Binzhang Liu, and Edward A. Fox, "Searching the WWW: Implications from studying different user behavior", in Proc. WebNet98 Conference, Orlando, FL, November 7-12, 1998, http://www.aace.org/conf/webnet
196. Binzhang Liu, Ghaleb Abdulla, Tommy Johnson, and Edward A. Fox, "Web Response Time and Proxy Caching", in Proc. WebNet98 Conference, Orlando, FL, November 7-12, 1998, http://www.aace.org/conf/webnet, http://memex.dlib.vt.edu/~liub/webnet.html, recognized as a "top paper".
197. Md. Habib, G. Abdulla, and E. Fox. "Web Traffic Chracterization with Time Zones: Seeking outside events that affect the traffic to a distance learning server", in Proc. International Symposium on Audio, Video, Image Processing and Intelligent Application (ISAVIIA-98), Germany, 17-21 August 1998. http://www.cs.vt.edu/~chitra/docs/ISAVIIA-98/final_i98.pdf
198. M. Bayraktar, C. Zhang, B. Vadapalli, N. Kipp, and E. Fox. "A Web Art Gallery", in Proc. Digital Libraries '98, The Third ACM Conf. on Digital Libraries, Pittsburgh, PA, June 23-26, 1998, 277-278.
199. G. Abdulla, E. A. Fox, M. Abrams, "Shared User Behavior on the World Wide Web", in Proc. WebNet97 Conference, Toronto, Canada Oct. 1997, http://www.AACE.org/conf/webnet http://www.cs.vt.edu/~chitra/docs/97webnet/
200. M. Kirschenbaum, E. Fox. 1997. Electronic Theses and Dissertations in the Humanities. In Proc. Joint Annual Conf. of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, ACH-ALLC'97, June 3-7, 1997, Queen's Univ., Kingston, Ontario.
201. L. Tinoco, E. Fox, N. D. Barnette. Online Evaluation in WWW-based Courseware. In SIGCSE '97, Proceedings of the 28th SIGCSE Technical Symp. on Computer Science Education, San Jose, 2/27-3/1/97, 194-198. http://ei.cs.vt.edu/papers/sigcse97.pdf
202. L. Tinoco, E. Fox, R. Ehrich, H. Fuks. 1996. QUIZIT: An interactive online quiz system for WWW-based instruction. In VII Proceedings of the Symposium of Educational Technology: Belo Horizonte, MG, Brazil, Nov 1996. http://ei.cs.vt.edu/papers/QUIZIT9611.pdf
203. S. Williams, M. Abrams, C. Standridge, G. Abdulla, E. Fox. Removal Policies in Network Caches for World-Wide Web Documents, in Proc. ACM SIGCOMM '96, Stanford U., Palo Alto, CA, Aug. 28-30, 1996, 293-305. See full paper and slides in various versions: http://www.cs.vt.edu/~nrg/williams.html
204. L. Nowell, R. France, D. Hix, L. Heath, and E. Fox. Visualizing Search Results: Some Alternatives to Query-document Similarity, in Proc. SIGIR'96, Zurich, Switzerland, Aug. 18-22, 1996, 67-75.
205. M. Abrams, C. R. Standridge, G. Abdulla, S. Williams and E. A. Fox. Caching Proxies: Limitations and Potentials, in Proc. 4th International World-Wide Web Conference, Boston, Dec. 1995, 119-133 http://www.w3j.com/1/abrams.155/paper/155.html
206. W. Wake and E. Fox. SortTables: A Browser for a Digital Library. In Proc. 4th Int. Conf. on Information and Knowledge Management, CIKM '95, Baltimore, MD, Nov. 28 - Dec. 2, 1995, 175-181.
207. M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Ribler and E. Fox. Multimedia Traffic Analysis Using Chitra95. In Proc. 3rd Int. Multimedia Conf. and Exhibition, Multimedia '95, San Francisco, Nov. 5-9, 1995, 267-276. http://www.cs.vt.edu/~chitra/docs/95multimediaAWAFPR/
208. W. Wake, B. D. Wake, and E. Fox. Improving Responsiveness in Interactive Applications Using Queues. In Proc. Pattern Languages of Program Design 2, PLOP '95.

209. E. Fox and D. Barnette. Improving Education through a Computer Science Digital Library with Three Types of WWW Servers. In Proc. Second International WWW '94: Mosaic and the Web, WWW'94, Chicago, IL, Oct. 17-20, 1994.
210. K. Maly, J. French, A. Selman and E. Fox. The Wide Area Technical Report Service. In Proc. Second International WWW '94: Mosaic and the Web, WWW'94, Chicago, IL, Oct. 17-20, 1994, 523-533.
211. H. Gladney, E. Fox, Z. Ahmed, R. Ashany, N. Belkin, and M. Zemankova. Digital Library: Gross Structure and Requirements: Report from a March 1994 Workshop. Digital Libraries '94, June 19-21, 1994, College Station, TX, ed. J. Schnase, J. Leggett, R. Furuta, T. Metcalfe, 101-107.
212. E. Fox and G. Abdulla. Digital Video Delivery for a Digital Library in Computer Science. High-Speed Networking and Multimedia Computing Workshop, IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Feb. 6-10, 1994, San Jose, CA, 7 pages.
213. F. Can, C.D. Snavely, E.A. Fox, and R.K. France. Incremental Clustering for Very Large Document Databases: Initial MARIAN Experience, In Proceedings of the Eighth International Symposium on Computer and Information Sciences (ISCIS 8), eds. L. Gun, R. Onvural, and E. Gelenbe, Istanbul, 1-3 Nov. 1993. (8 pages)
214. Y. Qian, E. Fox, and W. Farley. An Object-Oriented Database for the Display Measurement and Analysis System Proc. CIKM 93, 2nd International Conference on Information and Knowledge Management, Nov. 1-5, 1993, Washington, D.C., ACM Press, 384-392.
215. E. Fox, R. France, E. Sahle, A. Daoud, and B. Cline. Development of a Modern OPAC: From REVTOLC to MARIAN. Proc. 16th Annual Intern'l ACM SIGIR Conf. on R & D in Information Retrieval, SIGIR '93, Pittsburgh, PA, June 27 - July 1, 1993, 248-259.
216. D. Brueni, B. Cross, E. Fox, L. Heath, D. Hix, L. Nowell, and W. Wake. What if there were desktop access to the Computer Science Literature?, Proc. 21st Annual Computer Science Conference, ACM CSC '93, Feb.16-18, 1993, Indianapolis, IN, 15-22.
217. S. Patel, G. Abdulla, M. Abrams, and E. Fox. NMFS: Network Multimedia File System Protocol, Third International Workshop on Network and Operating System Support for Digital Audio and Video, Nov. 12-13, 1992, La Jolla, CA. Printed in P. Venkat Rangan, Ed., Network and Operating System Support for Digital Audio and Video, Lecture Notes in Computer Science 712, Springer-Verlag, Berlin Heidelberg, 1993, ISBN 3-540-57183-3 and 0-387-57183-3, 328-333.
218. E. A. Fox, Qi Fan Chen, and L. S. Heath. A Faster Algorithm for Constructing Minimal Perfect Hash Functions. Proc. 15th Intern'l Conf. on R & D in Information Retrieval, SIGIR '92, June 21-24, 1992, Copenhagen, Denmark, 266-273.
219. E. Fox, R. France, M. Koushik, J.-L. Menezes, Q.-F. Chen, A. Daoud, and J. Nutter. CODER: A Retrieval and Hypertext System using SGML and a Lexicon. Proc. ACH/ ALLC '91, The Combined 11th Int'l Conf. on Computers and the Humanities and 18th Int'l Conf. of the ALLC, Mar. 17-21, 1991, Arizona State Univ., Tempe, AZ. 5 pages.
220. A. Narasimhan and E. Fox. Application of Multimedia in Simulation. Proc. The Twentieth Annual Virginia Computer Users Conference, Sept. 14- 16, 1990, VPI&SU, Blacksburg, VA, 166-180.
221. E. Fox, Q.-F. Chen, A.M. Daoud, and L.S. Heath. Order preserving minimal perfect hash functions and information retrieval. Proc. SIGIR 90, 13th Int'l Conf. on R & D in Information Retrieval, ed. Jean-Luc Vidick, Sept. 5-7, 1990, Brussels, Belgium, 279-311.
222. E. Fox. A Retrospective on CODER: A Testbed for Artificial Intelligence Methods in Information Retrieval. 1990 Spring Symp. Series, Symposium on Text-Based Intelligent Systems, March 27-29, 1990, Stanford Univ., Palo Alto, CA, 81-82.
223. R. France, E. Fox, J.T. Nutter, Q.F. Chen. Building A Relational Lexicon for Text Understanding and Retrieval. Proc. First Int'l Language Acquisition Workshop, Aug. 21, 1989, Detroit, MI. 6 pages.
224. Nutter, J.T., Fox, E.A., and Evens, M. Building a lexicon from machine-readable dictionaries for improved information retrieval. The Dynamic Text: 16th ALLC and 9th ICCH International Conferences, Toronto, Ontario, June 6-9, 1989. [No proceedings were published; this paper was selected for journal publication and appeared in: Literary and Linguistic Computing, 1990, 5(2): 129-138.]
225. E. Fox, Q. Chen, L. Heath, and S. Datta. A More Cost Effective Algorithm for Finding Perfect Hash Functions. Proc. ACM 1989 Computer Science Conference, February 21-23, 1989, Louisville, KY, 114-122.
226. E. Fox, G. Nunn, and W. Lee. Coefficients for Combining Concept Classes in a Collection. Proc. 11th Int'l Conf.on Research and Development in Information Retrieval, June 13-15, 1988, Grenoble, France, 291-307.

227. E. Fox, M. Weaver, Q. Chen, and R. France. Implementing a Distributed Expert-Based Information Retrieval System. Proc. RIAO (Recherche d'Informations Assistee par Ordinateur) 88: User-Oriented Text and Image Handling, March 21-24, 1988, Cambridge, MA, 708-726.
228. E. Fox, J. Nutter, T. Ahlswede, M. Evens, and J. Markowitz. Building a Large Thesaurus for Information Retrieval. Proceedings Second Conference on Applied Natural Language Processing, Feb. 9-12, 1988, Austin, Texas, 101-108.
229. E. Fox and S. Winett. Extending Expert System Techniques with Information Retrieval Methods to Select Foster Homes. Proc. Third Ann. Expert Systems in Gov. Conf., Oct. 19-23, 1987, Washington, D.C., 26-32.
230. E. Fox. An Intelligent Information System for Electronic Mail Digests. Proceedings of the 50th Annual Meeting of the American Society for Information Science, Oct. 4-8, 1987, Boston, MA, 24: 74-78.
231. S. Winett and E. Fox. FOster Care Expert System: A Model Project to Select Foster Care Homes for Children. Databases in the Humanities and Social Sciences 4, Proc. of the Int. Conf. on Databases in the Humanities and Social Sciences, Auburn University, Montgomery AL, July 11-13,1987, ed. Lawrence J. McCrank, Learned Information, Inc., Medford, NJ.
232. E. Fox and Q. Chen. Text Analysis in the CODER System. Proceedings Fourth Annual USC Computer Science Symposium: Language and Data in Systems, April 8, 1987, Columbia, SC, 7-14.
233. R. France and E. Fox. Knowledge Structures for Information Retrieval: Representation in the CODER Project. Proc. IEEE Expert Systems in Government Symposium, Oct. 20-24, 1986, McLean, VA, 135-141.
234. E. Fox. Expert Retrieval for Users of Computer Based Message Systems. Proceedings 49th Annual Meeting American Society for Information Science, Sept. 28 - Oct. 2, 1986, Chicago, IL, 23: 88-95.
235. E. Fox and S. Birch. A UNIX Micro Requirement for Computer Science Majors. Proc. NECC '86: 7th National Educational Computing Conf., June 4-6, 1986, San Diego, CA, 157-160.
236. E. Fox and R. France. Architecture of a Distributed Expert System for Composite Document Entry, Analysis, Representation, and Retrieval. Proc. 3rd Ann. USC Computer Sci. Sym.; Knowledge-Based Systems: Theory and Applications, Mar. 31-Apr. 1, 1986, Columbia SC, 13 pg.
237. S. Winett and E. Fox. Using Information Retrieval Techniques in an Expert System. Proceedings Second Conference on Artificial Intelligence Applications: The Engineering of Knowledge-Based Systems, Dec. 11- 13, 1985, Miami Beach, FL, 230-235.
238. S. Winett and E. Fox. Information Retrieval Techniques in an Expert System for Foster Care. Proc. Expert Systems in Gov.Symp., Oct. 23-25, 1985, McLean, VA, 391-396.
239. E. Fox. Analysis and Retrieval of Composite Documents. ASIS '85, Proceedings of the 48th American Society for Information Science Annual Meeting, October 20-24, 1985, Las Vegas, Nevada, 22: 54-58.
240. E. Fox. Composite Document Extended Retrieval: An Overview. Proc. Research and Devel. in Information Retrieval, Eighth Annual International ACM SIGIR Conf., June 5-7, 1985, Montreal, 42-53.
241. E. Fox. Composite Document Search in a Networked Society. Telecommunications & Networking: supplying the missing link, Collected papers of the American Society for Information Science 14th Mid-Year Mtg., May 19-22, 1985, Fort Lauderdale, FL, Fiche 2, Paper 11, 12 pages.
242. E. Fox. Information Retrieval with Undergraduate Microcomputers. The Micro Revolution, Collected papers Amer. Soc. for Info. Sci.13th ASIS Mid-Year Mtg, May 20-23, 1984, Bloomington, IN, Fiche 5: 253-264.
243. E. Fox. Combining Information in an Extended Automatic Information Retrieval System for Agriculture. The Infrastructure of an Information Society (Proc. 1st Int'l Information Conf. Egypt, 13-16 Dec. 1982, Cairo), B. El-Hadidy & E.E. Horne (editors), North Holland, 1984, 449-466.
244. G. Salton, E. Fox, and H. Wu. An Automatic Environment for Boolean Information Retrieval. In Information Processing 83 (Proc. 1983 IFIP Paris Congress), R.E.A. Mason (ed.), North-Holland, 1983, 755-762.
245. E. Fox. Implementing SMART for Minicomputers Via Relational Processing with Abstract Data Types. Jt Proc. SIGSMALL Symp. on Small Systems and SIGMOD Workshop on Small Data Base Systems, Oct. 1981, Orlando, FL, ACM SIGSMALL Newsletter, 7(2): 119-129.

REFEREED POSTERS:

1. Muntabir Hasan Choudhury, Jian Wu, William A. Ingram, and Edward A. Fox. A Heuristic Baseline Method for Metadata Extraction from Scanned Electronic Theses and Dissertations. Proc. JCDL 2020, Aug. 1-5, Virtual Event, China, poster, 2 pages, https://doi.org/10.1145/3383583.3398590
2. Prashant Chandrasekar, Kris Wernstedt, Edward A. Fox, Pamela Murray-Tuite (2020). Hurricane Irma: Multiple Avenues of Study. Poster, with abstract in Proceedings International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020), May 2020, page 1166, to be presented in May 2021 at ISCRAM 2021. Blacksburg, VA.
3. Ziqian Song, Austin Spencer, Taylor Thackaberry, Kayley Bogemann, Shane Burchard, Jessie Butler, Liuqing Li, Kris Wernstedt, Pamela Murray-Tuite, Edward A. Fox (2020). Acknowledgement to CS4624 students Austin Spencer, Taylor Thackaberry, Kayley Bogemann, Shane Burchard, and Jessie Butler. A comparison of people's use of Twitter in Puerto Rico Earthquake and Hurricane Maria. Poster, with abstract in Proceedings International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020), May 2020, page 1163, to be presented in May 2021 at ISCRAM 2021. Blacksburg, VA
4. Prashant Chandrasekar, Edward A. Fox (2019). Interactive Digital Library Support for Workflows: Applying Machine Learning in Network Science. Poster in Algorithms That Make You Think, Fourth Annual Virginia Tech Workshop on the Future of Human-Computer Interaction, April 11-12, 2019. Blacksburg, VA
5. Saket Vishwasrao, Zhiwu Xie and Edward Fox. Web Archiving Through In-Memory Page Cache. Poster at WADL (Web Archiving and Digital Libraries) 2017, a workshop held in conjunction with JCDL 2017, Toronto, Canada, June 19-23, 2017.
6. Yinlin Chen, Zhiwu Xie, and Edward A. Fox. A Library to Manage Web Archive Files in Cloud Storage. Poster presented at WADL 2016: Third International Workshop on Web Archiving and Digital Libraries, June 22-23, 2016. In connection with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, http://fox.cs.vt.edu/wadl2016.html
7. Sunshin Lee and Edward A. Fox. Archiving and Analyzing Tweets and Webpages with the DLRL Hadoop Cluster. Poster presented at WADL 2016: Third International Workshop on Web Archiving and Digital Libraries, June 22-23, 2016. In connection with ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, http://fox.cs.vt.edu/wadl2016.html
8. Mohamed Farag, Pranav Nakate and Edward A. Fox. Big Data Processing of School Shooting Archives. Poster, in Proc. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2016, http://www.jcdl2016.org/), Rutgers Univ., Newark, NJ, June 19-23, 2016, 271-272, http://dx.doi.org/10.1145/2910896.2925466
9. Sunshin Lee, Mohamed Magdy, Richard Gruss, Tarek Kanan, Xuan Zhang, and Edward A. Fox. Enhanced problem-based learning connecting big data research with classes. Poster at HPC Day 2016, hosted by Virginia Tech's Advanced Research Computing, Blacksburg, VA, 11 April 2016
10. Chen, Yinlin, Edward A. Fox, and Tingting Jiang. "Performance of a Cloud-Based Digital Library." Poster in Proceedings Digital Libraries: Providing Quality Information: 17th International Conference on Asia-Pacific Digital Libraries, ICADL 2015, Seoul, Korea, December 9-12, 2015. ISBN 978-3-319-27974-9. Vol. 9469, pages 298-299. Springer, 2016. https://www.springer.com/gp/book/9783319279732
11. Tarek Kanan, Sagnik Ray Choudhury, C. Lee Giles, Prashant Chandrasekar and Edward Fox. Digital Library and Archiving for Qatar. 5 minute lightning talk, with poster in Proc. Web Archiving and Digital Libraries Workshop (WADL 2015) - Joint Conference on Digital Libraries (JCDL). Knoxville, TN. 24 June 2015
12. Zhiwu Xie, Prashant Chandrasekar, and Edward A. Fox. Using Transactional Web Archives to Handle Server Errors. Poster in Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2015). June 21-25, 2015. Knoxville, TN, pp. 241-242. http://dx.doi.org/10.1145/2756406.2756955

13. Sunshin Lee, Mohammed Farag, Tarek Kanan, and Edward A. Fox. Read between the lines: A Machine Learning Approach for Disambiguating the Geo-location of Tweets. Proc. JCDL 2015. June 21-25, 2015. Knoxville, TN, 273-274. http://dx.doi.org/10.1145/2756406.2756971
14. S.M. Shamimul Hasan, Sandeep Gupta, Edward A. Fox, Keith Bisset, Madhav V. Marathe. Data Mapping Framework in a Digital Library with Computational Epidemiology Datasets. Poster. In Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '14). International Conference on Digital Libraries (DL 2014), London, Sept. 2014. IEEE Press, Piscataway, NJ, USA, 449-450. http://dl.acm.org/citation.cfm?id=2740769.2740862
15. Sheetz, Steven; Kavanaugh, Andrea; Fox, Edward; Elmongui, Hicham; Hassan, Riham; Yang, Seugwon; Magdy, Mohammed; Shoemaker, Donald. Information Uses and Gratifications in Crisis: Student Perceptions since the Egyptian Uprising. Proceedings of the 11th International ISCRAM Conference, University Park, Pennsylvania, USA, May 18-21, 2014
16. Mohamed M. G. Farag and Edward A. Fox. Intelligent Event Focused Crawling. Proceedings of the 11th International ISCRAM Conference, University Park, Pennsylvania, USA, May 18-21, 2014
17. Hamed Alhoori, Carole Thompson, Richard Furuta, John Impagliazzo, Edward A. Fox, Mohammed Samaka, Somaya Al-Maadeed. The Evolution of Scholarly Digital Library Needs in an International Environment: Social Reference Management Systems and Qatar. In Proc. of the 15th International Conference on Asia-Pacific Digital Libraries, ICADL 2013, Bangalore, India, Dec. 9-11. Digital Libraries: Social Media and Community Networks, Springer Lecture Notes in Computer Science Volume 8279, 2013, pp 180-181. DOI 10.1007/978-3-319-03599-4_24, http://link.springer.com/chapter/10.1007/978-3-319-03599-4_24
18. Peter Brusilovsky, Yiling Lin, Chirayu Wongchokprasitti, Scott Britell, Lois Delcambre, Richard Furuta, Kartheek Chiluka, Lilian Cassel, Edward A. Fox. Social Navigation Support for Groups in a Community-Based Educational Portal. In Proc. TPDL 2013: 17th Int'l Conf. on Theory and Practice of Digital Libraries. Malta, 22-26 Sept., 429-433
19. Jonathan P. Leidig and Edward A. Fox. 2013. Formal foundations for systematic digital library generation. In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries (JCDL '13). ACM, New York, NY, USA, 391-392. DOI=10.1145/2467696.2467774 http://doi.acm.org/10.1145/2467696.2467774
20. Jennifer Murphy, Donald J. Shoemaker, Seungwon Yang, Andrea Kavanaugh, Steven D. Sheetz, and Edward A. Fox. 2012. Examining the Content of Online Resources Embedded in Tweets: Hurricane Isaac (8/23-9/26/2012). Cultivating Peace: A Student Research Symposium for Violence Prevention. Nov. 16-18, 2012. Blacksburg, VA
21. Sunshin Lee, Noha Elsherbiny, Edward A. Fox. A Digital Library for Water Main Break Identification and Visualization. Poster in Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 335-336, http://dx.doi.org/10.1145/2232817.2232878
22. Seungwon Yang, Kiran Chitturi, Gregory Wilson, Mohamed Magdy, and Edward A. Fox. A Study of Automation from Seed URL Generation to Focused Web Archive Development: The CTRnet Context. Poster in Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 341-342, http://dx.doi.org/10.1145/2232817.2232881
23. Scott Britell, Lois Delcambre, Lillian Cassel, Edward Fox, Richard Furuta. Exploiting canonical structures to transmit complex objects from a digital library to a portal. Poster in Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2012), Washington D.C., June 10-14, 2012, 377-378, http://dx.doi.org/10.1145/2232817.2232899
24. J. Leidig, S. Lee, E. Fox, "Teaching Advanced Content Using Formal Definitions", poster in 4th Annual Conference on Higher Education Pedagogy, Feb. 8-10, 2012, Virginia Tech, Blacksburg, VA
25. Donald J. Shoemaker, Edward A. Fox, and Steven D. Sheetz. CTRnet Digital Library for Disaster Information Services. Invited poster for Research Forum on Peace and Violence. Oct. 12, 2011, Virginia Tech
26. Monika Akbar, Weiguo Fan, Lillian Cassel, Lois Delcambre, Clifford A. Shaffer, Edward A. Fox, Yinlin Chen, "How Educators Find Educational Resources Online", ITiCSE 2011, June 27-29, Germany, 367

27. Prat Tanapaisankit, Min Song, and Edward A. Fox. Developing a Concept Extraction Technique with Ensemble Pathway. Refereed poster in Proceedings of JCDL 2011, Ottawa, June 13-17, 405-406
28. Jonathan Leidig, Edward A. Fox, Kevin Hall, Madhav Marathe, and Henning Mortveit. Improving Simulation Management Systems with Model Ontologies. Refereed poster in Proceedings of JCDL 2011, Ottawa, June 13-17, 435-436, http://dx.doi.org/10.1145/1998076.1998172
29. Seungwon Yang, Andrea Kavanaugh, Nadia P. Kozievitch, Lin Tzy Li, Venkat Srinivasan, Steven D. Sheetz, Travis Whalen, Donald Shoemaker, Ricardo da A. Torres, and Edward A. Fox. CTRnet DL for Disaster Information Services. Refereed poster in Proceedings of JCDL 2011, Ottawa, June 13-17, 437-438
30. Li, L.T., Yang, S., Kavanaugh, A., Fox, E., Sheetz, S., Shoemaker, D. Whalen, T., Srinivasan, V. Twitter Use During an Emergency Event: the Case of UT Austin Shooting. Refereed poster at ACM 2011 Digital Government Research Conference (dg.o 2011), June 12-15, 2011 (College Park, MD). New York: ACM Press, 335-336
31. Sheetz, Steve, Fox, Edward A., Fitzgerald, A., Palmer, S., Shoemaker, D., Kavanaugh, Andrea. Why Students Use Social Networking Sites After Crisis Situations. Refereed poster in Proceedings of the 8th International ISCRAM Conference, Lisbon, Portugal, May 8-11, 2011
32. Yinlin Chen and Edward A. Fox. Aggregate Computing Educational Resources in Social Interactive Environments. Refereed poster for Graduate Student Assembly's (GSA) 27th Annual Research Symposium, March 23, 2011, Virginia Tech
33. Monika Akbar, Peter Brusilovsky, Lillian Cassel, Steve Carpenter, Yinlin Chen, Lois Delcambre, Steve Edwards, Weiguo Fan, Edward Fox, Richard Furuta, Dan Garcia, Greg Hislop, Maggie Johnson, David Maier, Manuel Perez-Quinones, Steve Seidman, Frank Shipman, Chris Stephenson, Heikki Topi, Bryant York, "ENSEMBLE: Supporting the full range of computing education communities", NSDL Annual meeting 2010, Nov. 1-3, Washington, D.C.
34. Uma Murthy, Nadia P. Kozievitch, Edward A. Fox, Ricardo Torres, and Eric Hallerman. SuperIDR: A Tablet PC Tool for Image Description and Retrieval. Refereed poster for Workshop on the Impact of Pen-Based Technology on Education, WIPTE 2010, Oct. 25-26, Virginia Tech
35. Yinlin Chen and Edward A. Fox. IBM Cloud and Student Term Projects to Aid Computing Education. Poster for 3rd Annual Conference on Higher Education Pedagogy, Virginia Tech, October 7, 2010
36. Spencer J. Lee, Kriti Sen Sharma, Edward A. Fox, Matthew Ruder, and Ge Wang. Micro-CT Scanner Training in a 3D Virtual World: Second Life Aided Training and Education (SLATE). Refereed poster at BMES 2010 Annual Meeting, Oct. 6-9, 2010, Austin, TX
37. Frank M. Shipman, Lillian Cassel, Edward Fox, Richard Furuta, Lois Delcambre, Peter Brusilovsky, B. Stephen Carpenter II, Gregory Hislop, Stephen Edwards and Daniel D. Garcia. Ensemble: A Distributed Portal for the Distributed Community of Computing Education. Refereed poster in Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 506-509
38. Seungwon Yang, Tarek Kanan and Edward Fox. Digital Library Educational Module Development Strategies and Sustainable Enhancement by the Community. Refereed poster in Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 514-517
39. Nadia P. Kozievitch, Ricardo da Silva Torres, Felipe Andrade, Uma Murthy, Edward Fox, and Eric Hallerman. A Teaching Tool for Parasitology: Enhancing Learning with Annotation and Image Retrieval. Refereed poster in Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 466-469
40. G. Athanasopoulos, E. Fox, Y. Ioannidis, G. Kakaletris, N. Manola, C. Meghini, A. Rauber, and D. Soergel. A Functionality Perspective on Digital Library Interoperability. Refereed poster in Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 405-408
41. Lillian N. Cassel, Edward A. Fox, Richard Furuta, and Lois M.L. Delcambre. Many-to-Many Information Connections in a Distributed Digital Library Portal, Poster, JCDL 2010, June 21-25, Gold Coast, Australia, 369-370
42. Lillian (Boots) Cassel, Ed Fox, Frank Shipman, Peter Brusilovsky, Weiguo Fan, Dan Garcia, Greg Hislop, Richard Furuta, Lois Delcambre, Sridhara Potluri. Ensemble: enriching communities and collections to support education in computing: poster session. June 2010. Consortium for Computing Sciences in Colleges. Journal of Computing Sciences in Colleges, 25(6): 224-226

43. Nathan Short, Lynn Abbott, Supratik Misra, Michael Hsiao, Nadia Kozievitch, Sung Hee Park, Edward Fox. Latent Fingerprint Matching. Poster for CESCA (Center for Embedded Systems for Critical Applications) Day, Virginia Tech, Blacksburg, VA, May 6, 2010
44. Supratik Misra, Nathan Short, Michael Hsiao, Lynn Abbott, Edward Fox, Sung Hee Park, Nadia Kozievitch. Fingerprint Sufficiency. Poster for CESCA (Center for Embedded Systems for Critical Applications) Day, Virginia Tech, Blacksburg, VA, May 6, 2010
45. Sung Hee Park, Nadia Kozievitch, Edward A. Fox, Michael Hsiao, Lynn Abott, Nathan Short, Supratik Misra. Model-based fingerprint image quality Analysis. Poster for CESCA (Center for Embedded Systems for Critical Applications) Day, Virginia Tech, Blacksburg, VA, May 6, 2010
46. Nadia Kozievitch, Sung Hee Park, Supratik Misra, Nathan Short, Michael Hsiao, Lynn Abott, Edward A.Fox. Database for Fingerprint Experiments. Poster for CESCA (Center for Embedded Systems for Critical Applications) Day, Virginia Tech, Blacksburg, VA, May 6, 2010
47. Leidig J, Marathe M, Fox E. (2010) Scientific data management through digital libraries. Refereed poster presented at the NIH Models of Infectious Disease Agent Study (MIDAS) Network Meeting, Washington, DC, May 5-6, 2010.
48. Nadia P. Kozievitch, Ricardo da Silva Torres, Thiago Falcao, Evandro Ramos, Felipe Andrade, Silmara Marques Allegretti, Marlene Tiduko Ueta, Rubens Riscala Madi, Uma Murthy, Edward A. Fox, Yinlin Chen, and Eric Hallerman. A Geographic Annotation Service in SuperIDR. Poster at 2010 GIS and Remote Sensing Research Symposium, April 9, 2010, Virginia Tech
49. Monika Akbar, Edward A. Fox, Lillian N. Cassel, Yinlin Chen, Stephen Edwards, Peter L. Brusilovsky, Stephen Carpenter, Lois Delcambre, Weiguo Fan, Richard Furuta, Frank M. Shipman, Daniel Garcia, Greg Hislop, David Maier, Manuel Perez-Quinones, Christine Stephenson, Bryant York, Haowei Hsieh. ENSEMBLE: Supporting the Full Range of Computing Education Communities. Reviewed poster for ACM SIGCSE 2010, March 12, Milwaukee
50. Seungwon Yang, Edward A. Fox, John Ewers, and Tarek Kanan. A Methodology for Digital Library Educational Module Development and Continuous Improvements by the Community. Reviewed poster at 2010 Conference on Higher Education Pedagogy, Inn at Virginia Tech, Feb. 18-19, 2010, p. 111, http://www.cider.vt.edu/conference/2010/2010proceedings.pdf
51. Edward A. Fox, Monika Akbar, Yinlin Chen, Michael Stewart, Clifford A. Shaffer, Stephen H. Edwards, Patrick Fan. Ensemble: Enriching Communities and Collections to Support Education in Computing. Reviewed poster at 2010 Conference on Higher Education Pedagogy, Inn at Virginia Tech, Feb. 18-19, 2010, p. 102, http://www.cider.vt.edu/conference/2010/2010proceedings.pdf
52. Sanghee Oh, Barbara Wildemuth, Pomerantz Jeffrey, Seungwon Yang, Edward Fox. Using a Wiki as a Platform for Formative Evaluation. Poster at ASIS&T 2009 Annual Meeting, Thriving on Diversity - Information Opportunities in a Pluralistic World, November 6-11, 2009, Vancouver, BC, Canada, p. 46
53. Venkat Srinivasan, Bidisha Dewanjee, Edward A. Fox, Donald J. Shoemaker, Steven D. Sheetz, Andrea Kavanaugh, and Naren Ramakrishnan. "CTRnet: A Distributed Digital Library for Rescue and Recovery". Poster presented at International Outreach NOW conference at Virginia Tech, Blacksburg, Virginia, Sep 2009, winner of best poster in category award.
54. Barbara M. Wildemuth, Jeffery Pomerantz, Sanghee Oh, Seungwon Yang, Edward A. Fox. The Variety of Ways in which Instructors Implement a Modular Digital Library Curriculum. Poster at JCDL 2009
55. Yang, S; Fox, E; Wildemuth, B; Wildemuth, B; Oh, S; Pomerantz, J; Digital Library Educational Module Development and Field-Testing Results from Virginia Tech. Refereed poster 65 presented at Conference on Higher Education Pedagogy @ VT, February 18, 2009, Virginia Tech, http://www.ceutonline.com/proposal/viewPoster1.cfm?pid=127
56. Chen, Y; Fox, E; Steve , E; Shaffer, C; The National Science Digital Library resources to enhance learning: lessons from CITIDEL and the Ensemble pathways project. Refereed poster 66 presented at Conference on Higher Education Pedagogy @ VT, February 18, 2009, Virginia Tech, http://www.ceutonline.com/proposal/viewPoster1.cfm?pid=128
57. Chung, W; Yang, S; Fox, E; Sheetz, S; The Living In the KnowlEdge Society (LIKES) Community Building Project: Building Collaboration between Computing and Non-Computing Disciplines. Refereed poster 67 presented at Conference on Higher Education Pedagogy @ VT, February 18, 2009, Virginia Tech, http://www.ceutonline.com/proposal/viewPoster1.cfm?pid=130

58. Seungwon Yang, Edward A. Fox, Wingyan Chung. Living In the KnowlEdge Society (LIKES) Initiative and I-Schools' Focus on the Information. Poster 64 at 2009 iConference, Feb. 8-11, 2009, U. NC Chapel Hill
59. Barbara M. Wildemuth, Jeffrey Pomerantz, Sanghee Oh, Seungwon Yang, Edward A. Fox. i-Schools as a Natural Home for Digital Libraries Education. Poster 67 at 2009 iConference, Feb. 8-11, 2009, U. NC Chapel Hill
60. Jeffrey Pomerantz, Barbara M. Wildemuth, Sanghee Oh, Seungwon Yang, Edward A. Fox: Evaluation of a curriculum for digital libraries. In Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries (Pittsburgh PA, PA, USA, June 16 - 20, 2008). JCDL '08. ACM, New York, NY, p. 462. DOI= http://doi.acm.org/10.1145/1378889.1379007
61. Lois Delcambre, David Archer, Susan Price, Uma Murthy, Edward Fox, Lillian Cassel. Superimposing a Strand Map over a Database Lecture. Poster at 39th ACM Technical Symposium on Computer Science Education, SIGSCE 2008, March 12-15, 2008, Portland, Oregon
62. Barbara L. Moreira, Marcos A. Goncalves, Alberto H. F. Laender, and Edward A. Fox, Evaluating Digital Libraries with 5SQual. Research and Advanced Technology for Digital Libraries, ECDL 2007: pp. 466-470, Springer Berlin / Heidelberg, LNCS 4675, poster, http://dx.doi.org/10.1007/978-3-540-74851-9_44
63. Christopher L. Barrett, Keith Bisset, Stephen Eubank, Edward A. Fox, Yi Ma, Madhav V. Marathe, and Xiaoyu Zhang: A Scalable Data Management Tool to Support Epidemiological Modeling of Large Urban Regions. Research and Advanced Technology for Digital Libraries, ECDL 2007: pp. 546-548, Springer Berlin / Heidelberg, LNCS 4675, poster, http://dx.doi.org/10.1007/978-3-540-74851-9_65
64. Yi Ma, Edward A. Fox, Marcos A. Goncalves PIM through a 5S perspective. ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, poster, p. 491, http://doi.acm.org/10.1145/1255175.1255290
65. Ryan Richardson, Ben Goertzel, Edward A. Fox, and Hugo Pinto. Automatic Creation and Translation of Concept Maps for Computer Science-Related Theses and Dissertations. Poster for Proc. CMC 2006, Costa Rica, Sept. 5-8, 2006.
66. Michael Drutar, Charles Coleman, and Edward Fox. Creating a Multi Disciplinary Digital Library in the 5S Framework. Poster for JCDL 2006, June 11-16, Chapel Hill, NC, p. 351, http://doi.acm.org/10.1145/1141753.1141846
67. Deborah Duong, Ben Goertzel, Jim Venuto, Ryan Richardson, Edward Fox. Support Vector Machines to Weight Voters in a Voting System of Entity Extractors. Poster for 2006 International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence (http://wcci2006.org), July 16-21, 2006, Vancouver, 4 pages
68. Michael Drutar, Charles Coleman, and Edward Fox, Creating a Multi Disciplinary Digital Library in the 5S Framework. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, poster, p. 351
69. R. da S. Torres, C. B. Medeiros, R. Q. Dividino, M. A. Figueiredo, M. A. Goncalves, E. A. Fox, R. Richardson: Using Digital Library Components for Biodiversity Systems. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, p. 408
70. Edward A. Fox, Ryan Richardson, and Lillian Cassel. CITIDEL: Computing and Information Technology Interactive Digital Educational Library. Poster, ECDL'2003, 17-22 August, Trondheim, Norway
71. Rao Shen, Ryan Richardson, Edward A. Fox. Concept maps as visual interfaces to digital libraries. Poster, ECDL'2003, 17-22 August, Trondheim, Norway
72. Filip Jagodzinski, Lillian Cassel, Marcos A. Goncalves, and Edward A. Fox. Digital Library Analysis: Advantages of Using the XML Log Standard Versus the Apache Server Logs. Poster for JCDL'2003, May 27-31, 2003, Houston, TX
73. Edward A. Fox, Reagan W. Moore, Ronald L. Larsen, Sung Hyon-Myaeng, and Kim Sung-Hyuk. US-Korea Collaboration on Digital Libraries: An Overview and Generalization for Pacific Rim Collaboration. Poster for 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Dec. 2002, p. 519.
74. Posters at NSDL Launch Meeting, Washington, D.C., Dec. 2-4, 2002: for NSDL PC (with Ellen Hoffman), CITIDEL (with Lillian Cassel), DLbox (with SuShing Chen), and GetSmart (with Hsinchun Chen and Ann Lally)

75. MARIAN - Next Generation Digital Library System Built on 5S, with M. Goncalves, W. Richardson, and M. Luo; Honorable Mention, with $100 prize
76. Pavel Calado (Universidade Federal de Minas Gerais, Brazil), Marcos Goncalves (Virginia Polytechnic Institute and State University), Berthier Ribeiro-Neto (Universidade Federal de Minas Gerais, Brazil), and Edward Fox (Virginia Polytechnic Institute and State University). Making Digital Libraries from the Web. 11th International Conference on Information and Knowledge Management (CIKM'02), Nov. 4-9, 2002, McLean, VA
77. Deborah Knox, Lillian Cassel, John Impagliazzo, Edward Fox, JAN Lee, Manuel Perez, and C. Lee Giles. "Computing and Information Technology Interactive Digital Education Library (CITIDEL)". Poster at ACM SIGCSE Symposium 2002, http://www.cs.wpi.edu/~cew/sigcse2002posters.html, in Proc. of the 33rd SIGCSE Technical Symposium on Computer Science Education, Feb. 27 - March 3, 2002, Covington, KY, p. 427
78. E. Fox, G. Abdulla, W. Heagy. Quantitative Analysis and Visualization Regarding Interactive Learning with a Digital Library in Computer Science. ACM Digital Libraries'97, Philadelphia, PA, July 23-26, 1997

UNREFEREED INVITED CONFERENCE/WORKSHOP PAPERS:

1. Functionality and Interoperability with 5S. Presentation for Functionality Working Group Meeting, DL.Org (Digital Library Interoperability, Best Practices and Modeling Foundations), 29-30 June 2009, Athens, http://fox.cs.vt.edu/talks/2009/20090629FoxDLorgFunWG.ppt
2. Edward A. Fox. Introduction to NDLTD and Brief History of the ETD Movement. Invited presentation for Newcomers' Workshop, ETD 2009, Pittsburgh, PA, June 10, 2009, http://fox.cs.vt.edu/talks/2009/20090610ETD2009RookiesFoxIntro.ppt
3. Edward A. Fox and Gail McMillan. ETD-db: Workflow, the Short Story. Invited presentation for Newcomers' Workshop, ETD 2009, Pittsburgh, PA, June 10, 2009, http://fox.cs.vt.edu/talks/2009/20090610ETD2009RookiesETDdb.ppt
4. Edward A. Fox. It is easy for universities to support free culture with digital libraries: The NDLTD Example. Invited paper for Symposium on Free Culture & the Digital Library 2005, Emory University, Atlanta, 14 Oct., http://metascholar.org/events/2005/freeculture/, http://fox.cs.vt.edu/talks/2005/20051014Emory.ppt
5. Edward A. Fox. Toward a Science Extending Digital Libraries, with a Testbed of ETDs. Position paper for 15-17 June 2003 NSF-sponsored Workshop on Post-DL Research Directions, Chatham, MA, 6 pages
6. H. Suleman, E. Fox, and D. Madalli. Design and Implementation of Networked Digital Libraries: Best Practices. In Proc. DRTC Workshop on Digital Libraries: Theory and Practice, March 10-14, 2003, Indian Statistical Institute, Bangalore, India, 9 pages
7. Edward Fox, Hussein Suleman, and Ming Luo. Building Digital Libraries Made Easy: Toward Open Digital Libraries, pages 14-24 in: Ee-Peng Lim, Schubert Foo, Chris Khoo, Hsinchun Chen, Edward Fox, Shalini Urs, Thanos Constantino, eds. Digital Libraries: People, Knowledge, and Technology; Proceedings 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Dec. 2002. Springer, Lecture Notes in Computer Science 2555.
8. The OAI PMH (Open Archives Initiative Protocol for Metadata Harvesting). Invited presentation for MetaScholar Initiative All-Project Meeting, Atlanta, GA, June 18, 2002.
9. W. Xi and E. A. Fox. Machine Learning Approach for Homepage Finding Task. In Proceedings of the Tenth Text Retrieval Conference (TREC-2001). NIST Special Publication 500-250, entry 78, 2001. http://trec.nist.gov/pubs/trec10/t10_proceedings.html
10. E. Fox. "Future Funding and the NDLTD Vision." Invited closing plenary for Third International Symposium on Electronic Theses and Dissertations (ETDs), St. Petersburg, FL, March 18, 2000 http://www.ndltd.org/talks/ETD3close.ppt
11. E. Fox. "Future of Electronic Publishing." Invited seminar in Virginia Tech Symposium on Scholarship in the Electronic World, text and audio online, paper in published proceedings, Research and Graduate Studies, Blacksburg, VA, April 13, 1998

http://www.rgs.vt.edu/resmag/seminars.html
12. E. Fox. "Update on the Networked Digital Library of Theses and Dissertations (NDLTD)." In Proc. 35th Annual Clinic on Library Applications of Data Processing, GSLIS 98, DPC'98, U. Illinois Urbana-Champaign, March 22-24, 1998, ISSN 0069-4789
13. E. Fox. "Effects on Education, and a Proposal for Collection of Tools" Invitational Workshop on Information Retrieval Tools http://www.sis.pitt.edu/~erasmus/workshop.html, U. Pittsburgh, 3/19-21/98, http://fox.cs.vt.edu/talks/IRtools.htm
14. E. Fox. "A Scalable Digital Ecology for a Networked Digital Library of Theses and Dissertations (NDLTD)." Winter Workshop of the Human-Computer Interaction Consortium (HCIC), Fraser, Colorado, March 4-8, 1998
15. E. Fox. Networked Digital Library of Theses and Dissertations (NDLTD): A Worldwide Initiative to Advance Scholarship, invited presentation for session on Progress on Digital Dissertation Initiatives, Chair: Joan K. Lippincott, for Coalition for Networked Information Fall Meeting, Minneapolis, MN, Oct. 25-27, 1997 http://www.ndltd.org/talks/CNIF97.pdf
16. E. Fox. Digital Libraries and Virtual Universities, invited presentation for "Information research for designing and planning virtual universities" Seminar at Centro Universitario de Investigaciones Bibliotecolgicas Universidad Nacional Autnoma de Mexico Cd. Universitaria, Mexico, D.F. (Library and Information Research Center, National University of Mexico), Aug. 11-15, 1997, http://fox.cs.vt.edu/talks/Mexico97.html
17. J. Eaton, E. Fox, G. McMillan. Electronic Theses and Dissertations (ETDs) and Their Contribution to Graduate Education. Proc. 53rd Annual Meeting Midwestern Association of Graduate Schools, MAGS, 1997, 73-78
18. E.A. Fox. Interactive Learning with a Digital Library in Computer Science. Invited paper for Proc. Frontiers in Education - FIE'96, Salt Lake City, Utah, Nov. 6-9, 1996, 415-419.
19. E. Fox. Digital Libraries, WWW, and Educational Technology: Lessons Learned. Invited paper for Proc. ED-MEDIA 96, World Conference on Educational Multimedia and Hypermedia, Boston, MA, June 17-22, 1996, 246-251. See also http://ei.cs.vt.edu/~fox/EDMEDIA96/
20. E. Fox. Digital Library Support for Education - A Case Study of Advanced Networked Information Systems, Invited presentation for Dagstuhl Workshop on Networked Information Systems - Discovery, Retrieval, Dissemination, Feb. 26 - March 1, 1996, Dagstuhl, Germany.
21. J.M. Carroll, M.B. Rosson, E.A. Fox, A.M. Cohill, K.W. Schmidt, and N.A.Kipp. Community Networks as Working Memory, Invited presentation for HCIC (HCIC Consortium), Feb. 1996.
22. E. Fox, N. Barnette, C. Shaffer, L. Heath, W. Wake, L. Nowell, J. Lee, D. Hix, and H. Hartson. Progress in Interactive Learning with a Digital Library in Computer Science. Invited paper for Proc. ED-MEDIA 95, World Conference on Educational Multimedia and Hypermedia, Graz, Austria, June 17-21, 1995, pp. 7-12.
23. J. Shaw and E. Fox. Combination of Multiple Searches. In The Third Text REtrieval Conference (TREC-3), National Institute of Standards and Technology Special Publication, 500-225, April 1995, ed. D. Harman.
24. E. Fox. Seamless Multimedia Integration for Digital Libraries. Invited position paper for Dagstuhl Seminar on Fundamentals and Perspectives of Multimedia Systems, International Conf. and Research Center for Computer Science, Dagstuhl Castle, Germany, July 4-8, 1994, 118-123.
25. E. Fox. A digital library connecting Envision, KMS, and Mosaic with interfaces, communications, and data interchange. Invited presentation for 1994 Workshop on Digital Libraries: Current Issues, sponsored by: Rutgers and Purdue Universities, AT&T and Bellcore, at Rutgers Univ., Newark, NJ, May 19-20, 1994. Abstract in SIGOIS Bulletin, Aug. 1994, 15(1): 6.
26. E. Fox and O. Balci. MDDS Design Evaluation Environment. Massive Digital Data Systems Workshop, Feb. 1-2, 1994, Reston, VA, sponsored by Director of Central Intelligence, Community Management Staff, Advanced Technology Office, 1 page abstract, reviewed.
27. E. Fox and J. Shaw. Combination of Multiple Searches. In Proc. of The Second Text REtrieval Conference (TREC-2) (Aug. 30 - Sept. 1, 1993, NIST, Gaithersburg, MD), NIST Special Publication 500-215, 1994, ed. D. K. Harman, 243-252
28. E. Fox. A User-Centered Hypermedia Database from the Computer Science Literature. Invited presentation for Advances in Data Management for the Scientist and Engineer Session 2, AAAS '93, Feb. 11-16, 1993, Boston. Abstract in Proc. 159th National Meeting of the AAAS, AAAS'93: Science and Engineering for the Future, p. 145. Longer paper: E. Fox, L. Heath, and D. Hix, A User-Centered Database from the

Computer Science Literature, in Proceedings of the NSF Scientific Database Projects, 1991-1993, eds. W. Chu, A. Cardenas, and R. Taira, Feb. 14-16, 1993, Boston, 70-75.

29. E. Fox, M. Koushik, J. Shaw, R. Modlin, and D. Rao. Combining Evidence from Multiple Searches. In Proc. of The First Text REtrieval Conference (TREC-1) (Nov. 4-6, 1992, Rockville, MD), ed. D. K. Harman, NIST Special Publication 500-207, Washington: U.S. Government Printing Office, 1993, pp. 319-328.
30. E. Fox. An Information Retrieval Viewpoint of Query Processing for VIMS. Invited position paper for NSF Workshop on Visual Information Management Systems, organized by U. of Michigan and IBM Almaden Research Center, Feb. 24-24, 1992, San Francisco, CA.
31. E. Fox. Multimedia Update. Invited presentation appearing in ONLINE/ CD-ROM'91 Conference Proceedings, Nov. 11-13, 1991, San Francisco, CA, 65-69. In Support of Democratic Information Access. Invited presentation for "Emerging Technologies" panel at Future Directions in Text Analysis, Retrieval and Understanding workshop, sponsored by NSF, 11-12 Oct. 1991, Chicago IL, 7 pages.
32. E. Fox. Background on Information Retrieval Research. Invited presentation for AAAI-91 Workshop on Natural Language Text Retrieval, July 15, 1991, Anaheim, CA, 4 pages.
33. E. Fox, A. Daoud. CD-ROM Hardware: What's Real, What's Next. Invited presentation appearing in ONLINE/ CD-ROM '90 Conference Proceedings, Nov. 5-7, 1990, Washington Hilton and Towers, Washington, D.C., 63-67.
34. E. Fox. An Electronic Publishing / Information Storage and Retrieval Perspective on the Management of Scientific Databases. Invited presentation for National Science Foundation sponsored Workshop on Scientific Databases, March 12-13, 1990, Charlottesville, VA, 2 pages.
35. E. Fox, Qi Fan Chen. Why and How We Should Have Hypertext/Hypermedia Standards: ACM's Need and the CODER/LEND Approach. Selected for workshop discussion at Hypertext Standardization Workshop, National Institute of Standards and Technology, Jan. 16-18, 1990, Gaithersburg, MD, 10 pages.
36. E. Fox, A. Daoud. CD-ROM Hardware: An Update. Invited presentation for Online '89, November 7-9, 1989, Chicago, IL, 64-69.
37. E. Fox. Advances in Text and Image Publishing for CD-ROM. Invited presentation for National Online Meeting, May 10, 1989, New York, 165-170.
38. E. Fox. Building the CODER System: Working Toward a Comprehensive Testbed for AI in IR. Workshop on Distributed Expert-Based Information Systems, March 5-7, 1987, Rutgers Univ., New Brunswick, NJ, 9 pgs.
39. E. Fox. Advanced Information Retrieval Methods. Workshop on "Strategies for Information Access" at Second International Conference on CD ROM. Making It Happen, March 2-5, 1987, Seattle, WA, 12 pages.
40. E. Fox. A Design for Intelligent Retrieval: The CODER System. Proc. of the Second Conf. on Computer Interfaces and Intermediaries for Information Retrieval, May 28-31, 1986, Boston MA, Report DTIC / TR-86 / 5, 135-154.

HOSTING VISITS (selected):

- Hosting 1 year visit from Denilson Alves Pereira (Ph.D. 2009), professor in Dept. of CS, Universidade Federal de Lavras (UFLA), Feb. 2016 - Jan. 2017
- Hosting 1 year visit from Kuwait University, by Associate Professor Sultan Al-Daihani, starting Sept. 2014
- Hosting 1 year visit from UNICAMP, Brazil, by Ph.D. student Lin Tzy Li, August 2010 - July 2011
- Hosting 8 month visit from UNICAMP, Brazil, by Ph.D. student Nadia P. Kozievitch, Feb. 2010 - Sept. 2010
- Hosting visit by Ricardo daSilva Torres from UNICAMP, Brazil, July 31 - Aug. 4, 2008
- Hosting visit from Darmstadt U. of Technology by Prof. Dr. Rudi Schmiede, May 27-30, 2008
- Hosting 3 Fulbright Scholars visiting from India starting Aug. 2002: Buddhi Chauhan, Pradeep Cheemalavagupalli, Devika Madalli
- Hosting Sabbatical Visit by Lillian Cassel (Villanova U.), all of fall semester 2002 and part-time spring semester 2003

- Hosting Professor Byongsun (Sunny) Kim, Korean Studies Information Center, The Academy of Korean Studies, Seoul, visiting Dec. 14, 2000 - June 10, 2002
- Mann-Ho Lee, Chungnam National University (formerly here on sabattical), discussing with the Digital Library Research Laboratory MARIAN group and CITIDEL project team, Feb. 15, 2002. (He was part of delegation discussing a cooperative exchange program between Virginia Tech and Chungnam National University, with an MOU signed by respective university presidents.)
- Lillian Cassel and Deborah Knox, presenting to the CITIDEL project team, Feb. 15, 2002
- Russ Davis and Dennis Barnes, eNumerate, Inc., presenting to the Digital Library Research Laboratory, Feb. 6, 2002
- AOL team, Jan. 15, 2002
- Ann Lally, University of Arizona, co-PI on NSF grant, Dec. 5-6, 2001, giving Virginia Tech CS seminar on Dec. 5
- John Impagliazzo, Hofstra U., Nov. 29 - Dec. 1, 2001
- Bill Winter, of Internet TIC, Nov. 5, 2001
- UCL (Belgium) visit in conjunction with VTLS, July 23, 2002
- Professor Eberhard Hilf, April 11-13, 2001
- Professor Shalini R. Urs, Dept. Library and Information Science, U. Mysore (India), Fulbright Fellowship, 5 months: August 12, 2000 - Feb. 12, 2001
- Professor Mann-Ho Lee, Dept. of Computer Science, CNU (Chungnam National University) in Korea, visiting for 1 year starting Aug. 14, 2000
- Professor Pablo de la Fuente, Departamento de Informatica, University of Valladolid (Spain), visiting 4/27/2000 - 6/24/2000
- Dr. Ahira Maeda, Japan, postdoctoral fellow for Oct. 22, 2000 - March 21, 2001, after graduation from Nara Institute of Science and Technology (NAIST), Japan
- Professor Byongsun (Sunny) Kim, Korean Studies Information Center, The Academy of Korean Studies, Seoul, visiting for 1 year starting Dec. 14, 2000
- RAI-Channel Two News Group from Italy, hosted visit to Digital Library Research Laboratory and other parts of Torgersen Hall, Oct. 10, 2000
- ABD Fellow Denise Haskings, from The George Washington, Summer 1997
- Marian Bate, Univ. New South Wales, 7/18/97
- Jong-Min Bae, from Korea, Aug. 1997 - June 1998 (sabattical)
- WVU team visit 11/6/97
- W&M team visit 11/18-19/97
- ISOGEN team 2/26/98
- Sugimoto Shigeo (Japan) 3/18-20/98 and 12/1-2/99
- Delegation from Korea National Open U., Aug. 21, 1998
- Dr. Peter Schirmbacher, Humboldt U., Berlin, Feb. 1999
- Eugene Evans, Infinite Ventures, Inc., Sep. 2, 1999 (speaker on "Alternative Teaching/Learning Strategies" - brown-bag lunch)
- Susanne Dobratz, Humboldt U., Berlin, Sep. 16, 1999 (speaker on "Dissertations Online" - for brown-bag lunch)
- Professor Shalini R. Urs, Dept. Library and Information Science, U. Mysore, India, Fulbright Fellowship, 2000-2001 (scheduled)
- Professor Mann-Ho Lee, Dept. of Computer Science, CNU (Chungnam National University) in Korea, visiting for year starting fall 2000
- Professor Pablo de la Fuente, Departamento de Informatica, University of Valladolid (Spain), visiting 2 months summer 2000

AWARDS SERVING AS (CO-)PRINCIPAL INVESTIGATOR:

1. Mayfair Group LLC: Amazon Mechanical Turk Service for Thesis Research, $818, 2/24 - 8/31/2020, PI Edward A. Fox.

2. Mayfair Group LLC: Contribution for fellowship in 2020, $11,034.25, PI Edward A. Fox.
3. IMLS LG-37-19-0078-19: Opening Books and the National Corpus of Graduate Research. $505,214 for 8/1/2019 - 7/31/2022. PI: William A. Ingram (VT Library); Co-PIs: Edward A. Fox (VT CS), and Jian Wu (ODU CS).
4. Amazon Web Services (AWS): Research credits. $18,384 for 7/29/2019 - 9/30/2020. PI Edward A. Fox
5. NSF IIP-1924726: I-Corps: Automated Summarization Technology. $50,000 for 5/1/2019 - 6/30/2020, PI Edward A. Fox, Co-PI Naren Ramakrishnan
6. NSF flow through from economic development grant to John Provo: Legal document analysis and summarization pilot, PI Edward A. Fox and Co-PI Naren Ramakrishnan, $24,796 for 4/1/2019 - 7/31/2019.
7. NewWave: Contribution for fellowship 2018-2019, $77,419.53, PI Edward A. Fox.
8. Mayfair Group LLC: Contribution for fellowship in 2019, $19K, PI Edward A. Fox.
9. NSF OAC-1835660: Collaborative Research: Framework: Software: CINES: A Scalable Cyberinfrastructure for Sustained Innovation in Network Engineering and Science, $2,880,000 to VT for 11/1/2018-10/31/2023, PI Madhav Marathe; Co-PIs:Catherine Amelink, Edward Fox, Naren Ramakrishnan, Christopher Kuhlman
10. IMLS RE-70-18-0005-18: Continuing Education to Advance Web Archiving, $248,451 for 5/1/2018 - 4/30/2020, PI Zhiwu Xie, VT Co-PI Edward A. Fox, with key project staff at multiple other institutions
11. Mitre Corporation: Computer Science Capstone Project, $5000 for 1/25/2018 - 5/11/2018, PI Edward A. Fox
12. Mayfair Group: Deposition Summarization, $255,615 for 1/1/2018 - 8/15/2018, PIs Naren Ramakrishnan and Edward Fox
13. NSF CMMI-1638207, CRISP Type 2/Collaborative Research: Coordinated, Behaviorally-Aware Recovery for Transportation and Power Disruptions (CBAR-tpd), $876,913 for 1/1/2017-12/31/2020, PI Pamela Murray-Tuite, Co-PIs Edward Fox, Kris Wernstedt; in collaboration with grant 1638197 for $249,921 to U. Mich. Ann Arbor, PI Seth Guikema
14. NSF IIS-1619028, III: Small: Collaborative Research: Global Event and Trend Archive Research (GETAR), $446K for 1/1/2017-12/31/2019, PI Fox, Co-PIs Alla Rozovskaya (replaced by Chandan Reddy), Andrea L. Kavanaugh, Donald J. Shoemaker; in collaboration with grant 1619371 for $54K to Internet Archive, PI Jefferson Bailey.
15. IMLS LG-71-16-0037-16: Developing Library Cyberinfrastructure Strategy for Big Data Sharing and Reuse; $308,175 for 2 years starting June 1, 2016, extended to 5/31/2019; Zhiwu Xie (PI), Tyler Walters, Edward Fox (20%), Pablo Tarazaga
16. University of North Texas (NSF flow through supplement request): CREST Partnership Supplement: Building Capacity in Information Management through a Partnership with Virginia Tech's Digital Library Technology Center, sole PI (with main grant to UTEP), $36,659 to VT for 9/1/2015 - 5/31/2017. Funded by NSF to UTEP by 2/11/2016
17. VT ARC. VT-Rnet: A 10-Gbps Research Network for Virginia Tech. In-kind support to connect the Digital Library Research Laboratory Hadoop Cluster to VT's 10 gigabits per second network. Dec. 2015. PI Edward A. Fox.
18. NEH Grant EH-231264-15, Veterans in Society Summer Institute for College Teachers, 10/1/2015 - 12/31/2016, PI James M. Dubinsky, co-PI Bruce E. Pencek, $152,250, CS funding $1054
19. NIH Grant 1R01DA039456-01: The Social Interactome of Recovery: Social Media as Therapy Development; $1,674,440 for 9/15/2014-8/31/2017, PI Warren K. Bickel (VTCRI), Fox as co-PI (16% personal share)
20. Columbia U. (from Mellon Foundation): Archiving Transactions Towards Uninterruptible Web Service. $25,000 for 4/1/2014-9/30/2015. PI Zhiwu Xie, Co-PI Edward A. Fox (25% personal share), proposal 14-1578-01
21. Virginia Tech Institute for Creativity, Arts, and Technology, Science, Engineering, Arts, and Design: The ecology of 3D objects (3Dcology): Exploring and understanding creation, use, and flow of 3D objects. $3000, for 8/26/2013 - April 30, 2014; sole PI
22. NSF IIS - 1319578: III: Small: Integrated Digital Event Archiving and Library (IDEAL); $500,000 for 9/1/2013 - 8/31/2016, PI, with co-PIs Donald Shoemaker, Andrea Kavanaugh, Steven Sheetz, and Kristine Hanna (Internet Archive)

23. Villanova University (pass-through from NSF DUE-1141209, Account 416931): Computing in Context; $22,500 for 8/15/2012 - 7/31/2015; sole PI (with overall PI at Villanova: Robert Beck, $199,244 total)
24. NSF DUE-0840719 supplement to: Collaborative Research: Ensemble: Enriching Communities and Collections to Support Education in Computing; additional $84,895 to VT for 2 years starting 9/15/2012; VT PI Fox (50%), co-PIs Stephen H. Edward and Weiguo Fan
25. Qatar National Research Fund Project No. NPRP 4-029-1-007, Account 457837: Establishing a Qatari Arabic-English Library Institute, Lead PI Edward A. Fox, Co-Lead PI Mohammed Samaka (Qatar University), Co-PIs C. Lee Giles (Pennsylvania State University), John Impagliazzo (Hofstra), Richard Furuta (Texas A&M University), $1,030,484.98 total ($125,277 for Virginia Tech). 4/1/2012 - 12/31/2015, funds in place at VT by Jan. 2013.
26. VTX, Part of GlobalWeb Corp., Sao Paulo, Brazil: Organizing Crawled Web Data for Brazilian Customers, PI Edward A. Fox, investigators N. Ramakrishnan and H.R. Hartson, $199,920 for 12/5/2011 - 5/15/2012.
27. NSF OCI-1032677, SDCI NMI New: From Desktops to Clouds - A Middleware for Next Generation Network Science, PI Madhav Marathe (17%), Co-PIs Keith Bisset (17%), Anil Vullikanti (17%), Henning Mortveit (17%), E. Fox (15%), Annette Feng (17%), $1,350,000 for 8/1/2010-8/31/15
28. CCSR (partnership between Arlington Co., IBM Research, VT): Social Media for Cities, Counties, and Communities, $15,000 (half to IBM and half to VT) for 6/21/2010 - 12/31/2010, VT PI: Andrea Kavanaugh, co-PIs: Edward Fox, Donald Shoemaker, Steven Sheetz; IBM PI: Apostol Natsev, co-PI: Lexing Xie
29. BAE Systems (with funds from DHS) Proposal 75000-02-13, Account 416289: Biometrics Training and Performance Initiative (BTPI), PI Randall S. Murch, co-PIs: A. Lynn Abbott, Edward Fox, Michael Hsiao, Kathleen A. Meehan, Eric Smith; initial letter award made Dec. 2009, rest confirmed March 2010: $484,622 (CS share with Fox sole CS co-PI: $51,039.88), 11/3/2009-3/30/2013
30. VT Advance: Department Climate Mini-Grant Program, $2500 for Computer Science mentoring, 11/16/2009-6/30/2010, PI Edward A. Fox
31. National Institute of Justice NCJ 239049: Establishing the Quantitative Basis for Sufficiency: Thresholds and Metrics for Friction Ridge Pattern Detail Quality and the Foundation for a Standard, $854,907 for 10/1/2009-9/30/2011 followed by 6 month extension to 3/31/2012, PI Randall Murch, Co-PIs A. Lynn Abbott, Michael Hsiao, Bruce Budowle, and Edward A. Fox
32. NSF IIS-0916733: III:Small:Integrated Digital Library Support for Crisis, Tragedy, and Recovery, $500,000, 8/1/2009 - 7/31/2013, PI Edward A. Fox. Co-PIs: Naren Ramakrishnan, Steven Sheetz, Andrea Kavanaugh, Donald Shoemaker
33. NSF IIS-0910183: Collaborative Research: Curatorial Work and Learning in Virtual Environments, $75,000 + $8,000 REU Supplement (followed by 6 month extension), 4/1/09-3/31/10, PI at VT, with overall PI Gary Marchionini at UNC-CH
34. VT Center for Excellence in Undergraduate Training (CEUT): Living In the KnowlEdge Society. Instructional grant: $2100. PI: Edward A. Fox. Co-PIs: Weiguo Fan, Christopher Zobel, Carlos Evia, Steven Sheetz. November 14, 2008 (1 year)
35. NSF DUE-0840719: Collaborative Research: Ensemble: Enriching Communities and Collections to Support Education in Computing; $425,002 total to VT, with overall total to all partners $2M; VT PI Fox (50%), co-PIs Stephen H. Edward and Weiguo Fan. Plus $6,293 provided summer 2012 from the Villanova funds though a subcontract to support GRA Y. Chen.
36. IBM: Extraction of Named Entities from Documents for Business Purposes; $25,450; 2/2008 - 6/2008; GRA Yi Ma; sole PI
37. Google: Concept Maps for Discovering/Using Multilingual Electronic Theses and Dissertations, $50,000, PI Edward A. Fox, award Sept. 2007
38. NSF IIS-0736055: SGER: DL-VT416: A Digital Library Testbed for Research Related to 4/16/2007 at Virginia Tech, $199,993+REU Supplement, PI Edward A. Fox, Co-PIs: Christopher L. North, Donald J. Shoemaker, Naren Ramakrishnan, Weiguo Fan, August 15, 2007 - July 31, 2008
39. NSF CCF-0722259: Collaborative Research: CPATH CB: Living In the KnowlEdge Society (LIKES), $289,999 + $8,000 REU Supplement to VT, PI Edward A. Fox, Co-PIs: Carlos Evia, Christopher W. Zobel, Steven D. Sheetz, Weiguo Fan, in collaboration with 0722289 to co-PI Edward Carr, North Carolina A&T State University; 0722276 to Robert E. Beck, Villanova University; and 0722263 to Wingyan Chung, Santa Clara University (originally to University of Texas at El Paso), August 1, 2007 - July 31, 2010

40. Microsoft Research: Deployment and Assessment of an Image Annotation and Retrieval Tool, Including for Biodiversity, $50,000, 5/16/2007 - 5/15/2008, PI: Edward A. Fox, Co-PIs: Eric Hallerman, Ricardo da Silva Torres.
41. Idaho National Laboratory: Storytelling in the National Nuclear Archive, $10,866 in hand out of $100K expected, 4/1/2007-3/31/2008, PI: Naren Ramakrishnan, Co-PI: Edward Fox (50-50).
42. NSF OCI-0636151: CI-TEAM Implementation Project: Collaborative Research - A National Engineering Dissection Cyber-Collaboratory, $104,378.00, 1/1/2007 - 12/31/2008, PI Janis Terpenny, co-PIs: Lisa McNair, Richard Goff, Mary Kasarda, Edward Fox.
43. IBM Unstructured Information Management Architecture (UIMA) Faculty Innovation Award: Incorporating Superimposed Information into UIMA, and UIMA into Digital Library Curriculum Development, as a Prototype Tool for Resource Collection Building and Automatic Syllabus Generation in the Semantic Web; $16,740; 7/1/2006-6/30/2007; sole PI
44. NSF IIS-0535057: Collaborative Research: Curriculum Development: Digital Libraries, $272,187 + $3,200 REU Supplement (all to VT), as well as $262,407 to partners at UNC-CH under IIS-0535060 for 1/1/2006 - 12/31/2009, PI: Fox, co-PIs at UNC-CH: Barbara Wildemuth and Jeffrey P. Pomerantz.
45. Microsoft Research Digital Memories (Memex): Beyond Human Memory: SenseCam Use in Veterinary College and as Assistive Technology. $49,894 plus 2 SenseCam devices & software, 1/2006 - 1/2007, PI: Perez-Quinones, co-PI: Fox
46. NSF DUE-0532825: Personalization of Content: Bridging the gap between NSDL and its users through the course website, $449,912, 9/1/2005 - 8/31/2008, PI: Manuel A. Perez, co-PIs: Edward A. Fox, Lillian N. Cassel, and Weiguo Fan
47. IMLS: Study of User Quality Metrics for Metasearch Retrieval Ranking, $65,592, for 10/1/2004 - 12/31/2006, subcontract from Emory U. on grant from IMLS, sole PI: Fox.
48. NSF DUE-0435059: Collaborative Project: Superimposed Tools for Active Arrangement and Elaboration of Educational Resources, $397,914, 9/1/04-8/31/06, PI: Lois Delcambre and co-PI David Maier (Portland State U.) including subcontract of $107,001 to VT with Fox as sole local PI; plus collaborative project at Villanova of $101,593
49. NSF: Reformulating General Engineering and Biological Systems Engineering at Virginia Tech, $996,238, 9/1/04 - 8/31/07, PI: Vinod Lohani, co-PIs: G.V. Loganathan, Greg Adel, and E. Fox (5%). www.dlr.enge.vt.edu
50. Argyll Foundation: Relief for Canine Hip Dysplasia Related Pain through Reiki, $8,500, 8/1/04-7/31/05, PI: Marie Suthers-McCabe, co-PIs: Peter K. Shires and Edward A. Fox.
51. NSF IIS-0307867: Extending retrieval with stepping stones and pathways, $124,250, 9/1/03-8/31/05, sole PI: Fox.
52. CWRU (subcontracting to VT for NSF IIS-0325579): Information Technology Research: Managing complex information applications: An archaeology digital library, Project PI: J. Flanagan (CWRU), VT PI: Fox, VT co-PI: Weiguo Fan; subcontract: $189,500, 9/1/03-12/31/05.
53. NSF DUE-0333531: Collaborative Project: The OCKHAM Library Network, Integrating the NSDL into Traditional Library Services, NSF, $99,232, 9/1/03-8/31/06.
54. National Institute of Aerospace: Development of New Innovative Digital Library, $40,000, 12/1/03-5/31/05, sole PI: Fox.
55. Indo-US S&T Forum: Open Digital Libraries and Interoperability Workshop, June 23-25, 2003, Holiday Inn at Ballston, Arlington, VA, $50,826. Principal investigator. Co-chairs for the workshop: Shalini Urs, Mohammad Zubair, N. Balakrishnan.
56. SUN: Equipment donation in support of Computing and Information Technology Interactive Digital Educational Library (CITIDEL), 6/30/2003, $51,119
57. SOLINET: AmericanSouth.org: A Collaborative Project to Improve Access to Digital Resources on Southern History and Culture: $52,235 for 1/1/2002 - 2/28/2003. Project director: E. Fox (subcontract on Mellon award to SOLINET).
58. NSF DUE-0121741 (414201): Intelligent Collection Services for and about Educators and Students: Logging, Spidering, Analysis and Visualization: $425,000 for 9/10/2001-8/31/2003. PI: Hsinchun Chen (U. Arizona), Co-PI: Ann Lally (U. of Arizona), Co-PI: Edward Fox (VT) - with VT subcontract $104,773.

59. NSF DUE-0136690 (414214): A Technical Service Enhancing Interoperability of NSDL Collections and Services. $450,000 for 9/15/2001-9/30/2003. PI: Su-Shing Chen (University of Florida), Co-PI: Joe Futrelle (NCSA, University of Illinois, Urbana-Champaign), Co-PI: Edward A. Fox (VT) - with VT subcontract $142,501 for 1/1/2002 - 8/31/2003.
60. NSF DUE-0121679 (427106): Computing and Information Technology Interactive Digital Educational Library (CITIDEL): $800,000+35,000 for 9/1/2001 - 5/31/2005. Principal investigator Edward A. Fox. Co-PIs: John A. Lee, Manuel A. Perez, Lillian Cassel (Villanova), C. Lee Giles (Penn State), John Impagliazzo (Hofstra), Deborah Knox (College of New Jersey)
61. e-Numerate Solutions Inc. (414156): Integrating Digital Library and e-Numerate Technologies: $40,416 for 8/16/2001 - 8/15/2002. Project director: E. Fox
62. NLM: SIS/Virginia Tech Advanced Digital Library for Toxicology and Environmental Health Databases, Research Collaboration Program: $40,000+$5,500 for 5/16/2001 - 5/15/2002. Project director: E. Fox. (Processed through ORISE, with part to Virginia Tech, and tuition and stipend directly to the graduate assistants.)
63. NSF IIS-0122201 (427075): June 2001 Workshop for Digital Libraries Initiative Principal Investigators: $37,948 for 5/1/2001-6/30/2002. Project director: E. Fox.
64. Internet TIC (one of the awards related to account 437510): Extending Collaboration Between Internet TIC and Virginia Companies Regarding Digital Libraries, Information Retrieval, Networking, and Knowledge Management, $19,000 for 5/16/2001 - 12/31/2001. Principal Investigator Edward Fox. (Also, $10,000 for 8/10/2001 - 5/9/2002, as part of $30K for ODU/UVA/VT on this effort. Also, $10,000 for 3/15/2001 - 6/30/2002; $26,000 for 10/15/2001-6/30/2002.)
65. NSF IIS-0080748 (427043): High Performance Interoperable Digital Libraries in the Open Archives Initiative: $99,999 for 3/1/2001-3/31/2003. Project director: E. Fox (with CONACyT co-PIs J.Alfredo Sanchez, Universidad de las Americas-Puebla --- UDLA, and David Garza-Salazar, Monterrey Technology Institute --- ITESM, both funded by CONACyT in Mexico)
66. NSF IIS-0086227 (427042): Open Archives: Distributed services for physicists and graduate students (OAD): $315,259 for 3/1/2001-2/28/2004 (continuing over 3 years with remaining portions of funds). Project director: E. Fox (w. Royce Zia, Physics, VT, and E. Hilf, U. Oldenburg, PI on matching German DFG project)
67. AOL (433837): AOL Fellowship in Information Retrieval: $140,000 for 2001-2004. Project director: E. Fox
68. UNESCO (433800): International Guide for the Creation of Electronic Theses and Dissertations: $13,000 for 12/28/2000-3/31/2002. Project director: E. Fox.
69. SOLINET: Networked Digital Library of Theses and Dissertations: $1000 to establish Virginia Tech Foundation account 872037 for NDLTD, 2000. Project director: E. Fox.
70. PCGI (433766): Internet Development at Public Consulting Group, Inc., $7825 + $15,425 for 10/1/2000 - 5/15/2001. Project director: E. Fox
71. NSF IIS-0090153 (427963): US-Korea Joint Workshop on Digital Libraries: Removing Barriers to International Collaboration on Research and Education through Digital Libraries, $24,472 for 8/1/2000-12/31/2001, extended to 9/30/2002. Project director: E. Fox (with co-PIs Ronald L. Larsen, U. Md. and Reagan W. Moore, San Diego Supercomputer Center).
72. IBM (438220): IBM Faculty Partnership Award (UPP): $20,000, 8/15/2000-2/28/2002. Project director: E. Fox (on superstorage systems and video servers).
73. UNC Wilmington (434091): A Digital Library of Reusable Science and Math Resources for Undergraduate Education: $78,374+$10,000 for 5/1/2000 - 12/31/2002. Project director: E. Fox. (subcontract for IIS-0002935, originally CCRI proposal NSF 98-63 for $1,143,282, DLI-2, Undergraduate Emphasis)
74. NSF IIS-9986089 (SGER) (427905): Core Research for the Networked University Digital Library (NUDL): $79,997 for 1/1/2000 - 4/30/2002. Project director: E. Fox.
75. Advance Auto Parts: Internet Technology Innovation and Visioning with Advance Auto Parts: $193,014 for 1/7/2000 - 12/29/2000. Project director: E. Fox. Co-investigators at VT: N. Ramakrishnan, J. Hicks, L. Matheson. Co-investigators at UVA: J. French, A. Weaver.

76. NSF IIS-9905026: 1999 NSF Information and Data Management Workshop: Research Agenda for the 21st Century - IDM'99: $105,782 for 3/1/99 - 1/31/01. Project director: Edward A. Fox.
77. IBM: SUR Program Equipment Donation valued at approximately $120K for various computers for digital library research, Fall 1998. Project director: E. Fox.
78. CIT (437510): Internet Technology Innovation Center (ITIC): $2M for 5 years starting on 10/1/98. Project director at Virginia Tech: Edward A. Fox, also serving as director of a new university center to handle this (ITIC@VT). Virginia Tech is one of 4 universities in this virtual center, with UVA administering the grant from the Virginia Center for Innovative Technology. See http://www.internettic.org
79. NLM: Advanced Information Retrieval for Toxicology and Environmental Health Databases: $25,000 for 5/16/98 - 5/15/99. Project director: Edward A. Fox (processed through ORISE, with $7754 to Virginia Tech and tuition and stipend directly to the graduate assistant). Project period and funds are being extended into 2000.
80. NSF DUE-9752408: Curriculum Resources in Interactive Multimedia (CRIM): $87,000 for the DUE CCD program 97-29, 1/1/98 - 12/31/2000. Project director: Edward A. Fox.
81. NSF DUE-9752190: A Digital Library Based Computer Science Teaching Center (CSTC): $72,820 for 1/1/98-2/29/2000. Project director: Edward A. Fox. Subaward for proposal to NSF for the DUE CCD program 97-29 granted to D. Knox of The College of New Jersey
82. NSF NCR-9616956: A High Performance Connection for Research and Education Institutions and Facilities in Virginia: $350,000 for 1/1/97 - 12/31/98. Principal investigators: E. Blythe, E. Fox, J. Crowder.
83. OCLC: SiteSearch software license: $50,000 for 1997-1999 for electronic thesis and dissertation research. Project director: E. Fox.
84. IBM: SUR Program Equipment Donation valued at approximately $400K for various computers for digital library research, Dec. 1996. Project director: E. Fox.
85. NSF NCR-9627922: World-Wide Web Traffic Characterization with Application to In-Network Caching and Prefetching: $320,559 for 9/1/96 - 8/31/99. Principal investigators: M. Abrams, E. Fox, J. Pollard. REU supplement summer 1998: $5000.
86. U.S. Dept. of Education, FIPSE Program P116B61190: Improving Graduate Education with a National Digital Library of Theses and Dissertations: $208,040 requested for 9/1/96-8/31/99 (federal/nongovernmental percentages 53.3% / 46.7%. Principal investigators: E. Fox, J. Eaton, G. McMillan. In-kind cost sharing of $30K promised by SURA, and supporting software donations promised from Microsoft (est. value $107,820) and Adobe (est. value $100K). See final report http://www.ndltd.org/pubs/FIPSEfr.pdf
87. HAL Computers: HALstation 330 (64 bit) donated as follow-up to earlier license of hashing technologies, 1996. Est. value $10K. Principal investigators: E. Fox, L. Heath.
88. IBM: Information Retrieval Systems Evaluation: $11,002 for 5/22/96 - 3/15/97. Principal investigator: E. Fox.
89. SURA: Development and Beta Testing of the Monticello Electronic Library Thesis and Dissertation Program: $90,117 for 1/1/96 - 12/31/97. Principal investigators: E. Fox, J. Eaton, G. McMillan.
90. NSF DGE-9553458: Graduate Research Traineeship in Human Interface Design for Access to Computers and Networked Information at Virginia Tech. $562,500 for 9/5/95-8/31/2000. PIs: Carroll, J. (project director), Abrams, M., Ehrich, R., Rosson, M.B, Hix, D., Hartson, R., Shaffer, C., and Fox,E.A.
91. NSF SUCCEED: Using Computers and Networked Information: Distance Learning with Networked Multimedia: $35,432 for 5/15/95 - 8/15/97. Principal investigators: E. Fox, M. Abrams, N. D. Barnette.
92. IBM: SUR Program Equipment Donation valued at approximately $250K for 4-processor SMP machine for digital library research, Dec. 1995. Project director: E. Fox.
93. NSF: Building a History of the Blacksburg Electronic Village: $24,866 for 10/15/94 - 8/15/95. Principal investigators: J. Carroll, A. Cohill, G. Downey, E. Fox, and M.B. Rosson.
94. PRC Inc.: Hypertext and Automatic Document Indexing: $25,308 for 5/16/94-12/31/94. $27,592 extension to 8/15/95. Principal investigator: E. Fox.

95. NSF SUCCEED (Cooperative Agreement No. EID-9109853): Network Multimedia File System with HyTime: $60,000 for 8/16/93 - 2/15/95. Principal investigators: E. Fox (proj. dir.), M. Abrams, and S. Newcomb (of Florida State Univ., receiving $15K of the total).
96. NSF CISE Institutional Infrastructure (Education) Grant CDA-9312611: Interactive Learning with a Digital Library in Computer Science: $449,088 for 8/15/93 - 7/31/97. Principal investigators: E. Fox (project director), J. Lee, C. Shaffer, H. Hartson, D. Barnette. Supplement $67,802 in 1997 for workshop, EI home page, digital library courseware.
97. NSF Research Infrastructure CDA-9303152: Interactive Accessibility: Breaking Barriers to the Power of Computing: $1,375,000 for 7/1/93 - 6/30/98. Principal investigators: R. Ehrich, E. Fox, H. Hartson, D. Hix, R. Williges. Director of Information Access Laboratory.
98. NIST/DARPA, PRC Inc.: Text Retrieval Methodologies: $6,000 from DARPA for 1 year starting 4/1/93. Additional support for project provided by PRC Inc.: $11,489 for 1993. Principal, only investigator.
99. National Science Foundation CISE Grant NSF-CDA-9308259: Wide Area Technical Report Server: $22,736 for 1993. Principal investigators: K.Maly (ODU), E. Fox, J. French (UVA), and A. Selman (SUNY Buffalo). Awarded to Maly (ODU) with $4200 subcontract to VPI&SU.
100. Intel Corporation: Multimedia Research and Enhanced Surrogate Travel Simulations Using DVI Technology: $5,845 in equipment, Oct. 1992. Prin. investigators: L. Carstensen, E. Fox.
101. National Science Foundation CISE/CDA Grant: High Data Rate Network Research: $46,000 for 6/1/92 - 5/31/95. Principal investigators: M. Abrams, J.A. Wiencko, S.F. Midkiff, N.J. Davis, and E.A. Fox.
102. Dept. of Education (FIPSE): GeoSim: A GIS-Based Simulation Laboratory for Introductory Geography: $149,571 for 9/9/92-5/15/95. Principal investigators: C. Shaffer; L. Carstensen, R. Morrill, and E. Fox.
103. National Science Foundation Grant 9155943: GeoSim: A GIS-Based Simulation Laboratory for Introductory Geography: $107,940 for 5/1/92 - 10/31/93. Principal investigators: C. Shaffer; L. Carstensen, R. Morrill, and E. Fox.
104. NIST/DARPA, PRC Inc.: Advanced Routing and Retrieval from a Large Collection: $10,000 from DARPA for 1 year starting 2/1/92. Additional support for project provided by PRC Inc.: $28,571 for 2/1/92 - 12/30/92. Principal and only investigator.
105. Intel Corporation: Multimedia Research and Enhanced Surrogate Travel Simulations Using DVI Technology: $34,752 in equipment, November 1991. Principal investigators: L. Carstensen, E. Fox.
106. National Science Foundation Grant IRI-9116991: A User-Centered Database from the Computer Science Literature: $443,391 for 9/15/91 - 2/28/95, plus $12,000 REU supplements, plus $29,941 equipment supplement. Principal investigators: E. Fox, D. Hix and L. Heath.
107. NCR Corporation: Usable Environments for Developing Interactive Multimedia Applications, $20,000 for 5/15/91 - 12/31/91. PIs: E. Fox, D. Hix and E. Schwartz.
108. Virginia Center for Innovative Technology: Software for Large Multimedia Databases Including Facsimile Images: $8,000 for 6/1/91 - 5/31/92. Principal and only investigator.
109. PRC Inc.: Software for Large Multimedia Databases Including Facsimile Images: $20,000 for 6/1/91 - 5/31/92. Principal, only investigator.
110. NCR Corporation: Evaluation of Tools for Developing Interactive Multimedia Applications: $20,000 for 3/1/91 - 9/30/91. Principal investigators: E. Fox, D. Hix, and E. Schwartz.
111. NCR Corporation: Prototyping with New Metaphors for Describing Interactive Multimedia Applications: $20,000 for 1/1/91 - 6/30/91. Principal investigators: E. Fox, D. Hix, and E. Schwartz.
112. NCR Corporation: Extending the Integrator Component of an Interactive Multimedia Application Development Environment: $7763 for 1/1/91 - 5/31/91. Principal investigators: E. Fox, D. Hix, and E. Schwartz.
113. Chemical Abstracts Service: An Evaluation of Access Approaches to the Chemical Literature: $25,133 for 1/1/91 - 8/15/91. Principal investigators: E. Fox and J. Merola.
114. West Publishing Company: Advanced Retrieval Methods for Full Texts of Legal Documents: $74,717 for 1/1/91 - 12/31/91. Principal and only investigator.

115. NCR Corporation: Developing Digital Multimedia Applications for the General Public. $60,000 for 1/1/90 - 12/31/90. Principal investigators: E. Fox, D. Hix, and E. Schwartz.
116. Columbia University (subcontract under IAIMS grant from National Library of Medicine): Cardiology-CODER: $20,000 for 12/1/89-5/15/91. Principal and only investigator.
117. Online Computer Library Center (OCLC) Inc.: Advanced Retrieval Methods for Online Catalogs: $10,000 for 7/1/89-6/30/90. Principal and only investigator.
118. Naval Surface Warfare Center: Advanced Naval Message Analyzer: $26,594 for 7/10/89 - 6/9/90. Principal and only investigator.
119. Virginia Center for Innovative Technology: A Prototype Geographic Information System for Personal Workstations: $33,890 for 4/1/89-3/31/90. Principal investigators: Clifford A. Shaffer, James B. Campbell, Laurence W. Carstensen, Jr., Edward A. Fox.
120. General Dynamics: A Prototype Geographic Information System for Personal Workstations: $43,000 for 4/1/89-3/31/90. Principal investigators: Clifford A. Shaffer, James B. Campbell, Laurence W. Carstensen, Jr., and Edward A. Fox.
121. Virginia Center for Innovative Technology Grant INF-89-010: Interactive Digital Video Authoring and Prototyping. $50,000 for 5/1/89-4/30/90. Principal investigators: Edward A. Fox, Deborah Hix, and E. Schwartz.
122. NCR Corporation: Interactive Digital Video Authoring and Prototyping. $60,000 for 2/15/89-2/14/90. Principal investigators: Edward A. Fox, Deborah Hix, and E. Schwartz.
123. Council on Library Resources Grant CLR 4048-C: Comparison of Advanced Retrieval Approaches for Online Catalog Access. $3000 for 1/1/89- 6/30/90. Principal investigators: Edward A. Fox and Linda Wilson (Reference Dept., Newman Library, VPI&SU).
124. NCR Corporation: Prototype Application of Advanced Methods for Information Storage and Retrieval. $44,691 for 1/1-12/31/89. Principal and only investigator.
125. Nimbus Records, Inc. Grant: CD ROM Based Publishing and Access to Composite Document Databases. $9000 from CIT plus $10,000 from Nimbus (and $5000 in-kind) for 6/1/88-10/1/88. Principal and only investigator.
126. National Agricultural Library and Extension Service - USDA: Joint Assessment of Alternative Approaches for Database Development and Distribution. $41,400 for 1988-9. Principal investigators: Mary G. Miller, Michael T. Lambur, Thomas R. McAnge, and E. A. Fox.
127. National Science Foundation Grant IRI-8703580: Organizing Lexical Knowledge for Information Retrieval. $149,900 plus $8,284 REU Suppl. for 8/1/87-6/30/90. Prin. investigators: E. Fox and J.T. Nutter.
128. Council of Higher Education in Virginia Grant: Cooperative Library Services Program: The Virginia Disk: An Evaluation of Locally Produced CD ROM Products. $21,735 for 6/15/87 - 9/15/88. Principal investigators: E. Fox and P. Gherman.
129. Virginia Center for Innovative Technology Grant INF-87-012: CD ROM Based Publishing and Access to Composite Document Databases. $24,500 for 5/1/87-7/31/88. Renewal $9,000 through 1/15/89. Principal and only investigator.
130. Nimbus Records, Inc. Grant: CD ROM Based Publishing and Access to Composite Document Databases. $24,500 from CIT plus $12,000 from Nimbus (and $5000 in-kind from Highlighted Data, Inc.) for 5/1/87-7/31/88. Principal and only investigator.
131. AT&T Foundation Equipment Donation. $711,090 for 1987. Prin. invest's: D. Kafura, E. Fox. Virginia Center for Innovative Technology Grant INF- 85-016: Improved Automatic Analysis, Indexing, Storage and Retrieval of Corporate Documentation and Network Information. $82,500 (renewal) for 1/1/86-9/15/87. Principal and only investigator.
132. National Science Foundation Grant IST-8418877: Effective Retrieval of Composite Documents. $90,760, 2/15/85-7/31/87. Prin. investigator.
133. AT&T Foundation Equipment Donation. $7,580 in Dec. 1986. $50,030 in Dec. 1985. Principal and only investigator.
134. Virginia Center for Innovative Technology Grant INF-85-016: Improved Automatic Analysis, Indexing, Storage and Retrieval of Corporate Documentation and Network Information. $51,001 (to VPI&SU and $45,871 to George Mason Univ.) for 1/1/85-12/31/85. Principal investigators: M. Koll (GMU) and E. Fox (VPI&SU).
135. Virginia Center for Innovative Technology Contract C-85-026: Development of Prototype Electronic Mail and Bulletin Board Distributed System to Facilitate CIT Operations. $21,479 for 3/15/85-3/15/86. Principal and only investigator.

136. VPI&SU College of Arts and Science Small Proj. Grant 1895820: Information Retrieval for Office Information Systems. $1,600 for 12/1/83-11/30/85. Principal and only investigator.

SUMMARY:

>135 grants with funding from Advance Auto Parts, AOL, ARPA, AT&T, BAE Systems, CAS, Dept. of Educ., Google, IBM, IMLS, Intel, Mayfair, Mellon, Microsoft, Natl. Ag. Lib., NCR, NEH, Nimbus Records, NIH, NIJ, NIST, NLM, NSF, NSWC, OCLC, PRC Inc., QNRF, Sun, UNESCO, U.S. Dept. of Education, Virginia CIT, ...

ALSO, CONSULTANT ON:

- Patent cases, addressing invalidity, non-infringement, IPR, etc.
- US Intelligence Community: Massive Parallel Object Store, 12/1/94 - 12/1/95, to Knowledge Systems Inc., Principal investigators: R. Akscyn, D. McCracken.
- National Archives and Records Administration: Access to Archival Databases (AAD), $648,282.50 contract to SAIC for 1999-2000

KEYNOTE/BANQUET/INT'L INVITED/DISTINGUISHED SPEAKER PRESENTATIONS:

1. Edward A. Fox. Building and Using Digital Libraries. Proc. Electronic Theses and Dissertations, ETD 2020: Discovery, Data, and Dates, virtual conference, Nov. 16-18, 2020, UAE, to be presented
2. Edward A. Fox, How should one explore the digital library of the future? Invited keynote for JCDL 2020, Aug. 1-5, virtual conference hosted by Wuhan University, https://doi.org/10.1145/3383583.3398496
3. Edward A. Fox, Applications of Big Data Analysis for the Worldwide Collection of ETDs. Invited keynote for ETD 2018, Taiwan, Sept. 26-28, 2018, http://fox.cs.vt.edu/talks/2018/20180926FoxETD2018keynote.pptx
4. Edward A. Fox, Living In the KnowlEdge Society: the double duty of a librarian, invited speaker (by videoconference) at Linking Research and Education in Digital Libraries Workshop at TPDL 2011, 28-29 Sept. 2011, Berlin
5. E. Fox. A formal approach to digital libraries - the 5S framework: Societies, Scenarios, Spaces, Structures, Streams. Invited talk for Seminar Series, Department of Computer Science at Georgetown University, Washington, D.C., April 29, 2011
6. E. Fox. Digital Library Support for 1) Graduate Education and 2) Computing-related Education. Keynote presentation at Digital Libraries in Higher Education: Scientific Experiences, Riyadh, Saudi Arabia, 30-31 October 2010, http://fox.cs.vt.edu/talks/2010/20101030FoxRiyadhKeynote.pdf.
7. E. Fox. Digital Libraries, Scholarship, and the Humanities. Keynote presentation (as Guest of Honor for the conference), Libraries in the Digital Age (LIDA) 2010, Zadar, Croatia, 24-28 May 2010, University of Zadar, Zadar, Croatia (http://ozk.unizd.hr/lida/2010/, video: https://meduza.carnet.hr/index.php/media/watch/11219)
8. E. Fox. Opening and Expanding Digital Library Services. Invited keynote for Canadian ETD and Open Repositories Workshop, May 10-11, 2010, Carleton University, Ottawa
9. Digital Libraries: From Theory to CS/LIS Curricula, invited speaker for the Lazerow lectures, C.W. Post Campus, Long Island University, Brookville, New York, April 23, 2008

10. Digital Libraries: From Proposals to Projects to Systems to Theory to Curricula, invited keynote for DELOS Conference 2007, 14 Feb. 2007, Pisa, Italy, http://fox.cs.vt.edu/talks/2007/20070214DELOSkeynotePisa.ppt
11. Digital Libraries for Education: Lessons Learned and Promising Opportunities, invited keynote for LODL 2006, Workshop on "Learning Object Repositories as Digital Libraries: Current Challenges", ECDL 2006, Alicante Spain, 22 Sept 2006 (by videoconference)
12. Improving Graduate Education with Digital Libraries, invited keynote for the Inauguration of the DFG Graduiertenkollegs 1223 "Quality improvement in e-Learning by feedback processes" (Eroffnung des DFG-Graduiertenkollegs 1223 Qualitatsverbesserung im E-Learning durch ruckgekoppelte Prozesse), 14 Sept. 2006, Darmstadt, GE
13. A 5S Perspective on Digital Libraries for E-Learning: With case studies from Archaelogy, Computing, and Dissertations, invited keynote for DeLFI 2006 (Die 4. e-Learning Fachtagung Informatik), 14 Sept. 2006, Darmstadt, GE
14. Digital Libraries: 1991-2006 and beyond, with electronic theses and dissertations. Invited videoconference (since delayed 1 day in travel) presentation to approximate 300, at several sites and at University of Sao Paulo (USP-Brazil), for event to commemorate 5 years of electronic theses and dissertations (www.teses.usp.br). Aug. 23, 2006
15. Digital Library Curricula and Modules Informed by 5S, invited keynote for Workshop 6: Developing a Digital Libraries Education Program, JCDL 2006, June 11-15, 2006, Chapel Hill, NC
16. 5S and the Reference Model, invited talk at DELOS Reference Model Workshop, Frascati-Rome, June 1-2, 2006
17. From the WWW and Minimal Digital Libraries, to Powerful Digital Libraries: Why and How. Invited talk in Fox, E.A.; Neuhold, E.; Premsmit, P.; Wuwongse, V. (Eds.) Digital Libraries: Implementing Strategies and Sharing Experiences; Proceedings 8th International Conference on Asian Digital Libraries, ICADL 2005, Bangkok, Thailand, December 12-15, 2005, Lecture Notes in Computer Science, Vol. 3815, XVII, 529 p., ISBN: 3-540-30850-4, p. 525, DOI: 10.1007/11599517_74, http://fox.cs.vt.edu/talks/2005/20051213ICADLinvited.ppt
18. Digital Libraries: Archaeology, Automation, ETDs, and Enhancements. Invited talk. The International Advanced Digital Library Conference (IADLC), Nagoya University, Japan, August 25-26, 2005, http://fox.cs.vt.edu/talks/2005/20050823Nagoya.ppt
19. Digital Libraries for Education: Case Studies. Invited talk. In Zhaoneng Chen, Hsinchun Chen, Qihao Miao, Yuxi Fu, Edward Fox, Ee-Peng Lim, eds. Digital Libraries: International Collaboration and Cross-Fertilization; Proceedings 7th International Conference on Asian Digital Libraries, ICADL 2004, Shanghai, Dec. 2004. Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 3334, pp. 51-60, http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=3334&spage=51
20. Towards a Digital Library Theory: A Formal Digital Library Ontology. Invited talk at University of Athens, Greece, Dec. 9, 2004.
21. Towards a Quality Model for Digital Libraries. Invited talk for the DELOS workshop on digital library evaluation, Padova, Italy, October 4-5, 2004. Marcos Andre Goncalves, Edward A. Fox, Baoping Zhang, Layne T. Watson.
22. Digital Libraries of the Future: Integration through the 5S Framework. Keynote talk for RCDL'2004, the Sixth National Russian Research Conference: Digital Libraries: Advanced Methods and Technologies, Digital Collections, Sep. 30, 2004, Pushchino, Russia
23. Digital Libraries for Education: Foundations to Case Studies. Invited talk. Also, Chair of the conference for the morning of 7 September. In Proceedings Digital Library --- Advance the Efficiency of Knowledge Utilization, 2nd International Digital Library Conference, hosted by National Library of China, Beijing, September 6-8, 2004.
24. Case Studies in the US National Science Digital Library: DL-in-a-Box, CITIDEL, and OCKHAM. Invited talk in: Tengku Mohd Tengku Sembok, Halimah Badioze Zaman, Hsinchun Chen, Shalini R. Urs, Sung Hyon Myaeng, eds. Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access; Proceedings 6th International Conference on Asian Digital Libraries, ICADL 2003, Kuala Lumpur, Malaysia, Dec. 2003. Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2911, 17-25, http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2911&spage=17
25. Improving Graduate Education: Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for International Conference on University Libraries: Electronic publishing and library services, Mexico City, Oct. 23, 2003, 82 slides

26. Research Challenges to Semi-Automatically Enhance Quality in Distributed Open Archives, invited presentation for SINN03, Oldenburg, GE, 9/18/2003, 75 slides
27. Closing Keynote invited for ETD'2003 (The 6th Int'l Symp. on Electronic Theses and Dissertations), Humboldt U., Berlin, GE, May 23, 2003
28. The Future of Electronic Publishing, invited presentation for Symposium on The Library of the Future, Humboldt U., Berlin, GE, May 19, 2003 (at opening of new building)
29. Keynote on "Human Resources and the Information Professional in the Digital Age" for Madras Library Association (MALA) Platinum Jubilee Celebration, Chennai, India, Jan. 29, 2003 (videoconference presentation arranged by US State Dept.), http://smp2.cc.vt.edu/~fox/tmp/Chennai20030129Fox.ppt
30. ETD Digital Library, invited presentation for International Conference on Scientific Electronic Publishing in Developing Countries, Sept. 30 - Oct. 2., 2002, Valparaiso, Chile
31. From Electronic Theses and Dissertations to Supporting Teaching and Learning in Science, Technology, Engineering and Mathematics - The Benefits of Digital Libraries. Invited presentation for University of Chile, Sept. 30, 2002 (for ~200 librarians, etc. in region), Santiago, hosted by Gabriela Ortuzar
32. From Electronic Theses and Dissertations to Supporting Teaching and Learning in Science, Technology, Engineering and Mathematics - The Benefits of Digital Libraries, invited presentation for ISN Sommerfest, September 13, 2002, Oldenburg, Germany
33. Advancing Education through Digital Libraries: NSDL, CITIDEL, and NDLTD. Invited presentation, Proc. Digital Library: IT Opportunities and Challenges in the New Millennium, July 9-11, 2002, Beijing Library Press, Beijing, China, 107-117
34. ETD Roots: History of the ETD Initiative. Keynote invited for ETD'2002 (The Fifth Int'l Symp. on Electronic Theses and Dissertations: Pioneering on the Electronic Frontier: e-theses and Intellectual Solidarity), BYU, Provo, Utah, May 30 - June 1, 2002
35. Future of the ETD Initiative. Keynote invited for ETD'2002, BYU, Provo, Utah, May 30- June 1, 2002
36. Digital Library as Infrastructure for Knowledge Management in Academic and Research Institutions: Open Archives Initiative and Education Demonstrations (NDLTD, CITIDEL/NSDL), Construyendo una plataforma de tecnologia integrada para la administracion del conocimiento. Opening keynote for Amigos 2002 Conference, Comunicacion del conocimiento basada en la tecnologia de la informacion, The Mexican Network for Library Cooperation, http://biblio.pue.udlap.mx/congress/, Puebla, Mexico, Feb. 21, 2002. (7 page paper) (Also, panel moderator, with panel title "Construyendo una plataforma de tecnologia integrada para la administracion del conocimiento".)
37. Overview of Digital Libraries. Inaugural speaker in seminar series for the ASDC staff - Atmospheric Sciences Data Center (the former EOSDIS DAAC) - and related groups - full-day tutorial on digital libraries at NASA LaRC (NASA Langley), Hampton, VA, January 18, 2002, http://ei.cs.vt.edu/~dlib/tut/NASA20020118.htm
38. International Collaboration on Digital Libraries. Invited keynote presentation, International Conference on Asian Digital Libraries, ICADL'2001, Bangalore, India, Dec. 10-12, 2001.
39. Networked Digital Library of Theses and Dissertations: New Services, Standards, and Integration with the Open Archives Initiative. Invited keynote at the workshop Jornadas sobre Gestion de la Informacion en el siglo XXI, JOURNEYS ON INFORMATION MANAGEMENT ON XXI CENTURY, Montevideo, Uruguay, 12-14 November.2001 sponsored by UNESCO, ISTEC, the University of Republica of Uruguay and the ORT University
40. Terabyte scale multilingual digital library services and harvesting from Open Archives. Invited plenary for NIT'2001, Beijing, China, May 29-31, 2001. In Global Digital Library Development in the New Millennium: Fertile Ground for Distributed Cross-Disciplinary Collaboration, ed. Ching-chih Chen, Tsinghua U. Press, pp. 59-70
41. Status Report on the NDLTD. Invited closing plenary for Fourth International Symposium on Electronic Theses and Dissertations (ETDs), Caltech, Pasadena, CA, March 22-24, 2001, http://www.ndltd.org/talks/etd01key.ppt (plus helped run standards meeting, chaired 2 sessions, hosted a "networking" table on NDLTD future directions, coordinated meeting of working group on the UNESCO International Guide for ETDs)

42. Digital Libraries: From Theory to Applications in Education and Business. Invited keynote presentation, 2000 Asian DL conference, ICADL'2000, Seoul, Korea, Dec. 6-8, 2000. http://www.ndltd.org/talks/koreainv.ppt
43. Digital Libraries: Extending and Applying Library and Information Science and Technology, invited keynote for ACM CIKM 2000, McLean, VA, Nov. 9, 2000. http://www.ndltd.org/talks/cikm2000.ppt
44. The Co-Evolution of Digital Libraries and the WWW. Invited keynote for WebNet '2000, San Antonio, Nov. 4, 2000. http://www.ndltd.org/talks/webnet.ppt
45. Digital Libraries: Extending the Impact of Information Science, sole keynote for final DISSAnet Conference, Pretoria, South Africa, Oct. 26-27, 2000. DISSAnet is a project between the Royal School of Library and Information Science in Copenhagen and the Universities of Pretoria and the North in South Africa. It is funded by DANIDA. http://www.ndltd.org/talks/prolissa.ppt
46. Digital libraries, universities, and scholarly communication, keynote for EUNIS'2000, Poznan, Poland, April 14, 2000. http://www.ndltd.org/talks/poznan.ppt
47. Future Funding and the NDLTD Vision. Invited closing plenary for Third International Symposium on Electronic Theses and Dissertations (ETDs), St. Petersburg, FL, March 18, 2000 http://www.ndltd.org/talks/ETD3close.ppt
48. The 5S Framework for Digital Libraries and Two Case Studies: NDLTD and CSTC, keynote for NIT99 (http://artemis.simmons.edu/~cchen/nit/), The 11th International Conf. on New Information Technology, Taipei, Taiwan, Aug. 18, 1999, Word: http://www.ndltd.org/talks/nit99.ppt
49. Networked Digital Library of Theses and Dissertations. Sole keynote for 15th Workshop on Digital Library (DLW15), Nara Inst. for Sci. Tech. (NAIST), Japan, July 19, 1999. PowerPoint: http://www.ndltd.org/talks/dlw15.ppt, Word: http://www.ndltd.org/pubs/dlw15.doc
50. From Theory to Practice in Digital Libraries: 5S and Educational Applications (NDLTD, CSTC), Keynote at NSF-CONACyT Workshop on Digital Libraries, Albuquerque NM, July 7-9, 1999, http://ict.udlap.mx/dl/workshop99/), Word: http://www.ndltd.org/pubs/conacyt99.doc
51. Digital dissertations network worldwide, keynote presentation for joint session of Dissertation Online Workshop and IuK'99 - Dynamic Documents, University of Jena, Germany, March 24, 1999 http://elfikom.physik.uni-oldenburg.de/IuK/IuK99
52. Networked Digital Library of Theses and Dissertations: A Framework for East-West Collaboration, keynote talk, paper in Proc. First Asian Digital Library Workshop, "East meets West," August 5-7, 1998, Hong Kong, hosted by University of Hong Kong. http://www.ndltd.org/talks/Asia.htm
53. Digital Libraries: The Networked Digital Library of Theses and Dissertations (NDLTD) and the Computer Science Teaching Center (CSTC). Keynote for Computer Science Workshop, sponsored by CONACyT and NSF, Puebla, Mexico, June 10-12, 1998. http://www.ndltd.org/talks/CONACyTNSF.pdf http://www.ndltd.org/talks/CONACyTNSF.ppt
54. Improving Education through the Networked Digital Library of Theses and Dissertations (NDLTD) and the Computer Science Teaching Center (CSTC), invited international presentation for the Russian-American DL Workshop, http://www.risi.ru/radlw98/indexe.htm Moscow, April 16-17, 1998.
55. Digital Libraries: Their Educational Applications and Uses. Opening presentation, keynote session for Web Week and first talk in the Rice University Fondren Library and Information Technology 1997 Lecture Series "Rethinking Information Access in the Digital Age", March 17, 1997.
56. Digital Libraries through Information Retrieval for Education. Presentation for Distinguished Lecture Series, Univ. Utah Dept. of Computer Science, March 7, 1996.
57. Rethinking libraries in the information age: lessons learned with 5 digital library projects, Henderson Lecture, UNC Chapel Hill, Feb. 1, 1996.
58. Images of Digital Libraries. Invited opening keynote address. INFO Conference: Digital transfer of images, Helsinki, Finland, Nov. 10-11, 1994, 2 pg. extended abstract for conference plus longer paper for proceedings.
59. Multimedia in Education and Digital Libraries. Sole keynote speaker for Multimedia Systems: Technology and Applications, Oct. 12-13, 1994, Ottawa.

60. How to make intelligent digital libraries. Invited plenary presentation for 8th Int'l Symp. on Methodologies for Intelligent Systems (ISMIS'94), Charlotte, NC, Oct. 16-19, 1994.
61. Toward a Widely Used Hypermedia Digital Library in Computer Science. Invited plenary presentation for EG-MM '94, First Eurographics Symposium and Workshop on Multimedia: Multimedia/Hypermedia in Open Distributed Environments, June 6-9, 1994, Graz, Austria.
62. Digital Libraries: Why People Use Tools, Not AI. Invited plenary presentation for Tenth IEEE Conf. on Artificial Intelligence for Applications (CAIA), San Antonio, March 1-4, 1994.
63. From Information Retrieval to Networked Multimedia Information Access, keynote address. In G. Knorz, J. Krause, C. Womser-Hacker, eds., Information Retrieval '93, Proc. der 1. Tagung Information Retrieval '93, 13-15 September, 1993, University of Regensburg, Germany, Univ. of Konstanz Press, 116-124.
64. Multimedia Systems and Electronic Libraries, IBM/FAU Distinguished Lecture Series, April 8, 1993, Florida Atlantic Univ., Boca Raton, FL.
65. Multimedia Standards and Systems. Banquet Presentation, Third International Workshop on Network and Operating System Support for Digital Audio and Video, Nov. 12-13, 1992, La Jolla, CA.
66. Building a User-Centered Database from the ACM Literature. Invited plenary presentation for Symposium on Document Analysis and Information Retrieval, March 16-18, 1992, Tropicana Hotel, Las Vegas, Nevada, 235-246.
67. Hypermedia: Flexible Access to Multimedia Information. Invited presentation at National Institute of Standards and Technology (NIST), Gaithersburg, MD, September 27, 1991. Initial lecture in series on Hypermedia.
68. Electronic Publishing and Intelligent Information Retrieval. Invited presentation for OCLC (Online Computer Library Center, Inc.) Office of Research Distinguished Seminar Series, Feb. 10, 1989, Dublin, OH.
69. The Role of CD ROM in Libraries of the Future. Academia Keynote, Compact Disks for Information Storage and Access, Cornell Univ. Libraries sponsored conf., April 27, 1987, Cornell Univ., Ithaca NY.

TUTORIALS:

(over 86 tutorials in over 28 countries - Argentina, Australia, Austria, Belgium, Brazil, Canada, China, Costa Rica, Croatia, Denmark, England, Germany, Hungary, India, Ireland, Italy, Malaysia, Portugal, Russia, Scotland, Singapore, S. Africa, S. Korea, Spain, Switzerland, Thailand, Uruguay, USA, Vietnam, etc.)

1. Edward A. Fox and William A. Ingram. Introduction to Digital Libraries. Proc. JCDL 2020, Aug. 1-5, Virtual Event, China, half-day tutorial, 2 pages, hosted by Wuhan University, to be given, https://doi.org/10.1145/3383583.3398501
2. William A. Ingram and Edward A. Fox. Preparing Code and Data for Computational Reproducibility. Proc. JCDL 2020, Aug. 1-5, Virtual Event, China, tutorial, 2 pages, hosted by Wuhan University, to be given, https://doi.org/10.1145/3383583.3398714
3. William A. Ingram and Edward A. Fox (2019). Preparing ETD code and data for reproducible publication: a hands-on workshop. ETD 2019: Fruits of knowledge, 22nd International Symposium on Electronic Theses and Dissertations. Porto, Portugal
4. Edward Fox. Introduction to Digital Libraries, tutorial at ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2019), half-day on June 2, 2019, UIUC, Urbana-Champaign, Illinois, https://doi.org/10.1109/JCDL.2019.00111, http://fox.cs.vt.edu/talks/2019/20190602FoxTutorialSlidesJCDL.pptx
5. Edward Fox. Introduction to Digital Libraries, tutorial at ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2018), half-day on June 3, 2018, Ft. Worth, Texas. https://doi.org/10.1145/3197026.3201779, http://fox.cs.vt.edu/talks/2018/20180603FoxTutorialSlidesJCDL.pptx
6. Edward Fox. Introduction to Digital Libraries, JCDL 2017 (ACM/IEEE), Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries, pages 342–343, full-day on June 19, 2017, Toronto, Ontario, Canada.

http://fox.cs.vt.edu/talks/2017/20170613JCDL2017FoxTutorialSlides.pptx
7. Edward Fox. Introduction to Digital Libraries, JCDL 2016 (ACM/IEEE), full-day on June 19, 2016, Rutgers U., Newark, NJ. https://doi.org/10.1145/2910896.2925429, http://fox.cs.vt.edu/talks/2016/20160619JCDLFoxTutorialSlides.pptx
8. Edward Fox. Introduction to Digital Libraries, JCDL 2015 (ACM/IEEE), full-day on June 21, 2015, Knoxville, TN. https://doi.org/10.1145/2756406.2756927, http://fox.cs.vt.edu/talks/2015/20150621FoxTutorialJCDL.pptx
9. Edward Fox. Introduction to Digital Libraries; DL 2014 (ACM/IEEE JCDL + TPDL), London, 8 Sept. 2014 (full-day)
10. Edward Fox. Introduction to Digital Libraries; Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2013), Indianapolis, IN, July 22-26, 2013 (half-day)
11. E. Fox. Introduction to Digital Libraries. Refereed 1/2 day tutorial for ACM/IEEE Joint Conf. on Digital Libraries, JCDL 2011, Ottawa, June 13-17
12. E. Fox. Guidelines and Resources for Teaching Digital Libraries. Refereed 1/2 day tutorial for ACM/IEEE Joint Conf. on Digital Libraries, JCDL 2011, Ottawa, June 13-17
13. Gregory W. Hislop, Lillian N. Cassel, Lois M.L. Delcambre, Edward A. Fox, Richard Furuta, Peter Brusilovsky, and Daniel D. Garcia. Sharing Your Instructional Materials via Ensemble. Tutorial for Proceedings Consortium for Computing Sciences in Colleges - Northeastern Region, CCSCNE 2011, Western New England College, Springfield, MA, April 15-16, 2011
14. Introduction to (Teaching / Learning about) Digital Libraries. Refereed 1/2 day tutorial for Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10
15. Introduction to Digital Libraries. Refereed full-day tutorial for JCDL 2010, June 21-25, Gold Coast, Australia
16. Introduction to (Teaching / Learning about) Digital Libraries. Full-day tutorial, JCDL 2009, Austin, June 15, 2009
17. Introduction to (Teaching / Learning about) Digital Libraries. Half-day tutorial, JCDL 2008, Pittsburgh, June 19, 2008
18. Introduction to (Teaching / Learning about) Digital Libraries. Half-day tutorial, ICADL 2007, Dec. 10, 2007, Hanoi
19. Introduction to (Teaching / Learning about) Digital Libraries. Full-day tutorial (part 1, part 2), ECDL 2007, Sept. 16, 2007, Budapest
20. Introduction to (Teaching / Learning about) Digital Libraries. Half-day tutorial, ECDL 2006, Sept. 17, 2006, Alicante, Spain
21. Introduction to (Teaching / Learning about) Digital Libraries. Full-day tutorial with M.A. Goncalves, JCDL 2006, June 11-15, 2006, Chapel Hill, NC
22. Learning / Teaching Digital Libraries: An Overview. Half-day tutorial, ICADL 2005, Dec. 12-15, 2005, Bangkok, http://fox.cs.vt.edu/talks/2005/20051212ICADLtutorial.ppt
23. Digital Libraries: An Overview and Formal Framework. Full-day tutorial, ECDL 2005, Sep. 18-23, 2005, Vienna, http://fox.cs.vt.edu/talks/2005/20050918ECDLtutorial.ppt
24. Digital Libraries: An Overview and Formal Framework. Full-day tutorial with M.A. Goncalves, SIGIR 2005, Aug. 15-19, 2005, Salvador, Brazil, http://fox.cs.vt.edu/talks/2005/20050815SIGIRtutorial.ppt
25. Introduction to (Teaching / Learning about) Digital Libraries. Full-day tutorial with M.A. Goncalves, JCDL 2005, June 7-11, 2005, Denver, p. 419, https://doi.org/10.1145/1065385.1065524
26. How to Build Digital Libraries. 2 day tutorial (lecture in morning, lab session with M.A. Goncalves in afternoon) for First Latin-American Course on Building Digital Libraries, 21-25 February 2005, Fortaleza, Brazil, sponsored by UNESCO, CETREDE, IFLA-LAC, AUGM, CLEI, and UFC
27. Digital Library: Overview and Framework. 1/2 day tutorial for Digital Libraries: International Collaboration and Cross-Fertilization; International Conference on Asian Digital Libraries, ICADL 2004, Shanghai, Dec. 2004.
28. Digital Library Foundations. Half day tutorial for RCDL'2004, the Sixth National Russian Research Conference: Digital Libraries: Advanced Methods and Technologies, Digital Collections, Oct. 1, 2004, Pushchino, Russia
29. Overview of Digital Libraries. Half day tutorial for ECDL'2004, Sept 2004, Bath, England

30. Three days short course covering Digital Libraries, Information Retrieval, and Multimedia, NIDA, Bangkok, Thailand, June 21-23, 2004
31. Overview of Digital Libraries. Half day tutorial for JCDL'2004, June 2004, Tucson, AZ
32. Two days short course in Bahia Blanca, and one day short course in Buenos Aires, covering Digital Libraries, Semantic Web, Open Source. Argentina, May 17-19, 2004
33. Overview of Digital Libraries. Half day tutorial for ICADL 2003, Dec. 2003, Kuala Lumpur, Malaysia
34. Overview of Digital Libraries. Full day tutorial for JCDL'2003, May 27, Houston, TX
35. Introduction to ETDs and NDLTD. Tutorial for ETD'2003 (The 6th Int'l Symp. on Electronic Theses and Dissertations), Humboldt U., Berlin, GE, May 24, 2003
36. Practical Digital Libraries Overview. Two-hour tutorial at ICADL'2002, Singapore, Dec. 17-20, 2002, http://www.cais.ntu.edu.sg:8000/icadl02
37. Overview of Digital Libraries. Half-day tutorial at JCDL'2002, Portland, July 14, 2002 (see p. 417 in Proc. JCDL 2002).
38. Digital Libraries: from requirements to theory to systems to projects. 2 hour tutorial. In Tutorials/Panels, Digital Library: IT Opportunities and Challenges in the New Millennium, pp. 233-363, July 9, 2002, Beijing, China, http://researchsmp2.cc.vt.edu/~fox/tmp/FoxDLOC2002.ppt
39. Practical Digital Libraries Overview. Two hour tutorial at ICADL'2001, December 10, 2001 (9-11am), Bangalore, India, http://ei.cs.vt.edu/~dlib/tut/ICADL01.htm
40. Construyendo Bibliotecas Digitales Interoperables: Una Guia Practica para crear Archivos Abiertos (in English this is "Building Interoperable Digital Libraries: A Practical Guide to Creating Open Archives"). 1.5 hour tutorial (presented in Spanish, prepared with the assistance of Ryan Richardson and Hussein Suleman), as part on UNESCO- sponsored training course on electronic theses and dissertations, Montevideo, Uruguay, Nov. 12-14, 2001, hosted by UNESCO, ISTEC, ...
41. Practical Digital Libraries Overview. Half-day tutorial, Pre-conference for LITA National Forum, Milwaukee, WI, 10/12/2001
42. Practical Digital Libraries Overview. Half-day tutorial, scheduled for ACM SIGIR'2001, New Orleans, LA, Sept. 7-12, 2001
43. Overview of Digital Libraries. Half-day tutorial, for 5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'2001, September 4-8 2001, Darmstadt, Germany, http://www.ecdl2001.org
44. DL Interoperability via Metadata Harvesting - Applying the Open Archives Initiative Protocol. Half-day tutorial by Carl Lagoze and Edward A. Fox, scheduled for 5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL'2001, September 4-8 2001, Darmstadt, Germany, http://www.ecdl2001.org
45. Practical Digital Libraries Overview. Full-day tutorial, JCDL'2001, Roanoke, VA, June 24, 2001
46. Digital Libraries from Theory to Framework to Application: 5S Model, Open Archives Initiative, and Networked Digital Library of Theses and Dissertations, Dec. 6, 2000, Seoul, S. Korea, half-day tutorial for 2000 Asian DL conference, ICADL'2000 http://www.ndltd.org/talks/foxtutkr.ppt
47. Practical Digital Libraries Overview. Half-day tutorial, ACM Multimedia 2000, Los Angeles, CA, Oct. 30 - Nov. 4, 2000, http://ei.cs.vt.edu/~dlib/tut/MM00.htm
48. Digital Libraries. One-and-a-half day (12 hour) tutorial for DISSAnet workshop, Pretoria, South Africa, Oct. 23-24, 2000, http://ei.cs.vt.edu/~dlib/tut/SA.htm
49. Practical Application of Digital Libraries. Full-day tutorial, RRTC Advanced Technology Seminar, Rosslyn, VA, October 13, 2000, http://ei.cs.vt.edu/~dlib/tut/RRTC4.htm
50. Digital Libraries: An Overview. Full-day tutorial, with ECDL'2000, European Conference on Digital Libraries, Lisbon, Portugal, Sept. 18, 2000, http://ei.cs.vt.edu/~dlib/tut/ECDL2000.htm
51. Digital Libraries: An Overview. Two half-day tutorials (intro., advanced), ACM Digital Libraries '2000, June 4-7, 2000, http://www.ndltd.org/talks/DL2000i.ppt, http://ei.cs.vt.edu/~dlib/tut/indexi.htm, http://ei.cs.vt.edu/~dlib/tut/indexa.htm
52. Principios basicos para desarrollar bibliotecas digitales (Fundamental principles to develop digital libraries). Half-day workshop tutorial for Primer Seminario-Taller Subregional Sobre Bibliotecas Digitales, Ciudad Universitaria Rodrigo Facio, San Jose, Costa Rica, Dec. 8-10, 1999.

Sponsored by Universidad de Costa Rica, the Iberoamerican Science and Technology Education Consortium (ISTEC), Executive Secretariat for Integral Development, the Organization of American States (OAS), and the American Embassy in Costa Rica (http://www.istec.org/jerome/costarica/engsemucr.html)

53. Practical Digital Libraries Overview Part 2. Half-day tutorial for ACM Digital Libraries '99, Berkeley, CA, August 11-14, 1999.
54. Digital Libraries. Full-day tutorial for CoLIS 3, Dubrovnik, May 23, 1999.
55. Digital Libraries. Full-day tutorial for ACM Multimedia'98, Bristol, UK, Sept. 12-16, 1998.
56. Digital Libraries. Full-day tutorial for SICON'98, Singapore, July 1, 1998.
57. Fundamentals of Digital Libraries. Half-day tutorial (with Rob Akscyn), ACM Digital Libraries '98, June 23, 1998, Pittsburgh, PA.
58. Multimedia Information and Systems. Half-day tutorial selected by ACM SIGIR'97, July 27-31, 1997, Philadelphia.
59. Building and Applying Digital Libraries. Full-day tutorial for ICMCS'97, IEEE International Conference on Multimedia Computing and Systems, June 3, 1997, Ottawa, Canada.
60. Digital Libraries for Computer Science Education. Half-day workshop (tutorial) presentation for ACM SIGCSE '97, San Jose, 3/1/97.
61. Building and Applying Digital Libraries: Part 1, Introduction and Part 2, Research. Two half-day tutorials for ACM Multimedia'96, Boston, MA, Nov. 18, 1996, with Robert M. Akscyn.
62. Building and Applying Digital Libraries. Half-day tutorial for SIGIR'96, Zurich, Switzerland, Aug. 18, 1996, by E. Fox.
63. Information Retrieval and Hypertext. Half-day tutorial for ACM DL'96, Bethesda, MD, March 20, 1996, by E. Fox and R. Akscyn.
64. Digital Libraries. Half-day tutorial for CIKM'95, Baltimore, MD, Nov. 28, 1995, by E. Fox.
65. Research Issues and Design Issues for Digital Libraries. Half-day tutorial for ACM Multimedia'95, San Francisco, Nov. 6, 1995, by E. Fox and R. Akscyn.
66. Background for Digital Libraries: Information Retrieval and Hypertext. Half-day tutorial for ACM Multimedia'95, San Francisco, Nov. 6, 1995, by E. Fox and R. Akscyn.
67. Design and Use of Digital Libraries. Half day tutorial for ACM SIGIR '94, International Conference on R&D in Information Retrieval, July 3-6, 1994, Dublin, Ireland, by R. Akscyn and E. Fox.
68. Introduction to Information Retrieval. Afternoon tutorial for ACM Workshop on Information Retrieval and Genomics, sponsored by ACM SIGIR and SIGBIO, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD, May 2- 4, 1994
69. Multimedia Technology. Afternoon tutorial for 1994 Custom Integrated Circuits Conf. Educational Session, May 1, 1994, San Diego.
70. Introduction to Information Retrieval. Half day tutorial for CIKM'93, 2nd International Conference on Information and Knowledge Management, Nov. 1-5, 1993, Arlington, VA.
71. Multimedia Information Systems, Knowledge Based Information Retrieval, Electronic Libraries and other presentations (total: 15 hours of lectures). Invited visit to People's Republic of China, Nov. 20-29, 1992, Beijing, sponsored by Beijing Information Technology Inst., plus 1/2 day visit (including presentation on Digital Video Technology) to Tsinghua Univ.
72. Multimedia Information Systems. Half day tutorial for CIKM-92, First International Conference on Information and Knowledge Management, Nov. 8-11, 1992, Baltimore, MD.
73. Introduction to Multimedia. Half day tutorial for SCAMC (16th Annual Symposium on Computer Applications in Medical Care), Nov. 8-11,1992, Baltimore.
74. Multimedia Information Systems. Half day tutorial for Philips Laboratories, Oct. 23, 1992, Briarcliff, NY.
75. Multimedia: Applications and Practice. One day course for Eurographics '92, 7-11 Sept. 1992, Cambridge, England.
76. Knowledge-Based Information Retrieval. Half day tutorial for SIGIR '92, 15th Intern'l Conf. on R & D in Information Retrieval, June 21-24, 1992, Copenhagen, Denmark.
77. CD-ROM and Multimedia Technologies. Four day UCLA Extension course with E. Fox doing 1 day session on multimedia, Jan. 21-24, 1991, Los Angeles, CA.

78. Multimedia Information Systems. Half day tutorial for SIGIR '91, 14th Int'l Conf. on R&D in Information Retrieval, Oct. 13-16, 1991, Chicago, IL.
79. Interactive Technologies. Three day short course, conference presentation, sponsored by Sistema Nacional de Informacao Geografica, June 11-14, 1991, Lisbon, Portugal.
80. Multimedia/Hypermedia Computing and Applications. Half-day educational session for International Conference on Consumer Electronics (ICCE, 10th Anniversary), June 4, 1991, Chicago, IL.
81. Knowledge Based Information Retrieval. Half day tutorial for SIGIR '90, 13th Int'l Conf. on R&D in Information Retrieval, Sept. 5-7, 1990, Brussels, Belgium.
82. Interactive Digital Video Technologies. Three day short course, sponsored by Sistema Nacional de Informacao Geografica, July 16-18, 1990, Lisbon, Portugal.
83. Knowledge Based Techniques. Half day course for European Summer School in Information Retrieval, organized by A.I.C.A. Special Interest Group in Information Retrieval, Sezione A.I.C.A. delle Tre Venezie, 9-12 July, 1990, Bressanone, Italy.
84. CD-ROM Publishing and Access. Full day Professional Development Seminar for the Washington, D.C. Chapter of the ACM, Nov. 9, 1989, U. Maryland, College Park, MD.
85. Knowledge Based Information Retrieval. Half day tutorial for SIGIR '89, 12th Int'l Conf. on R&D in Information Retrieval, June 25-28,1989, Cambridge MA.
86. CD-ROM Interfaces -- From Authoring to Access. Half day workshop for American Society for Information Sci. Mid-Year Meeting, May 21-24, 1989, San Diego, CA.
87. CD-ROM Interfaces: from authoring to access. Half day tutorial for CHI '89, ACM Conf. on Human Factors in Computing Systems, April 30-May 4,1989, Austin TX.
88. CD-ROM Publishing and Access. Half day tutorial, demonstration of the Virginia Disc Series of CD-ROMs, for ACM Conf. on Document Processing Systems, Dec. 5-9, 1988, Santa Fe, NM.

DEMONSTRATIONS:

1. Edward A. Fox, Liuqing Li. (2020). Twirole - A User Classification Tool for Supporting User-centered Analysis on Twitter. Demonstration, with abstract in Proceedings International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020), May 2020, page 1169, to be presented in May 2021 at ISCRAM 2021. Blacksburg, VA.
2. Sunshin Lee, Mohamed Magdy, Edward A. Fox. IDEAL: Integrated Digital Event Archive & Library. Demonstration, Center for Human Computer Interaction (CHCI) 20-Year Celebration Conference, 15-17 October 2015, Blacksburg, VA, USA.
3. Edward A. Fox and the IDEAL team. IDEAL (Integrated Digital Event Archive & Library). Virginia Science Festival, Sept. 26, 2015, Blacksburg, VA, USA. http://www.cpe.vt.edu/sciencefestival/plan.html
4. Rizmari Versfeld, Spencer Lee, Edward Fox, Hussein Suleman and Kyle Williams. Digital Library in a 3D Virtual World: The Digital Bleek and Lloyd Collection in Second Life. Refereed demo in Research and Advanced Technology for Digital Libraries, Proc. 14th European Conference, ECDL2010, Glasgow, Sept. 6-10, 550-553
5. Spencer Lee, Bradley Willis, Joseph S. Bourne Jr., Edward A. Fox. Entertainment History Museums in Virtual worlds: Video Game and Music Preservation in Second Life, Demonstration, JCDL 2010, June 21-25, Gold Coast, Australia, 403-404
6. B. Stephen Carpenter II, Richard Furuta, Frank Shipman, Allison Huie, Daniel Pogue, Edward A. Fox, Spencer J. Lee, Peter Brusilovsky, Lillian N. Cassel, and Lois M.L. Delcambre, Multiple Sources with Multiple Portals: A Demonstration of The Ensemble Computing Portal in Second Life, Demonstration, JCDL 2010, June 21-25, Gold Coast, Australia, 397-398

7. Gregory W. Hislop, Lillian N. Cassel, Richard Furuta, Lois M. L. Delcambre, Peter Brusilovsky, Edward A. Fox, Daniel D. Garcia. Ensemble - the online community center for computing educators: demonstration. June 2010. Consortium for Computing Sciences in Colleges. Journal of Computing Sciences in Colleges, 25(6): 74-75
8. Spencer Lee, Edward Fox, Gary Marchionini, Javier Velasco, Goncalo Antunes and Jose Borbinha. Virtual DL Poster Sessions in Second Life. Refereed demo at JCDL 2009
9. Spencer Lee, Seungwon Yang, Javier Velasco, Edward Fox. Second Life Demo for U. Texas Austin, April 3, 2009
10. Spencer Lee, Seungwon Yang, Larry Taylor, Edward Fox. Second Life Demo for VT CSRC, Feb. 17, 2009
11. Murthy, U., Torres, R., Fox, E.A., Yang, S., Venkatachalam, L., and Goncalves, M. From Concepts to Implementation and Visualization: Tools from a Team-Based Approach to IR. Demo D7 at SIGIR 2008 (Singapore, July 20-24, 2008). ACM, New York, NY, p. 889. DOI= http://doi.acm.org/10.1145/1390334.1390563
12. Douglas Gorton, Weiguo Fan and Edward A. Fox: Specification and Generation of Digital Libraries into DSpace Using the 5S Framework. Research and Advanced Technology for Digital Libraries, ECDL 2007: pp. 567-569, Springer Berlin / Heidelberg, LNCS 4675, demo, http://dx.doi.org/10.1007/978-3-540-74851-9_71
13. Barbara Lagoeiro Moreira, Marcos Andre Goncalves, Alberto Henrique Frade Laender, Edward A. Fox. 5SQual: a quality assessment tool for digital libraries. ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007, Vancouver, British Columbia, Canada, demonstration, p. 513, http://doi.acm.org/10.1145/1255175.1255313
14. Naga Srinivas Vemuri, Ricardo da S. Torres, Rao Shen, Weiguo Fan, and Edward A. Fox. A Content-Based Image Retrieval Service for Archaeology Collections, demonstration for ECDL 2006, Alicante, Spain, Sept. 17-21, 2006, Research and Advanced Technology for Digital Libraries, ISBN 978-3-540-44636-1, Lecture Notes in Computer Science, Volume 4172, ISSN 0302-9743, Springer Berlin / Heidelberg, 2006, http://dx.doi.org/10.1007/11863878_37, pp. 438-440
15. Johnny L. Sam-Rajkumar, Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, and Edward A. Fox. EtanaCMV: A Visual Browsing Interface for ETANA-DL Based on Coordinated Multiple Views, demonstration for ECDL 2006, Alicante, Spain, Sept. 17-21, 2006, Research and Advanced Technology for Digital Libraries, ISBN 978-3-540-44636-1, Lecture Notes in Computer Science, Volume 4172, ISSN 0302-9743, Springer Berlin / Heidelberg, 2006, http://dx.doi.org/10.1007/11863878_51, pp. 492-494
16. Uma Murthy, Ricardo da S. Torres and Edward A. Fox. SIERRA - A Superimposed Application for Enhanced Image Description and Retrieval, demonstration for ECDL 2006, Alicante, Spain, Sept. 17-21, 2006, Research and Advanced Technology for Digital Libraries, ISBN 978-3-540-44636-1, Lecture Notes in Computer Science, Volume 4172, ISSN 0302-9743, Springer Berlin / Heidelberg, 2006, http://dx.doi.org/10.1007/11863878_63, pp. 540-543
17. Seungwon Yang, Ben Congleton, George Luc, Manuel A. Perez-Quinones, and Edward A. Fox. Demonstrating the Use of SenseCam in Two Domains. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, demonstration, p. 376, http://doi.acm.org/10.1145/1141753.1141872
18. Uma Murthy, Kapil Ahuja, Sudarshan Murthy, and Edward A. Fox. SIMPEL: A Superimposed Multimedia Presentation Editor and Player. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, demonstration, p. 377, http://doi.acm.org/10.1145/1141753.1141873
19. Douglas Gorton, Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, and Edward A. Fox. ETANA-GIS: GIS for Archaeological Digital Libraries. In Proc. JCDL 2006, June 11-15, 2006, Chapel Hill, NC, demonstration, p. 379, http://doi.acm.org/10.1145/1141753.1141875
20. Ryan Richardson and Edward A. Fox. Using Concept Maps as a Cross-Language Resource Discovery Tool. In Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries (JCDL2005), June 7-11, 2005, Denver, p. 415
21. Ananth Raghavan, Divya Rangarajan, Rao Shen, Marcos Andre Goncalves, Naga Srinivas Vemuri, Weiguo Fan, and Edward A. Fox. Schema Mapper: A Visualization Tool for DL Integration. In Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries (JCDL2005), June 7-11, 2005, Denver, p. 414
22. U. Ravindranathan, R. Shen, M. A. Goncalves, W. Fan, E. A. Fox, and J.W. Flanagan. ETANA-DL: Managing Complex Information Applications - An Archaeology Digital Library. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach

and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, p. 414
23. R. da S. Torres, C. B. Medeiros, M. A. Goncalves, and E. A. Fox. An OAI Compliant Content-Based Image Search Component. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, p. 418
24. Qinwei Zhu, M.A. Goncalves, E.A. Fox. 5SGraph Demo: A Graphical Modeling Tool for Digital Libraries. JCDL'2003, May 27-31, 2003, Houston, TX, p. 385
25. P. Calado, M. Goncalves, E. Fox, B. Ribeiro-Neto, A. Laender, A. da Silva, D. Reis, P. Roberto, M. Vieira, J. Lage. The Web-DL Environment for Building Digital Libraries from the Web. JCDL'2003, May 27-31, 2003, Houston, TX
26. CITIDEL, by Edward A. Fox and Lillian Cassel. Poster and demonstration, at PI meeting of the National SMETE (science, mathematics, engineering, and technology education) Digital Library initiative, Washington, D.C., Dec. 2-3, 2001
27. Building Interoperable Digital Library Services: MARIAN, Open Archives, and NDLTD, by Fox, E. A., France, R. K., Goncalves, M. A., and Suleman, H. In Proceedings of the 24th Annual International SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA, Sept. 9-14, 2001
28. Computer Science Teaching Center (CSTC, www.cstc.org), IDM'2001, Dallas, TX, April 30, 2001
29. Building a Unified Library for Theses and Dissertations, by Fox, E. A., France, R. K., Goncalves, M. A., Suleman, H., and Lee, M. H. In Proceedings of the 3rd International Conference of Asian Digital Library, Seoul, December 6-8, 2000
30. XXDL and CSTC and Virginia Tech, demonstration for NSF-sponsored NSDL workshop, September 22-24, 2000, NSF, Arlington, VA, http://www.ndltd.org/talks/nsdlcstc.ppt
31. Computer Science Teaching Center, demonstration at NSF IDM'2000, Chicago, March 5-7, 2000, http://boston.cwru.edu/idm00/
32. NDLTD and CSTC/CRIM, demonstrations for ACM DL'99, with Neill Kipp, Aug. 12, 1999
33. Networked Digital Library of Theses and Dissertations. Demonstration for CoLIS 3, Dubrovnik, May 25, 1999.
34. Improving Post-Secondary Education through the Networked Digital Library of Theses and Dissertations. Poster and demonstration for FIPSE PI Meeting, Washington, D.C., Oct. 23-25, 1998
35. Networked Digital Library of Theses and Dissertations. Demonstration for ACM SIGIR'97, July 28, 1997, Philadelphia.
36. Visualizing Search Results with Envision. Demonstration by Lucy T. Nowell, Robert K. France, and Edward A. Fox for ACM SIGIR'96, Zurich, Switzerland, Aug. 19, 1996.
37. Automatic Building of Hypertext Links in Digital Libraries. Demonstration for ACM SIGIR'95, Seattle, WA, July 10, 1995, by Robert B. Kellogg, Madhan Subhas and Edward A. Fox.
38. Envision: Information Visualization in a Digital Library. Demonstration for ACM SIGIR'95, Seattle, WA, July 10, 1995, by Lucy T. Nowell and Edward A. Fox.
39. NSF Education Infrastructure Project. Hour demonstration for Multimedia Expo, Virginia Tech, March 24, 1995.
40. MARIAN and Envision System Demonstrations, for 16th Intern'l Conf. on R & D in Information Retrieval, SIGIR '93, Pittsburgh, PA, June 27 - July 1, 1993.
41. MARIAN System Demonstration, at Software Fair, ACH-ALLC93, Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, June 16-19, 1993, Georgetown Univ., Washington, D.C.

PATENTS, SOFTWARE AND DATABASE PRODUCTS:

1. Co-inventor on: US Provisional Patent Application 63/039,725; filed June 16, 2020; Methods and Systems for Generating Summaries Given Documents with Questions and Answers (Edward A. Fox, Saurabh Chakravarty, Maanav Mehrotra, Satvik Chekuri, and Aarohi Sumant)

2. Co-inventor on: US Provisional Patent Application 62/945,202; filed December 8, 2019; Methods and Systems for Generating Declarative Statements Given Documents with Questions and Answers (Edward A. Fox, Saurabh Chakravarty, Raja Venkata Satya Phanindra Chava, and Maanav Mehrotra)
3. Liuqing Li, Ziqian Song, Xuan Zhang, and Edward A. Fox. TwiRole: A Hybrid Model for Role-related User Classification on Twitter. Code Ocean reproducible software capsule, Nov. 2019, https://codeocean.com/capsule/9584745/tree/v4
4. Co-inventor on US patent no. 7346621, filed May 14, 2004, issued March 18, 2008, Method and System for Ranking Objects Based on Intra-type and Inter-type Relationships (Benyu Zhang, Hua-Jun-Zeng, Wei-Ying Ma, Wensi Xi, Zheng Chen, and Edward A. Fox)
5. SMART: Proposed, designed, and led implementation effort of a new version of the SMART system, 1980-2, written in the C language under UNIX. From 1986, extended this for use at Virginia Tech. The version of SMART further developed at Cornell by others since 1983 has been used around the world for information retrieval research. The databases called "CACM" and "ISI" developed 1980-2 for dissertation research with SMART have been used in scores of published studies, as benchmark test sets.
6. CED lexicon: Proposed and supervised preparation of a database form of the "Collins English Dictionary" that has been distributed from the Oxford Text Archive since 1988.
7. CODER: Proposed and supervised development of the COmposite Document Expert/extended/effective Retrieval System, 1985-92, which was used as a testbed for the study of artificial intelligence concepts in the field of information retrieval, and applied to electronic mail digests, navy messages, and literature (and thesaurus data) on cardiology.
8. LEND: Proposed and supervised development of the Large External object-oriented Network Database system, an object-oriented database management system, representing content as a graph/network, which was distributed starting in 1992 by Virginia Tech Intellectual Properties, and was used with both the CODER and MARIAN systems.
9. MARIAN: Proposed and supervised development starting in 1991 of the MARIAN system, an alternative to searching on the VTLS system for library catalog data. The campus production service version of this ran on a collection of PC and NeXTstep machines in the Computing Center and was designed to support scores of simultaneous users. It was built upon data and code from the CED lexicon and CODER (both developed here in the 1980s). With support from NLM, and new support from NSF, it was revamped in Java to become a key part of our research on digital libraries. URL: http://www.dlib.vt.edu/products/marian.html
10. ENVISION: Proposed and supervised development 1991-93, 1995 of Project Envision, developing a large hypermedia database (digital library) from the ACM publications, along with a retrieval system and graphical user interfaces to support access to the computer science literature.
11. TULIP: Technical coordinator for Virginia Tech's efforts to make about 40 journals in material science and engineering from Elsevier available for campus use, working with OCLC software
12. EI: Proposed and supervised development of Computer Science courseware and related digital library content, 1993-98, which has led to most CS courses at Virginia Tech having online materials, and some functioning essentially paperless, with up to 150K accesses/wk from local and world-wide users. URL: http://ei.cs.vt.edu
13. QUIZIT: Supervised development of an automated WWW-based quiz system to use with CS4624 and CS5604 as well as other courses like CS1604. This has led to several publications and is available for use. See the online papers and software under http://ei.cs.vt.edu
14. Boy Scout Computers Merit Badge web site used with local group
15. ETD: Supervised development of electronic thesis and dissertation related software and systems in connection with building the Networked Digital Library of Theses and Dissertations (NDLTD). This includes extensive online information at http://www.ndltd.org, and earlier info at http://etd.vt.edu, with documentation, automated submission software, multimedia tutorials, SGML/XML document type definitions, conversion tools (to SGML/XML, from SGML/XML to HTML), services (searching, browsing, submission, and conversion from PostScript to PDF), and adaptation of Open Archives Initiative. Collaboration on these efforts includes staff on multiple grants, personnel in Graduate School, Library, Computing Center, and other universities. The Library now maintains the http://etd.vt.edu site, while I am responsible for domains ndltd.org and theses.org (also known as dissertations.org) and related listservs. As Executive Director of NDLTD, I coordinate efforts of its members; in

March 2000 roughly 225 people attended our 3rd International Symposium, and similar numbers have continued in subsequent years. According to Don Fischer, Program Officer at FIPSE (US Dept. of Education), in email dated 2/14/2000 "Your work is outstanding. Your outreach efforts are incredibly extensive."

16. NRG: The Web Characterization Repository developed for the WWW Consortium (W3C), helped catalog papers, tools, and trace files related to Web traffic characterization. It was used by researchers and practitioners worldwide.
17. CSTC: The Computer Science Teaching Center (www.cstc.org) was hosted at Virginia Tech where I served as an editor and was responsible for the software, server, and repository. This supported teachers and learners interested in computing. This led to CITIDEL - see below.
18. Open Archives Initiative: This international digital library initiative involves specifications, protocols, and other standards (www.openarchives.org). I was one of the founding members and served on the Steering Committee. Virginia Tech students developed a range of widely used software to support development, testing, and operation. I proposed and ran 2 workshops for OAI in 2000 and one in 2001, attended the public launch meeting 1/23/2001, and had several grants related that yielded new tools.
19. CITIDEL: Computing and Information Technology Interactive Digital Educational Library, www.citidel.org, collection effort that is part of NSDL (www.nsdl.org), supporting English and Spanish access to resources to help with teaching and learning about computing and IT, for various ages and types of users, incorporating CSTC, ACM JERIC, NCSTRL, ResearchIndex, and other content. Collaborative project funded by NSF for 2001-2003, with College of NJ, Hofstra, Penn State, and Villanova. This led to Ensemble - see below.
20. Crawls: In collaboration with Charlie Clarke at U. of Waterloo, crawls of large parts of WWW to allow research with large text collections. One collection of 1 Tbyte was collected in 2001. Another collection, focused on government content, was collected and was adapted to yield a new TREC test collection for 2002.
21. PlanetMath and PlanetPhysics: www.planetmath.org and www.planetphysics.org are online community-based encyclopedia initiatives, covering mathematics (and a little computer science) and physics. Software used was developed by Aaron Krowne, whose MS thesis on this I supervised. Mr. Krowne continues to run the sites, which Virginia Tech hosted under my supervision.
22. ETANA: http://www.etana.org/; developed at Virginia Tech, then ported to Vanderbilt Library that agreed to sustain this effort to provide coordinated access to archaeological information.
23. Digital library curricular guidelines and modules available on our web site (http://curric.dlib.vt.edu/), through Wikiversity (http://en.wikiversity.org/wiki/Curriculum_on_Digital_Libraries), and as a collection in Ensemble (see on http://www.computingportal.org/collections) that also is available through the National Science Digital Library (NSDL)
24. Ensemble Portal: http://www.computingportal.org, with Virginia Tech leading work on the distributed portal, with PI Boots Cassel managing the overall Ensemble effort involving 6 universities as well as other partners - the Computing Portal in NSF's NSDL (National Science Digital Library). This portal to educational resources in computing is being expanded in 2016-2017 through machine learning methods to include entries from YouTube and SlideShare.
25. Crisis, Tragedy, and Recovery Network (CTRnet): http://www.ctrnet.net; developed at Virginia Tech with NSF funding, and partnering with the Internet Archive; digital library and archiving support related to communities, supporters, and scholars working with disasters and emergencies
26. IDEAL (Integrated Digital Event Archiving and Library) and GETAR (Global Event and Trend Archive Research): http://eventsarchive.org. Making use of a Hadoop cluster with more than 20 nodes and more than 150 terabytes of storage, these NSF funded projects support an advanced information retrieval system integrating search engine and Web archiving technologies, including collecting tweets, focused crawling for webpages, classification, topic analysis, clustering, entity recognition, geo-location, indexing, searching, and multiple user interfaces.
27. Social Interactome, an NIH-funded project, running clinical trials on interventions employing social networks to aid with recovery from addictions, uses a series of virtual machines in Computer Science, as well as other services at VTCRI (managed by PI Warren Bickel), involving an enhanced version of Friendica and related interface and analysis software.

DIGITAL LIBRARY RESEARCH LABORATORY (DLRL, www.dlib.vt.edu) (earlier: INFORMATION ACCESS LABORATORY):

- Fox was director of the NSF supported Information Access Laboratory in Computer Science. His offices at the Computing Center were moved to 840 Pointe West Commons (PWC), Suite 8, allowing merger with the IAL, and the new complex is now the DLRL, located at 2030 Torgersen Hall. We are responsible for domains including: dissertations.org, theses.org, ndltd.org, and dlib.vt.edu.
- Facilities dating back to the early 1990s have included (in addition to upgrades/replacements/additions):
  - VT-PetaPlex-1, with 2.5 terabytes of disk storage, over 100 processors, developed by Knowledge Systems Incorporated, a very early cluster focused on storage
  - Roughly $770K of hardware donated by IBM for digital library research, located in Computing Center, PWC, or engaged in logging in locations around campus
  - A wide variety of PC-based workstations purchased 2001-2002
  - Older servers of various types, including DEC Alpha, Sun Ultra, Gateway, Mac, IBM
  - Multimedia peripherals: camcorders and cameras, VTRs (digital, Betacam, Hi-8, SVHS, U-matic), audio units (CD-DA, DAT, cassette), laserdisk player, preamp, amplifier, mixer, CD-ROM drives, professional microphones, headphones, speakers, monitors - many donated to VT's New Media Center in return for access by classes and research groups
  - CD-ROM production (Kodak PCD Writer 225) for DOS/Mac/UNIX
- Current facilities include servers (in the Computing Center and in 2070 Torgersen Hall), and virtual machines in Computer Science, for the IDEAL/GETAR projects, as well as a full lab of units at DLRL.

ORAL PRESENTATIONS:

1. Edward A. Fox. Reiki at Virginia Tech. Invited (by Professor Young Ju) presentations in two sections of HFNE 2334, Feb. 18 and 19, 2019, Virginia Tech, Blacksburg, VA, http://fox.cs.vt.edu/talks/2019/20190218ReikiSlidesHFNE2334IntegrativeHealth.ppt
2. Edward A. Fox. Reiki Overview, Invited presentation at Carilion New River Valley Medical Center, Radford, VA, 17 Dec. 2019, http://fox.cs.vt.edu/talks/2019/20191217ReikiSlidesRadford.ppt
3. Edward A. Fox. Dialog Act Classification for Question Answer Corpora. Presentation for Intro to Scieneering, 3100 Torgersen Hall, Virginia Tech, Blacksburg, VA 24061; 6 Dec. 2019 (based on work with Saurabh Chakravarty and Raja Venkata Satya Phanindra), http://fox.cs.vt.edu/talks/2019/20191206ScieneeringFoxASAIL.pptx
4. Edward A. Fox. Information Friendly. Presentation for Engineering Research Seminar, ENGR 1014, Hancock 100, Virginia Tech, 15 November 2019, http://fox.cs.vt.edu/talks/2019/20191115ENGE1014Fox.pptx
5. Edward A. Fox. Information Friendly. Presentation for Hypatia Seminar, Hancock 209, Virginia Tech, 11 November 2019, http://fox.cs.vt.edu/talks/2019/20191111HypatiaFox.pdf
6. Edward A. Fox. Conference welcome and closing, at ETD 2019, the 22nd International Symposium on Electronic Theses and Dissertations, Porto, Portugal, Nov. 6-8. http://fox.cs.vt.edu/talks/2019/20191106FoxETD2019welcome.pptx
7. Edward A. Fox. Reiki at Virginia Tech. Invited (by Professor Young Ju) presentations in two sections of HFNE 2334, Feb. 18 and 19, 2019, Virginia Tech, Blacksburg, VA, http://fox.cs.vt.edu/talks/2019/20190218ReikiSlidesHFNE2334IntegrativeHealth.ppt
8. Edward A. Fox. Integrating Research and Education Regarding Global Event and Trend Archiving. Computer Science Colloquium, Old Dominion University, Norfolk, VA, 18 January 2019, http://fox.cs.vt.edu/talks/2019/20190118ODUseminarFox.pptx

9. Edward A. Fox. Information Research. ENGR 1014: Engineering Research Seminar, 5 October 2018, Virginia Tech, http://fox.cs.vt.edu/talks/2018/20181005ENGR1014Fox.pptx
10. Edward A. Fox. Conference welcome and closing, at ETD 2018, the 21st International Symposium on Electronic Theses and Dissertations, Taiwan, Sept. 26-28. http://fox.cs.vt.edu/talks/2018/20180926FoxETD2018welcome.pptx
11. Edward A. Fox. Tips for maximizing your ETD program. Workshop presentation at ETD 2018, the 21st International Symposium on Electronic Theses and Dissertations, Taiwan, Sept. 26-28. http://fox.cs.vt.edu/talks/2018/20180926FoxETD2018workshop.pptx
12. Edward A. Fox. Information - Friendly Hypatia Seminar, 18 September 2018, Virginia Tech, http://fox.cs.vt.edu/talks/2018/20180918HypatiaFox.pptx
13. Edward A. Fox. Information Research Galileo Seminar, 13 September 2018, Virginia Tech, http://fox.cs.vt.edu/talks/2018/20180913GalileoFox.pptx
14. Edward A. Fox. Big Data and Machine Learning in Virginia Tech's Digital Library Research Laboratory. STEP Faculty Seminar, 6 July 2018, Virginia Tech, http://fox.cs.vt.edu/talks/2018/20180706FoxSTEP.pptx
15. Edward A. Fox. Big Data and Machine Learning in Virginia Tech's Digital Library Research Laboratory. Invited presentation at Yahoo! Research, Sunnyvale, CA, 21 June 2018, http://fox.cs.vt.edu/talks/2018/20180621FoxYahoo.pptx
16. Edward A. Fox and Zhiwu Xie. Web Archiving and Digital Library Projects and Technologies. Presentation for Linux/Unix Users Group @ VT (VTLUUG). 15 March 2018. http://fox.cs.vt.edu/talks/2018/20180315WADL-VTLUUG.pdf
17. Fox, Edward A. Information Research. STEP (Student Transition Engineering Program) Faculty Research Seminar, College of Engineering, Virginia Tech, 7 July 2017, http://fox.cs.vt.edu/talks/2017/20170707STEPfacultySeminar.pptx
18. Fox, Edward A. Digital Library Framework for Behavioral Science: Social Interactome (based on research with Prashant Chandrasekar, Allison Tegge, Christopher T. Franck, Warren K. Bickel, Sandesh Bhandari, Brian Brown, John Crawford, Kirstin Gathalian, Mikhail N. Koffarnus, Patsy A. Marshall, Lexie Mellis, Elan Perry, Derek Pope, Amanda J. Quisenberry, Sarah E. Snider, and Robert C. Reese). Introduction to Scieneering (ENGR/COS 2164) Seminar, CEED (Center for Enhancement of Engineering Diversity), College of Engineering, Virginia Tech, 15 September 2017
19. Fox, Edward A. Digital Library Framework for Behavioral Science: Social Interactome (based on research with Prashant Chandrasekar, Allison Tegge, Christopher T. Franck, Warren K. Bickel, Sandesh Bhandari, Brian Brown, John Crawford, Kirstin Gathalian, Mikhail N. Koffarnus, Patsy A. Marshall, Lexie Mellis, Elan Perry, Derek Pope, Amanda J. Quisenberry, Sarah E. Snider, and Robert C. Reese). First Year Galileo Seminar (ENGR 1054) Seminar, CEED (Center for Enhancement of Engineering Diversity), College of Engineering, Virginia Tech, 10 October 2017
20. Fox, Edward A. Information Friendly. Hypatia Seminar (ENGR 1034) Seminar, CEED (Center for Enhancement of Engineering Diversity), College of Engineering, Virginia Tech, 12 October 2017
21. Edward A. Fox. NDLTD Committee Meetings. Presentation at ETD 2017: 20th Int'l Symposium of the NDLTD, Washington, D.C., USA, August 7-9, 2017. http://fox.cs.vt.edu/talks/2017/20170808ETD2017committees.pptx
22. Edward A. Fox. ETD Metadata / Union Catalog / IR Consortium Networks. Presentation at ETD for Rookies Workshop at ETD 2017: 20th Int'l Symposium of the NDLTD, Washington, D.C., USA, August 7-9, 2017. http://fox.cs.vt.edu/talks/2017/20170807ETD2017FoxRookiesUnion.pptx
23. Edward A. Fox. Introduction to NDLTD and Brief History of the ETD Movement. Presentation at ETD for Rookies Workshop at ETD 2017: 20th Int'l Symposium of the NDLTD, Washington, D.C., USA, August 7-9, 2017. http://fox.cs.vt.edu/talks/2017/20170807ETD2017FoxRookiesIntro.pptx
24. Edward A. Fox. Web Archiving Research Supported by Text and Data Mining. Invited panelist for Text and Data Mining Discussion Forum, Virginia Tech, Newman Library, April 12, 2017. Video recording of the entire forum is available from http://hdl.handle.net/10919/77526

25. Edward A. Fox. "Launch, Persevere, and Collaborate." Invited panelist for Faculty Senate Forum on Research and Scholarship, 16 March 2017, The Inn, Virginia Tech, Blacksburg, VA
26. Edward A. Fox. GETAR Overview. Invited presentation for Discovery Analytics Center, Virginia Tech, Blacksburg, VA. 2 February 2017. http://fox.cs.vt.edu/talks/2017/20170202GETAR-DAC.pptx
27. Edward A. Fox. Overview for EFO Annual Meeting. 27 September 2016, The Inn, Virginia Tech, Blacksburg, VA. http://fox.cs.vt.edu/talks/2016/20160927FoxEFO.pptx
28. Edward A. Fox. Information Research. Presentation for ENGR 1014: Engineering Research Seminar. 2 September 2016, Virginia Tech, Blacksburg, VA. http://fox.cs.vt.edu/talks/2016/20160902ENGR1014Fox.pptx
29. Edward A. Fox. "Technologies in an Emerging Academic Field." Presentation for 27 July 2016 part of 2016 NEH Summer Institute for College and University Teachers: Veterans in Society: Ambiguities & Representations, 10-29 July 2016, Blacksburg, VA, http://www.veteransinsociety.org/, notes at http://fox.cs.vt.edu/talks/2016/20160727ViSlinks.txt
30. Edward A. Fox. Welcome (as NDLTD Executive Director) presentation for ETD 2016: 19th Int'l Symposium on ETDs. Lille, France. 11 July 2016. http://fox.cs.vt.edu/talks/2016/20160711ETD2016FoxWelcome.pptx
31. Edward A. Fox; Andrea Kavanaugh; Donald Shoemaker; Steven Sheetz; Mohamed Magdy Farag; and Sunshin Lee. "Can Collecting, Archiving, Analyzing, and Accessing Webpages and Tweets Enhance Resilience Research and Education?" Global Forum on Urban & Regional Resilience Spring 2016 Seminar Series, 2/11/2016, Virginia Tech, Blacksburg, VA
32. Edward A. Fox. From Man and Computer to Digital Libraries. Invited presentation for panel "HCI Research: Where are we coming from?". Plenary opening session for the CHCI 20-year Celebration, Oct. 16, 2015, in the Fife Theatre of the Moss Arts Center, Blacksburg, VA, USA.
33. Edward A. Fox. Quarter-day of presentations, including: Edward A. Fox: "ELISQ Seminar"; Tarek Kanan and Edward Fox: "Arabic Natural Language Processing: P-Stemmer, Browsing Taxonomy, Text Classification, RenA, ALDA, and Template Summaries - for Arabic News Articles"; Qatar University Library, Doha, Qatar, 19 May 2015, http://fox.cs.vt.edu/ELISQ/201505/
34. Edward A. Fox. Half-day of presentations, including: Edward A. Fox: "ELISQ Discussion with QNL Director Lux"; Edward A. Fox: "ELISQ Seminar"; Tarek Kanan and Edward Fox: "Arabic Natural Language Processing: P-Stemmer, Browsing Taxonomy, Text Classification, RenA, ALDA, and Template Summaries - for Arabic News Articles"; Qatar National Library, Doha, Qatar, 20 May 2015, http://fox.cs.vt.edu/ELISQ/201505/
35. Edward A. Fox. Half-day of presentations, including: Edward A. Fox: "Web Archives, IDEAL, and PBL Overview"; Richard Gruss and Edward A. Fox: Problem Based Learning To Build And Search Tweet And Web Archives"; Sunshin Lee and Edward A. Fox: "Big Data and Hadoop and DLRL: Introduction to the DLRL Hadoop Cluster"; Mohamed Farag and Edward A. Fox: Web Archive Content Analysis: Disaster Events Case Study". Qatar National Library, Doha, Qatar, 21 May 2015, http://fox.cs.vt.edu/ELISQ/201505/
36. At ETD 2014, University of Leicester, England, July 23-25: NDLTD Welcome, and panelist for "ETDs for life: opportunities, challenges, benefits"; also talks at the ETD for Rookies pre-conference workshop.
37. From CTRnet to IDEAL (and Qatar, VT, SiteStory, UPS, ...). Invited talk at NSF-funded workshop, Working with Internet Archives for Research (WIRE), Harvard University, Cambridge, MA, June 16, 2014, with online slides
38. Chair of panel, and panelist, on Educational Repositories for Teaching AI, with Lois Delcambre and Todd Neller. Fourth Symposium on Educational Advances in Artificial Intelligence (EAAI-13), July 15-16, 2013, Bellevue, WA
39. Many talks related to electronic theses and dissertations, including a number from years ago, including:
    - An ETD and 5S Related Perspective. Invited presentation in session on Scholarly Communication: New Models for Digital Scholarship Workflows at Coalition for Networked Information (CNI) Spring 2013 Membership Meeting, April 4-5, 2013, San Antonio, TX (with Stephen M. Griffin as session chair)
    - Co-chair of Webinar on Emergency Informatics and Digital Libraries, July 24, 2012 (as follow-up to NSF/CCC Workshop on Computing for

- E. Fox. MIT 6/18/96; Georgia Tech 10/4/96; U. Ill. Urbana-Champaign 10/29/96; Univ. Utah 11/8/96; MIT 11/20/96; Stanford U. 12/16/96; ... Ohio State U. 5/29/98, U. Pitt. 6/23/98, Singapore 6/30/98, U. Arizona 7/30/98, U. Southampton 9/11/98, ...
- E. Fox with other members of project team: NC State 7/9/96 and 10/9/96, ...
- Posters: Networked Digital Library of Theses and Dissertations, FIPSE PI Meeting, Nov. 1, 1997, Arlington, VA

40. Invited attendee and panelist (on 3 panels), NSF workshop on Interdisciplinary Computing, Washington, D.C., Nov. 3-5, 2011
41. Edward A. Fox, "Living In the KnowlEdge Society: the double duty of a librarian", invited speaker (by videoconference) at "Linking Research and Education in Digital Libraries" Workshop at TPDL 201, 28-29 Sept. 2011, Berlin
42. Edward A. Fox. Introduction to NDLTD and Brief History of the ETD Movement. Invited 30 minute plenary opening talk for day long ETDs for Rookies - Newcomer's Pre-Conference Tutorial. 14th International Symposium on Electronic Theses and Dissertations, ETD 2011, Cape Town, S. Africa, Sept. 13-17, 2011
43. Edward A. Fox. Other ETD Systems: ETD-db, the Short Story. Invited 40 minute plenary talk for day long ETDs for Rookies - Newcomer's Pre-Conference Tutorial. 14th International Symposium on Electronic Theses and Dissertations, ETD 2011, Cape Town, S. Africa, Sept. 13-17, 2011
44. Edward A. Fox. Welcome, invited 5 minute opening for 14th International Symposium on Electronic Theses and Dissertations, ETD 2011, Cape Town, S. Africa, Sept. 13-17, 2011
45. Edward A. Fox. Moderator and speaker at Panel on NDLTD Union Catalog. 14th International Symposium on Electronic Theses and Dissertations, ETD 2011, Cape Town, S. Africa, Sept. 13-17, 2011
46. Edward A. Fox. Some Reflections on Students' Roles. Invited presentation for Workshop on Data Curation. 14th International Symposium on Electronic Theses and Dissertations, ETD 2011, Cape Town, S. Africa, Sept. 13-17, 2011
47. Kavanaugh, A., Fox, E., Sheetz, S., Yang, S., Si, L.T., Shoemaker, D., Natsev, A., Xie, L. Social Media for Cities, Counties and Communities. Invited presentation, Inaugural conference on 'Communicating Disaster,' Bielefeld, Germany, January 12-15, 2011.
48. E. Fox. Invited presentation for panel "When Universities Put Dissertations on the Internet: New Practice; New Problem?", at American Historical Association 125th Annual Meeting, Jan. 6-9, 2011, Boston
49. M. S. Hsiao, A. L. Abbott, E. A. Fox, R. Murch, B. Budowle, N. J. Short, S. Misra, N. P. Kozievitch, and S. H. Park, "Toward a Quantitative Basis for Sufficiency of Friction Ridge Pattern Detail," Presentation at Impression and Pattern Evidence Symposium, Clearwater Beach, FL, Aug. 2010.
50. A. L. Abbott, M. S. Hsiao, E. A. Fox, R. Murch, B. Budowle, N. J. Short, S. Misra, N. P. Kozievitch, and S. H. Park, "Toward a Quantitative Basis for Sufficiency of Friction Ridge Pattern Detail," Presentation at 95th International Educational Conference, International Association for Identification, Spokane, WA, July 2010.
51. E. Fox. Invited presentation for panel on Innovation vs. Preservation: Does it Have to be One or the Other? Panelists Ed Fox, Geneva Henry; moderators Jill Kleister, Billie Peterson-Lugo; for Texas ETD Association Annual Meeting, a pre-conference workshop for ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
52. E. Fox. Basic Principles for ETD programs. Invited presentation for preconference workshop "Advice from NDLTD Leaders", ETD 2010 Conference, Austin, TX, June 16-18
53. E. Fox. Introduction to NDLTD and Brief History of the ETD Movement. Invited presentation for "ETDs for Rookies" - Newcomer's Workshop, at ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
54. E. Fox. ETD-db, the Short Story. Invited presentation for "ETDs for Rookies" - Newcomer's Workshop, at ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
55. E. Fox. NDLTD Welcome and Introduction. Invited presentation for Opening Session at ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010

56. E. Fox. Invited presentation for Plenary Panel: ETD Veterans. Panelists John Hagen (West Virginia University), Ed Fox (Virginia Tech), Thomas Dowling (OhioLink), Terry Kahn (UT Austin), Gail McMillan (Virginia Tech). At ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
57. E. Fox. NDLTD Wrapup. Invited presentation for Closing Session at ETD 2010 - 13th International Symposium on Electronic Theses and Dissertations. Austin, TX. June 16-18, 2010
58. A. L. Abbott, M. S. Hsiao, E. A. Fox, R. Murch, B. Budowle, N. J. Short, S. Misra, N. P. Kozievitch, and S. H. Park, "Development of a Quantitative Basis for Sufficiency in Friction Ridge Pattern Detail," Presentation for NIJ Panel on Impression Evidence, NIJ Conference 2010, National Institute of Justice, Arlington, VA, June 14, 2010.
59. E. Fox. NDLTD: Past, Present, and Future. Invited presentation for Canadian ETD and Open Repositories Workshop, May 10-11, 2010, Carleton University, Ottawa
60. Edward A. Fox. Mentoring in Computer Science. Panel presentation on How to be a Successful Mentor, 7th Annual Advancing Diversity at Virginia Tech Workshop, Jan. 11, 2010, Inn and Skelton Conference Center, Virginia Tech, http://fox.cs.vt.edu/talks/2010/20100111Mentoring.ppt
61. Edward A. Fox. CTR. Presentation (with same title as session since it was about our project) as part of panel for session "Crisis, Tragedy, and Recovery Network (CTRnet)" at the Coalition for Networked Information Fall Meeting, Dec. 15-16, 2009, Washington, D.C.
62. Edward A. Fox. Digital Libraries. Graduate seminar presentation for Virginia Tech Dept. of Computer Science, Nov. 20, 2009, Blacksburg, VA
63. Ensemble Project Overview. Presentation as panelist at NSDL Annual Meeting, Nov. 18, 2009, Washington, D.C.
64. Electronic Theses and Dissertations (ETDs) - A worldwide initiative. University of Iowa, Oct. 29, 2009, http://fox.cs.vt.edu/talks/2009/20091029IowaETD.pptx
65. CTRnet (Crisis, Tragedy, & Recovery Network) (www.ctrnet.net): A global human network and distributed digital library. University of Iowa, Oct. 29, 2009, http://fox.cs.vt.edu/talks/2009/20091029IowaCTRnet.pptx
66. CTRnet: A Crisis, Tragedy, & Recovery Network) (www.ctrnet.net). Virginia College of Osteopathic Research Day, Oct. 16, 2009, http://fox.cs.vt.edu/talks/2009/20091016CTR-VCOM.ppt
67. Donald J. Shoemaker, Steven D. Sheetz, and Edward A. Fox. Initiatives to Assemble a Record. Presentation for the conference Aftermath Dynamics and Management: Through the Lens of the Virginia Tech Incident, July 22, 2009, The Inn, Blacksburg, VA
68. E. Fox. Reiki. Presentation for the Natural Healthcare Series at the Blacksburg Library, July 16, 2009, http://fox.cs.vt.edu/talks/2009/20090716FoxReiki.pdf, http://www.mfrl.org/view_entry.php?id=1812&date=20090716
69. Daphne Rainey and Edward A. Fox. Cyber-Learning & Digital Libraries: are traditional classrooms going the way of the horse and buggy? Cafe Scientifique, Blacksburg, VA, May 26, 2009
70. Molly Bragg, Edward A. Fox, Margaret Hedstrom, and Christopher A. Lee. Moving Web Archiving into the Classroom. Panel at DigCCurr2009: Digital Curation Practice, Promise and Prospects. April 1-3, 2009, Chapel Hill, NC, pp. 112-113 in proceedings: http://www.lulu.com/items/volume_64/6482000/6482253/5/print/DigCCurr2009Proceedings_032609.pdf
71. Jeffrey Pomerantz, Sanghee Oh, Barbara Wildemuth, Carolyn Hank, Helen Tibbo, Edward Fox, and Seungwon Yang. Comparing Curricula for Digital Library and Digital Curation Education. Panel at DigCCurr2009: Digital Curation Practice, Promise and Prospects. April 1-3, 2009, Chapel Hill, NC, pp. 2-3 in proceedings: http://www.lulu.com/items/volume_64/6482000/6482253/5/print/DigCCurr2009Proceedings_032609.pdf
72. Uma Murthy, Edward A. Fox, Yinlin Chen, Eric Hallerman, Ricardo Torres, Evandro J. Ramos, and Tiago R. C. Falco. Image Description and Retrieval for Fish Species Identification. Oral presentation by Uma Murthy at 25th Annual GSA Research Symposium, March 25, 2009, VT, Blacksburg, VA, http://filebox.vt.edu/users/techy/public/RS/
73. Joseph E. Urban, Jesse M. Heines, Edward A. Fox, Harriet G. Taylor: Panel on revitalized undergraduate computing education. In Proceedings of the 40th ACM Technical Symposium on Computer Science Education (Chattanooga, TN, USA, March 04 - 07, 2009). SIGCSE '09. ACM,

New York, NY, 69-70. DOI= http://doi.acm.org/10.1145/1508865.1508893, refereed

74. Edward Fox, "Living In the KnowlEdge Society (LIKES)" (CPATH). Invited presentation for NSF CCLI/CPATH Showcase of the 40th ACM Technical Symposium on Computer Science Education (Chattanooga, TN, USA, March 04 - 07, 2009). SIGCSE '09.
75. Edward A. Fox, presentation as part of Panel 3 - Computational Thinking Everywhere (Part II). Computational Thinking for Everyone Workshop I, The National Academies, 2/19-2/20/2009, Washington, DC
76. Edward A. Fox, "CNI Podcast: An Interview with Edward Fox on Digital Scholarship", Podcast: 28 min., 19.24MB, Editor Gerry Bayne, In CNI Podcasts by EDUCAUSE http://www.educause.edu/blog/gbayne/CNIPodcastAnInterviewwithEdwar/16809, posted on February 3, 2009
77. Kristine Hanna, Edward A. Fox, Jamaica Jones, and Padmini Srinivasan. Capturing Crisis: A Digital Library to Study Tragedy and Recovery from Around the World. Panel at CNI Fall Meeting, Washington, DC, Dec. 9, 2008, http://www.cni.org/tfms/2008b.fall/Abstracts/PB-digital-hanna.html, http://www.cni.org/tfms/2008b.fall/Abstracts/handouts/cni_digital_hanna.pdf
78. Edward A. Fox, Carlos Evia, Weiguo Fan, Steve Sheetz, Chris Zobel, and Seungwon Yang. Sakai LIKES Initiative. Panel at the Sakai Regional Conference, Virginia Tech, Nov. 11-12, 2008
79. NDLTD Welcome and Introduction, invited opening presentation, ETD 2008, June 4-6, Aberdeen Scotland
80. Introduction to NDLTD and Brief History of the ETD Movement, invited opening presentation for ETD Rookies Workshop, ETD 2008, June 4-6, Aberdeen Scotland
81. ETD-db, the Short Story, invited presentation for ETD Rookies Workshop, ETD 2008, June 4-6, Aberdeen Scotland
82. Closing Session, invited closing presentation, ETD 2008, June 4-6, Aberdeen Scotland
83. Manuel A. Perez-Quinones, Stephen Edwards, Manas Tungare, Edward A. Fox, and Lillian Cassel. Using Web 2.0 Technologies in your Computer Science Classes. Workshop #16 at 39th ACM Technical Symposium on Computer Science Education, SIGSCE 2008, March 12-15, 2008, Portland, Oregon
84. Invited visit and presentation at World Bank, Nov. 20, 2007, Washington, D.C.
85. Internet 2 - Digital Library community of Mexico Virtual Day on Digital Theses, 25 min. videoconference presentation, Oct. 5, 2007
86. From information retrieval to digital libraries to computer science education, Sep. 21, 2007, Vienna University of Technology
87. Living In the KnowlEdge Society, July 30, 2007, Univ. Arizona, Tucson
88. Enhancing Computing Education through LIKES, NSDL, and NDLTD. Invited presentation for Google Education Summit 2007 (part of Google Faculty Summit), July 27, 2007, Mountain View, CA
89. 5S Perspective. Invited presentation for 1st Workshop on Digital Library Foundations, ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, http://fox.cs.vt.edu/talks/2007/20070623JCDLdlfWkshpFox-5S.ppt
90. Quality Modeling. Invited presentation for 1st Workshop on Digital Library Foundations, ACM IEEE Joint Conference on Digital Libraries, June 18-23, 2007 - Vancouver, British Columbia, Canada, http://fox.cs.vt.edu/talks/2007/20070623JCDLdlfWkshpFox-Quality.ppt
91. Core DL Curriculum. Invited presentation (with co-authors Seungwon Yang, Barbara M. Wildemuth, Jeffrey Pomerantz, Sanghee Oh) for JCDL 2007 Workshop 1: Developing a Digital Libraries Education Program, with 2007 Joint Conference on Digital Libraries, June 18, 2007, Vancouver, http://fox.cs.vt.edu/talks/2007/20070617JCDLeducWkshpFox-Core.ppt
92. NDLTD: Past, Present, and Future. ETD 2007: 10th Int. Symp. on ETDs, Uppsala, Sweden, http://fox.cs.vt.edu/talks/2007/20070613ETDfoxOpening.ppt
93. Two presentations, Overview of VT/NDLTD Plans: Resources & Projects; Update on Digital Library Technology Trends: From Proposals to Projects to Systems to Theory to Curricula. Invited talks to Scirus and Scopus groups, Elsevier, Amsterdam, 18 March 2007
94. Edward Fox, Uma Murthy, and Ricardo Torres. Automation and Quality in Image Digital Libraries with Annotations, invited presentation for University of Florence, Florence, Italy, 17 Feb. 2007, http://fox.cs.vt.edu/talks/2007/20070216Florence.ppt
95. Reiki at Virginia Tech, invited talk for Companion Animal Club, VA-MD Regional College of Veterinary Medicine, Blacksburg, VA, 26 Jan. 2007, http://fox.cs.vt.edu/talks/2007/20070126ReikiLecture.ppt

96. Indexing and searching heterogeneous information, invited presentation at Lawrence Livermore National Laboratory, Livermore, CA, Nov. 3, 2006
97. "NDLTD Resources & Projects", invited hour-long plenary presentation for ETD 2006 U.S. Regional Conference: Revealing the Potential of ETDs, October 27-28, 2006, J.C. Penney Conf. Center, University of Missouri-St. Louis
98. Invited presentations for SIMPOSIO DE ACESSO LIVRE A INFORMACAO (Symposium on Open Access to Information, http://si.ibict.br), Aug 24-25, 2006, at Hotel Nacional, SHS (Setor Hoteleiro Sul), Quadra 01, Brasilia, DF, Brasil 70322-900; speaker on 2 panels:
    1. "Digital Libraries, Electronic Theses and Dissertations (ETDs), and NDLTD" for 5 hour panel on August 24, 2006, "Open Access para Repositorio Institucionais" (Open Access for Institutional Repository)
    2. "NDLTD: From Local to National to Global", for 3.5 hour panel on August 25, 2006, "BDTD: Biblioteca Digital de Teses e Dissertacoes" (Digital Library of Theses and Dissertations)
99. "Introductionary words", opening talk (30 minutes) for ETD 2006, Quebec City, June 6-10, 2006
100. Digital libraries, computer science, and education: 10 projects. Invited seminar at Fraunhofer IPSI, Darmstadt, GE, Dec. 9, 2005 http://fox.cs.vt.edu/talks/2005/20050929ETDclose.ppt
101. NDLTD Update: Past, Present, Future, and the Expanding ETD Community. Closing plenary invited presentation for ETD2005: evolution through discovery, 8th International Symposium on Electronic Theses and Dissertations, Sydney, Sept. 28-30, 2005, http://fox.cs.vt.edu/talks/2005/20050929ETDclose.ppt
102. ADT and the Future of NDLTD. Invited presentation for ADT Program Workshop, Sydney, Australia, Sept. 26, 2005, http://fox.cs.vt.edu/talks/2005/20050926ADT.ppt
103. Networked Digital Library of Theses and Dissertations (NDLTD, www.ndltd.org). Invited presentation for Oregon State U., 20 July 2005, http://fox.cs.vt.edu/talks/2005/20050720OregonSt.ppt
104. The Global Reach of Electronic Theses and Dissertations (ETDs). Edward Fox, Joan K. Lippincott, Eva Muller, Susan Copeland. Project Briefing: Spring 2005 Task Force Meeting. CNI, April 4-5, 2005, Washington, DC. http://www.cni.org/tfms/2005a.spring/abstracts/PB-fox-global.html
105. Electronic Tools and Near Eastern Archives Digital Library (ETANA-DL). Presentation by Edward A. Fox, with co-presenter ill (James W. Flanagan). Session on "Long-Term Digital Preservation and Archiving Strategies for Archaeological Data" at 106th Annual Meeting of the Archaeological Institute of America, AIA 2005, Boston, January 6-9, 2005
106. Digital Libraries Settling the Score: 10 Years Hence and 10 Before. Moderator for closing plenary panel of the conference. In Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries: Global Reach and Diverse Impact, JCDL2004, Tucson, AZ, June 7-11, 2004, p. 374
107. J. W. Flanagan, E. A. Fox, D. R. Clark, W. Fan, Ravindranathan, R. Shen, M. A. Goncalves, "Jordanian Archaeology and ETANA: Developing a Digital Library for Near Eastern Archaeology", History and Archaeology of Jordan Conference, Petra, Jordan. May 23-27, 2004
108. Improving Graduate Education: Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Villanova University, hosted by graduate school, Jan. 15, 2004, 69 slides
109. Networked Digital Library of Theses and Dissertations (NDLTD), La Red de Bibliotecas Digitales de Tesis y Disertaciones, invited presentation using videoconferencing for Dia Virtual de Bibliotecas Digitales, Nov. 25, 2003, hosted by UDLA, Puebla, Mexico, 41 slides
110. Edward A. Fox and James W. Flanagan. ETANA-DL: Managing complex information applications: An archaeology digital library. 2 hr session (ETANA Workshop: Integrated Technology-Enabled Archaeology), followed by 1 day workshop, at American Schools of Oriental Research (ASOR) Annual Meeting, Atlanta, GA, Nov. 19-22, 2003, 68 slides
111. Research Overview Related to NSDL, invited presentation, Library, UDLA, Puebla, Mexico, Oct. 24, 2003, 86 slides
112. Edward A. Fox and Lillian Cassel. Computing and Information Technology Interactive Digital Educational Library (CITIDEL). Poster at NSDL Annual Meeting, Oct. 12-15, 2003, Washington, DC

113. CITIDEL, in Session: Sustainability II: NSDL (2-3:30pm, Tue). At NSDL Annual Meeting, Oct. 12-15, 2003 Washington, DC
114. Edward A. Fox and Martin Halbert. Needs Addressed by the OCKHAM Project, in Session: NSDL Service Interoperability and Web Services (9-10:30am, Wed). At NSDL Annual Meeting, Oct. 12-15, Washington, DC
115. Distributed vs. Centralized Systems: Funding and Maintenance of International Services. Panel presentation for SINN03, 9/19/2003, Oldenburg, GE
116. Extending Retrieval with Stepping Stones and Pathways. Brief research overview at NSF IDM PI workshop, Sept. 14-16, 2003, Seattle, Washington, 4 slides
117. Integration of regular and static OAI repositories. Presentation for OAI Metadata Harvesting Workshop, JCDL 2003, Houston, TX, 31 May 2003, http://www.cs.cornell.edu/people/simeon/workshops/JCDL2003/
118. Digital Libraries Support for Education: Two Case Studies - CITIDEL and NDLTD. Invited presentation, Dept. of Computer Science, Rice University, Houston, TX, May 28, 2003
119. Panel member, Special Session: Report on the NSF major funding initiative for a National Science, Technology, Engineering, and Mathematics Education Digital Library (NSDL) with special emphasis on the Computing Education Component, ACM SIGCSE'2003, Reno, Nevada, March 16, 2003
120. National Science Digital Library Work in CS at Virginia Tech, AI Seminar, U. of Arizona, Tucson, AZ, March 7, 2003
121. Digital libraries for e-rulemaking: integrating the information fields (hypertext, information retrieval, multimedia, etc.), presentation as part of serving as invited participant in NSF-funded research workshop on "E-Rulemaking: New Directions for Technology and Regulation", Jan. 21-22, 2003, John F. Kennedy School of Government, Harvard, Cambridge, MA
122. OCKHAM, 5S, and ODL. Invited presentation for OCKHAM Initiative Meeting, Emory U., Atlanta, GA, Jan. 8, 2003
123. Edward Fox, Hussein Suleman, and Ming Luo. Building Digital Libraries Made Easy: Toward Open Digital Libraries, invited paper, pages 14-24 in: Ee-Peng Lim, Schubert Foo, Chris Khoo, Hsinchun Chen, Edward Fox, Shalini Urs, Thanos Constantino, eds. Digital Libraries: People, Knowledge, and Technology; Proceedings 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, Dec. 2002. Springer, Lecture Notes in Computer Science 2555.
124. Trends on e-learning from the corporate and academic perspective, and NSDL Governance, invited presentations for IDA (Infocomm Development Authority) Visiting Professor Programme, Dec. 13, 2002, Singapore
125. Open Digital Libraries, reviewed presentation for AOL-CIT Finding Common Ground University Research Day, Nov. 6, Herndon, VA. First Prize Winner for Track 3, with $1500 prize, based on doctoral work of Hussein Suleman
126. Edward A. Fox and Ellen Hoffman, Comments from NSDL PC Chair and Vice Chair, in NSDL New Projects Welcome, repeated Oct. 30 and Nov. 4, 2002, teleconference, http://nsdl.comm.nsdlib.org/community/?pager=61
127. NDLTD and the ETD Initiative: Enhancing Graduate Education, Ohio State University/Virginia Tech Videoconference, Oct. 24, 2002, presented by Edward A. Fox, Gail McMillan, and David Pluska, http://researchsmp2.cc.vt.edu/~fox/tmp/ETDs4OSU20021024.ppt
128. History of ETDs and Practice at Virginia Tech, invited presentation for Howard U. ETD Workshop, Washington, D.C., Oct. 4, 2002
129. Hybrid Partitioned Inverted Files for Large-Scale Digital Libraries, for Dept. of Computer Science Seminar, University of Chile, Sept. 30, 2002, Santiago (hosted by Ricardo Baeza-Yates)
130. Foundations of, and Experiences with, Componentized Digital Libraries. Presentation as part of OCKHAM panel, ECDL'2002, Rome, Sep. 16-18, 2002
131. Invited presentations in 3 cities in Brazil (Rio de Janeiro - visiting PUC Rio, Belo Horizonte - visiting Federal University of Minas Gerais, and Curitiba - visiting Catholic University of Parana), as part of week long visit arranged by the American Embassy in Brasilia and the American Consulate in Rio de Janeiro, Brazil, July 20-28, 2002 (discussing NDLTD, NSDL, and related matters)
132. Panel member and proposer of panel "NSDL (National SMETE Digital Library): From Prototype to Production to Transformational National Resource" at JCDL'2002, Portland, July 14-18, 2002 (p. 368 of JCDL 2002 Proceedings)

133. Mirroring an OAI archive with an I2-DSI channel, invited presentation for session on Distributed Storage for Digital Libraries, Spring 2002 Internet2 Members Meeting in Washington, DC, May 7, 2002 (by videoconference), http://www.internet2.edu/activities/html/spring_02.html
134. Invited presentation as well as speaker on at least one panel, for symposium "What Scholars Need to Know to Publish Today: Digital Writing and Access for Readers", April 8, 2002, University at Albany, Albany, NY
135. Digital Library Support of Education. Invited presentation for mini-conference on ETDs, Southern Illinois University, Carbondale, IL, March 22, 2002.
136. Digital Library Support of Learning: Open Archives Initiative and Educational Demonstrations (NDLTD, CITIDEL/NSDL). Invited presentation at University of Arizona, March 8, 2002 (145 slides)
137. GWU and NDLTD: Technical Issues. Invited presentation at GWU, Washington, D.C., Feb. 13, 2002
138. GWU and NDLTD: General Issues. Invited presentation at GWU, Washington, D.C., Feb. 13, 2002
139. Virginia Tech Perspective on AmericanSouth.org. Presentation for meeting of the project team for AmericanSouth.org: A Collaborative Project to Improve Access to Digital Resources on Southern History and Culture Funded by The Andrew W. Mellon Foundation. Emory University, Atlanta, GA, Feb. 1, 2002.
140. Digital Library Support of Teaching and Learning. Invited presentation for colloquium series of talks in the area of digital libraries (SIS Colloquium Series on Digital Libraries) at School of Information Sciences; and additional talk "Networked Digital Library of Theses and Dissertations" for ETD Working Group, U. of Pittsburgh, Jan. 25, 2002.
141. Networked Digital Library of Theses and Dissertations: New Services, Standards, and Integration with the Open Archives Initiative. Invited presentation for Jornadas ISTEC, Montevideo 2001, "Jornadas sobre Gestion de la Informacion en el siglo XXI", Montevideo, Uruguay, Nov. 12-14, 2001, hosted by UNESCO, ISTEC, ... http://www.ndltd.org/talks/Montevideo2001.ppt
142. From Information Retrieval to Digital Libraries: Scenarios, Software, Systems, Services, Simulations. Presentation for Virginia Tech Dept. of CS Seminar, Blacksburg, VA, Oct. 31, 2001
143. Invited panel (Arts and Humanities Discussion: Non-traditional Theses) presentation for Fall Internet2 Conference "Virtual Internet2 Member Meeting", 4 October 2001, by telephone
144. Invited presentation for panel "Digital Library Programs: Current Status and Future Plans", ECDL'2001, Darmstadt, GE, Sept. 4-8, 2001
145. International Interdisciplinary Open Archives and ETDs. Invited presentation for Workshop, International Interdisciplinary Open Archives and Subject Specific Services in Mathematics and Physics, Duisburg, Germany, 9/3/2001
146. Panel moderator for IDM'2001, Dallas, TX, April 30, 2001
147. Internet Technology Innovation Center at VT. Invited presentation at Fall Centers Director Luncheon, "Interdisciplinary Frontiers 2001: Science and Technology in Service to Society", Virginia Tech, Blacksburg, VA, April 18, 2001
148. Parts 1 and 6, statement for panel 1: Developing the UNESCO International Guide for ETDs: A Multilingual Training Resource of Best Practices, for Fourth International Symposium on Electronic Theses and Dissertations (ETDs), Caltech, Pasadena, CA, March 22-24, 2001, http://www.ndltd.org/talks/etd01p1.ppt
149. Collaboration on Digital Libraries. Invited presentation at NEC RI, Princeton, NJ, Dec. 27, 2000. http://www.ndltd.org/talks/nec2000.ppt
150. Collaboration on Digital Libraries. Invited presentation at Hofstra U., Hempstead, NY, Dec. 22, 2000. http://www.ndltd.org/talks/hofstra2.ppt
151. Digital Libraries: Starting with Authors/Creators. Invited presentation for Digital Strategies 2000, National Archives, College Park, MD, Nov. 16-17, 2000. http://www.ndltd.org/talks/nara.ppt
152. Plenary panel 1 slides for ACM Multimedia '2000: Curricula and Resources for Courses about Multimedia, Oct. 30 - Nov. 4, 2000, Marina del Rey, CA, http://www.ndltd.org/talks/mm00pan1.ppt
153. Digital Libraries, OAI, ETDs, and NDLTD. Invited presentation for SEALS ETD Meeting, Oct. 25, 2000, Port Elizabeth, S. Africa. http://www.ndltd.org/talks/seals.ppt

154. Extending Interoperability of Digital Libraries: Building on the Open Archives Initiative, OAI Workshop at ECDL'2000, Lisbon, September 21, 2000, http://www.ndltd.org/talks/oaiecdl.ppt
155. Networked Digital Library of Theses and Dissertations (www.ndltd.org). Invited presentation for ECDL 2000 Panel 3 "Digital Libraries Programs and International Cooperation", September 20, 2000, Lisbon. http://www.ndltd.org/talks/ecdlpnl.ppt
156. Open Archives initiative and the Networked Digital Library of Theses and Dissertations. Invited presentation for seminar at UCSD, Geisel Lirary, Seuss Room, Aug. 9, 2000 http://www.ndltd.org/talks/UCSD20000809.ppt
157. Virginia Tech's Digital Library Research Laboratory: From Hardware to Software to Projects to Theory, invited seminar talk at IBM T. J. Watson Research Center, July 26, 2000. http://www.ndltd.org/talks/ibm20000726public.ppt
158. Digital Libraries, invited presentation requested by Virginia Center for Innovative Technology for AOL - University Research Day: Finding Common Ground, Reston, VA, July 14, 2000 (and separate breakout session for those desiring further info.)
159. Digital Libraries and the Open Archives Initiative, invited presentation for Louisiana State U., June 30, 2000. http://www.ndltd.org/talks/lsuoai.ppt
160. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for Louisiana State U., June 30, 2000. http://www.ndltd.org/talks/lsundltd.ppt
161. OAi, Electronic Theses, and Scholarly Communication: The Way to the Virtual University, Sonder-Kolloquium Einladung, Carl von Ossietzky Universitat Oldenburg, June 19, 2000 http://www.ndltd.org/talks/oldnbrg.ppt
162. Virtual Universities: Myth or Reality? Invited panel presentation at EUNIS 2000, Poznan, Poland, April 13-14, 2000. http://www.ndltd.org/talks/poznanPanel.ppt
163. Electronic Theses and Dissertations (ETDs): Improving Graduate Education through the Networked Digital Library of Theses and Dissertations (NDLTD), invited campus-wide presentation at Johns Hopkins U., April 7, 2000. http://www.ndltd.org/talks/jhundltd.ppt
164. E-Commerce Business Laboratory, presentations to stimulate e-commerce in Virginia, coordinated by Global Opportunities, Inc. of Blacksburg, VA and the Internet TIC of Virginia: March 21, 2000 at New River Community College, Dublin and March 30, 2000 at Southwest Virginia Higher Education Center, Abingdon, using notes including http://fox.cs.vt.edu/talks/VTITIC.html
165. Digital Libraries: An Aid to Education through Interoperable Open Archives of Resources, invited campus-wide presentation at Johns Hopkins U., April 7, 2000. http://www.ndltd.org/talks/jhuoai.ppt
166. The Open Archives initiative (OAi) and Electronic Theses and Dissertations (ETDs), invited presentation for the Spring 2000 Meeting of the Association of Information and Dissemination Centers (ASIDIC), Orlando, FL, March 27, 2000, http://www.ndltd.org/talks/asidic.ppt
167. Electronic Theses and Dissertations (ETDs): Improving Graduate Education through the Networked Digital Library of Theses and Dissertations (NDLTD), invited campus-wide presentation at U. Kentucky, Lexington, Feb. 24, 2000.
168. Digital Libraries: An Aid to Education through Interoperable Open Archives of Resources, invited campus-wide presentation at U. Kentucky, Lexington, Feb. 24, 2000.
169. Involving New Scholars in Digital Libraries through the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Telcordia, Morristown, NJ, Dec. 30, 1999.
170. Involving New Scholars in Digital Libraries through the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at NEC Research Institute, Princeton, NJ, Dec. 21, 1999.
171. Case study of Information Technologies in Action: Virginia Tech and Digital Libraries, invited presentation for State of the Art (SOTA), "Technology: The Next Revolution", Special Libraries Association (SLA) Nov. Forum, Washington, D.C., Nov. 19, 1999, http://www.sla.org/professional/sota/index.shtml
172. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Virginia Commonwealth U., Richmond, VA, Nov. 17, 1999

173. From NDLTD to NUDL to a NODL Framework, invited presentation for Universal Preprint Service Workshop, Santa Fe, NM, Oct. 21-22, 1999.
174. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for U. New Mexico, Albuquerque, NM, Oct. 20, 1999
175. Digital Libraries: A Pictorial View, invited presentation for Bioinformatics '99, Virginia Tech, Oct. 7-8, 1999.
176. Networked Digital Library of Theses and Dissertations (NDLTD); ETD Needs, Possibilities, Action; invited presentations for Workshop on an international project of electronic dissemination of theses and dissertations, http://www.unesco.org/webworld/etd, UNESCO, Paris, September 27-28, 1999: http://www.unesco.org/webworld/etd/contributions/fox.rtf
177. From NDLTD to Technology for Digital Libraries: Progress and Challenges, part of DL'99 tutorial, Aug. 11, 1999
178. From NDLTD to Technology for Digital Libraries: Progress and Challenges, invited presentation for Ricoh, Tokyo, Japan, July 28, 1999
179. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for NACSIS, Japan, July 28, 1999
180. From NDLTD to Technology for Digital Libraries: Progress and Challenges, invited presentation for IBM Tokyo Research Lab, July 27, 1999
181. From NDLTD to Technology for Digital Libraries: Progress and Challenges, invited presentation for NEC Human Media Research Lab, July 27, 1999
182. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for Tokyo Institute of Technology, Japan, July 26, 1999
183. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for ULIS, Japan, July 23, 1999
184. How to Build Digital Libraries: A Methodology and Examples (NDLTD, CSTC), invited presentation for Kyoto U., Japan, July 22, 1999
185. Digital Libraries to Support Graduate and Undergraduate Education (NDLTD, CSTC), invited presentation at Kobe U., Japan, July 22, 1999.
186. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Arizona State U., July 6, 1999
187. Digital Libraries and their Support of Education, invited presentation at U. of Arizona, Tucson, AZ, July 1, 1999
188. Every University Should Have Its Own Digital Library, invited presentation at IBM Almaden Research Laboratory, June 7, 1999
189. Theses and Dissertations in the Digital Library, invited presentation in Split, Croatia, May 28, 1999.
190. Continuing Evolution of the NDLTD, invited presentation for the ETD Workshop (2nd in series of national/international meetings), Virginia Tech, Blacksburg, VA, May 18, 1999
191. Technology in the Classroom. Presentation for Virginia Tech Multimedia Users Group (VTMMUG, http://www.nmc.vt.edu/vtmmug/) meeting, Litton Reaves 1670, Virginia Tech, April 23, 1999
192. Theses and Dissertations in the Digital Library. Presentation for Spring 1999 CNI Meeting, Renaissance Washington D.C. Hotel, April 16, 1999
193. Networked Digital Library of Theses and Dissertations, invited multimedia presentation sent for launch of ETD initiative at Nanyang Technological Institute (NTU), Singapore, April 8, 1999 (Powerpoint, audio, video)
194. A Computer Science Perspective on Bioinformatics. Presentation by L. Heath, C. Ribbens, and E. Fox, Biochemistry seminar, Virginia Tech, March 29, 1999
195. Enhancing Learning through Digital Libraries and Inter-University Cooperation, invited presentation at Valencia Tech, Valencia, Spain, March 26, 1999 (delivered electronically followed by visit and discussion after plane delays)
196. Digital Libraries, Electronic Publishing, CRIM, NDLTD, and Collaboration, invited presentation for Multimedia-Discussion Group (MUME), Humboldt U., Berlin, Germany, March 25, 1999
197. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for campus community, Caltech, Pasadena, CA, March 12, 1999
198. Current and Future Technical Structure of the NDLTD, invited library presentation, Caltech, Pasadena, CA, March 12, 1999
199. The Networked Digital Library of Theses and Dissertations, invited presentation for CS201 Seminar Series, UCLA Computer Science Department, Los Angeles, March 11, 1999 http://www.cs.ucla.edu/classes/cs201/winter99/calendar.html#Fox
200. Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation for Dept. of Information Studies, UCLA, Los Angeles, March 11, 1999

201. Developing a University-Based Federated Digital Library of Worldwide Graduate Research (NDLTD), Computer Science Colloquium Series, USC, Los Angeles, March 10, 1999
202. Networked Digital Library of Theses and Dissertations: A Vehicle for University Collaboration, seminar hosted by University of North Carolina, Chapel Hill, October 16, 1998.
203. Improving Graduate Education through the Networked Digital Library of Theses and Dissertations (NDLTD), seminar hosted by University of Southampton, England, September 11, 1998
204. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations seminar hosted by University of Arizona, Tucson, AZ, July 30, 1998, http://www.ndltd.org/talks/ndltd980713.ppt
205. The Worldwide Electronic Thesis and Dissertation (ETD) Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), seminar hosted by National Univ. of Singapore, June 30, 1998. http://www.ndltd.org/talks/ndltd980630.ppt
206. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations seminar hosted by University of Pittsburgh, Pittsburgh, PA, June 23, 1998, http://www.ndltd.org/talks/ndltd980623.ppt
207. DL Metrics: Web Characterization and 4S, presented by Edward A. Fox, written with Ghaleb Abdulla and Neill Kipp, DL'98 Metrics Workshop, June 27, 1998, Pittsburgh, PA http://www.canis.uiuc.edu/workshop/presentations.html
208. Position paper (http://ks.com/icdl/254.html) for 1st Summit on International Cooperation on Digital Libraries, 27-28 June 1998, Marriott City Center, Pittsburgh, PA [following Digital Libraries '98] http://www.ks.com/icdl/
209. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations seminar hosted by Ohio State University, Columbus, Ohio, May 29, 1998.
210. Context for DL Collaboration Between VTLS and Virginia Tech, invited presentation, VTLS Inc., Blacksburg, VA, May 12, 1998 http://fox.cs.vt.edu/talks/VTLS.htm
211. Digital Libraries to Enhance Learning: Case Studies in Computer Science and Graduate Education. Invited presentation at Pacific Northwest National Laboratory, Pasco, WA, April 20, 1998.
212. Update on Digital Dissertations. Presentation for Spring 1998 CNI Meeting, Crystal Gateway Marriott, Arlington, April 14, 1998 http://fox.cs.vt.edu/~fox/test/CNIS98.pdf
213. Helping Learners through Digital Libraries: The Networked Digital Library of Theses and Dissertations (NDLTD) and the Computer Science Teaching Center (CSTC), invited seminar for Dept. of Computer Science, Univ. of Mass., Amherst, March 12, 1998. Seminar series page is http://www.cs.umass.edu/csinfo/colloquia/DEPT/Seminars1997-98.html
214. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations seminar hosted by University Denver for all neighboring universities, Denver, CO, March 9, 1998
215. Progress on the National Digital Library of Theses and Dissertations (NDLTD), invited presentation for Computers in Libraries '98, Hyatt Regency, Crystal City, Arlington, VA, March 2, 1998
216. Digital Libraries: Preparing the Next Generation of Scholars in session "Driving Influences: An Update on Issues, Trends and Current Developments that will Affect Secondary Publishers", NFAIS'98, Four Seasons, Philadelphia, PA, Feb. 24, 1998 http://www.pa.utulsa.edu/nfais.html
217. The Networked Digital Library of Theses and Dissertations: Improving Graduate Education, invited seminar for Drexel U. College of Information Science and Technology, Phil. PA, Feb. 23, 1998
218. Proposing a Partnership for Education Innovation: NSF, Virginia Tech, and SCT. Invited presentation, Malvern, PA, Feb. 23, 1998
219. The Worldwide Electronic Thesis and Dissertation Initiative: Building the Networked Digital Library of Theses and Dissertations, presentation with John Eaton for National Agricultural Library, MD, Feb. 20, 1998
220. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations presentation with John Eaton, James Madison U., Harrisonburg, VA, Feb. 19, 1998

221. Digital Libraries, presentation arranged by Assistant Professor of IS, Youngjin Yoo, Weatherhead School of Management, Case Western Reserve U., through videoconference for MIDS411: Advances in Information Technology, Feb. 5, 1998, 6-8pm
222. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations, invited videoconference presentation, Florida International U., 1pm Jan. 27, 1998, hosted by Jackie Zelman, Director, UCS
223. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations, invited presentation, VCU, Richmond, VA, Jan. 12, 1998
224. The Networked Digital Library of Theses and Dissertations, invited presentation for Seminar on Information Access, School of Information Management & Systems, University of California, Berkeley, CA, Jan. 6, 1998
225. Transfer Conference presentations, on changes in Computer Science instructional offerings utilizing instructional technology, Donaldson Brown Hotel and Conference Center, Blacksburg, Dec. 8, 1997. http://fox.cs.vt.edu/talks/Transfer98.htm
226. Distributed Learner Spaces with Digital Libraries: Future digital library technologies across high-speed distributed systems. Presentation for signing of Memorandum of Understanding between VPI&SU and the Institute of Systems Science of Singapore, as part of the ceremony launching the SINGAREN-vBNS connection, Washington, D.C., Nov. 7, 1997. http://www.ndltd.org/talks/singapore
227. Implications of the Electronic Thesis and Dissertation (ETD) Initiative, invited presentation for Electronic Publishing session of DTIC's Annual Users Conference, November 5, 1997, Arlington, VA
228. Improved Education with a Digital Library: the NDLTD Case Study, invited session and presentation (with Gail McMillan) for FIPSE PI meeting, Nov. 2, 1997, Arlington, VA
229. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations, invited presentation at Fall CNI Meeting, Minneapolis, MN 10/26-27/97 http://www.ndltd.org/talks/CNIF97.pdf
230. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at U. Michigan, Oct. 8, 1997
231. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Tech. Univ. of Lisbon, Oct. 2, 1997
232. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at U. Illinois Urbana-Champaign, Sept. 26, 1997
233. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at U. Virginia, Aug. 29, 1997
234. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Florida Inst. of Tech, Aug. 21, 1997
235. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at University N. Florida, Aug. 18, 1997
236. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at U. Pennsylvania, July 29, 1997
237. Panel discussion "NCSTRL: Experience with a Global Digital Library" on Networked CS Technical Report Library, D-Lib Panel on Interoperability I, ACM DL'97, July 24, 1997, Philadelphia, PA
238. Renaissance Consortium Report on the Worldwide ETD Initiative / Networked Digital Library of Theses and Dissertations (NDLTD), IBM Almaden Lab, San Jose, CA, July 11, 1997.
239. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), informal presentation at Stanford U., July 11, 1997
240. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Naval Postgraduate School, July 9, 1997

241. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at U. Ca. Berkeley, July 8, 1997
242. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), informal presentation at U. Ca. Santa Barbara, July 7, 1997
243. Information Retrieval, Digital Libraries, Education Innovation, Theses and Dissertations, and WWW Traffic Analysis/Modeling: Related Work at Virginia Tech. Presentation July 3, 1997, ISS, Singapore.
244. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at National U. Singapore, July 1-3, 1997
245. Powerful Interactivity in a Networked World. Presentation to the Computer Society of Singapore, July 1, 1997, Singapore.
246. The Digital Era: Implcations for Librarians, Information Providers and Users. Presentation for National Computer Board Digital Libraries Cluster. June 30, 1997, ISS Auditorium, Singapore.
247. The Worldwide Electronic Thesis and Dissertation Initiative: Joining the Networked Digital Library of Theses and Dissertations seminar in conjunction with NSF DLI briefing, Carnegie-Mellon U., June 3, 1997
248. Networked Digital Library of Theses and Dissertations, or, the ETD Initiative, presentation at Univ. Waterloo, June 2, 1997, http://www.lib.uwaterloo.ca/~uw-etpt/Flyer.html
249. E. Fox, J. Eaton, G. McMillan. National Digital Library of Theses and Dissertations, invited session for CAUSE/CNI regional conference, Univ. of Del., May 22, 1997.
250. Networked Digital Library of Theses and Dissertations, presentation at NYU, NYC, May 20, 1997.
251. Networked Digital Library of Theses and Dissertations, presentation for the AAP PSP Executive Council, AAP, NY, May 20, 1997.
252. Networked Digital Library of Theses and Dissertations. Invited videoconference presentation at Univ. S. Florida, April 11, 1997 (w. J. Eaton, G. McMillan, N. Kipp)
253. Publishers and Electronic Theses, project briefing at Coalition for Networked Information Spring Meeting, Crystal City, VA, April 1-2, 1997.
254. The Worldwide ETD Initiative: Joining the Networked Digital Library of Theses and Dissertations (NDLTD), invited presentation at Rutgers, The State University, New Brunswick, NJ, March 6, 1997
255. Networked Digital Library of Theses and Dissertations. Invited presentation at Clemson U., March 5, 1997 (w. J. Eaton)
256. Networked Digital Library of Theses and Dissertations. Invited presentation at U. Georgia, Athens, March 5, 1997 (w. J. Eaton)
257. Networked Digital Library of Theses and Dissertations. Invited presentation at U. Alabama, Tuscaloosa, March 4, 1997 (w. J. Eaton)
258. Networked Digital Library of Theses and Dissertations. Invited presentation at U. Alabama, Birmingham, March 4, 1997 (w. J. Eaton)
259. Networked Digital Library of Theses and Dissertations. Invited presentation at U. Tennessee, Knoxville, March 3, 1997 (w. J. Eaton)
260. Networked Digital Library of Theses and Dissertations. Informal presentation at San Jose State U., Feb. 28, 1997
261. Multimedia, Hypertext and Information Access. Panel presentation in session on "Defining Multimedia Courses within a Computer Science Education", ACM SIGCSE'97, San Jose, 2/27 - 3/1/97.
262. Digital Libraries: Theory, Educational Applications, and Use. Invited colloquium series talk for Dept. of Computer & Information Science, Ohio State Univ., Feb. 11, 1997.
263. Networked Digital Library of Theses and Dissertations. Invited presentation at Ohio State U., Feb. 10, 1997
264. Networked Digital Library of Theses and Dissertations. Invited presentation at Vanderbilt U., Jan. 17, 1997 (w. J. Eaton, G. McMillan)
265. National Digital Library of Theses and Dissertations. Invited presentation, AT&T Laboratories, Murray Hill, NJ, Dec. 27, 1996.
266. National Digital Library of Theses and Dissertations. Invited presentation and chairing of working group session on this topic, NSF/DARPA/NASA DLI (Digital Library Initiative) meeting, Dec. 16-17, 1996
267. Electronic Theses and Dissertations, project briefing at Coalition for Networked Information Fall Meeting, San Francisco, Dec. 6-7, 1996

268. The Future Role of Publishers in the New Educational Dynamic. Chaired plenary luncheon for Proc. Frontiers in Education - FIE'96, Salt Lake City, Utah, Nov. 7, 1996.
269. Interactive Learning with a Digital Library in Computer Science. Invited long presentation for NSF CISE EI PIs workshop at FIE'96, Salt Lake City, Utah, Nov. 5-8, 1996.
270. IR Curriculum: Information Engineering to Digital Libraries. Invited presentation for Drexel University hosted Workshop/Symposium sponsored by the W.K. Kellogg Foundation "Information Retrieval 2000 --- Workplace Needs and Curricular Implications", Marriott Hotel, Philadelphia PA, May 24, 1996.
271. Virginia Tech's Digital Library Project. Invited short presentation for IBM Digital Library Consortium Meeting, Case Western Reserve University, Cleveland, May 2, 1996.
272. Perspectives on Information Technology. Invited presentation for the executives of the Corporate Research Center, Blacksburg, VA, Dec. 21, 1995.
273. Digital Libraries and CS Education. Invited Colloquium, George Mason U., Nov. 13, 1995.
274. Education and Research with a Digital Library in Computer Science. Invited presentation for IBM Almaden Research Center, Nov. 3, 1995.
275. Education and Research with a Digital Library in Computer Science. Invited presentation for IBM Watson Research Lab, Oct. 27, 1995.
276. Electronic librarians, intelligent network agents, and information catalogues. Invited presentation for Reconnecting Science and Humanities in Digital Libraries, sponsored by the Univ. of Kentucky and the British Library, 19-21 October 1995, Lexington, KY.
277. Computer Science Technical Reports Library. Invited presentation for Unlocking University Information, sponsored by SURA/SOLINET, Atlanta, Sept. 6-7, 1995.
278. IR Education in CS. Invited presentation for Panel on Education for IR, ACM SIGIR'95, Seattle, WA, July 12, 1995.
279. IR is at the Heart of Digital Libraries and the Global Information Infrastructure. Invited presentation for A SMART Celebration, Cornell Univ., Ithaca, NY, April 22, 1995.
280. How the Blacksburg Electronic Village will Lead to Changes in Careers. Invited presentation for Golden Keys Society Conference, Virginia Tech, Blacksburg, VA, April 1, 1995.
281. Improving CS Education with Digital Libraries. Invited Colloquium for Dept. of Computer Science, SUNY Buffalo, Dec. 2, 1994.
282. Interactive Accessibility. Invited presentation at MCNC, Research Triangle Park, NC, October 30, 1994.
283. Requirements for Knowledge Workers and Education. Invited presentation for Digital Library Academy Workshop, sponsored by IBM Academy of Technology, Edith Macy Conference Center, Briarcliff Manor, NY, Sept. 12-13, 1994.
284. Digital Libraries. Norfolk State University Seminar, Norfolk, VA, March 28, 1994.
285. Components of the Informatics Discipline. Panel presentation for session on Education of the New Information Specialist, ACM CSC'94, ACM Computer Science Conference, Phoenix, AZ, March 8, 1994.
286. Electronic Publishing Experiments: Report on Digital Libraries and their Relationship to SIG Activities. Presentation for ACM SIG Chairs Meeting, Phoenix, AZ, March 6, 1994.
287. Background to Current Work on Digital Libraries. Invited presentation for Workshop on Intelligent Access to On-line Digital Libraries, in connection with IEEE CAIA '94, San Antonio, TX, March 1, 1994
288. Electronic Theses and Dissertations. Invited presentation for Workshop: Innovative Uses of High-Tech in Graduate School Operations, CSGS'94, Conf. of Southern Graduate Schools, 23rd Annual Meeting, Clearwater Beach, FL, Feb. 18-21, 1994.
289. Digital Library Related Research. Seminar for Case Western Reserve University, Cleveland, OH, Dec. 21, 1993.
290. Interactive Learning with a Digital Library in Computer Science. Invited presentation, as part of this year's program on Applications of Computers in Instruction, for Virginia Tech Chapter of Sigma Xi, Nov. 30, 1993, Blacksburg, VA.
291. Worldwide Digital Libraries. Invited presentation for Virginia Tech Center for the Study of Science and Society, Nov. 18, 1993, Blacksburg, VA.

292. Research In Digital Libraries and Networked Multimedia Information Access. Invited presentation for panel on Academic Research Trends in Information Storage and Retrieval, sponsored by SIG/SRT and SIG/MGT, ASIS'93, October 25-29, 1993, Columbus, OH.
293. Improving the Undergraduate Experience through Research in Interactive Accessibility and Digital Libraries. Invited presentation for ACM Student Chapter, Virginia Tech, Blacksburg, VA, Sept. 29, 1993.
294. Networked Multimedia Information Retrieval. Invited presentation for IBM European Networking Center, Heidelberg, Germany, Sept. 16, 1993.
295. Blacksburg Electronic Village and the NREN. Coordinator and presenter in team representing the Blacksburg Electronic Village in briefing at National Science Foundation, Aug. 27, 1993, Washington, D.C.
296. Envision-ing a Computer Science Digital Library. Invited presentation and chair for panel on Digital Libraries of the Future, ACM Multimedia 93, Aug. 4-6, 1993, Anaheim, CA.
297. Technical Reports / Dissertations. Invited project presentation for Monticello Electronic Library meeting, sponsored by SURA, SURAnet, SOLINET, and NSF, July 29-30, 1993, Atlanta.
298. Electronic Dissertation Project. Presentation for Invited SIGIR Panel on Information Retrieval, and panel chair, ACH-ALLC93, Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, June 16-19, 1993, Georgetown Univ., Washington, D.C..
299. An Information Retrieval and Digital Library Perspective. Invited Presentation for Panel on Multimedia Databases, Joint Conf.: 1993 ACM SIGMOD International Conference on Management of Data and Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Washington, D.C., May 25-28, 1993.
300. Envision, the Future. Invited presentation for Digital Libraries of the Future session, Online Publishing '93, Pittsburgh, PA, March 22-24, 1993.
301. Digital Multimedia Standards and Systems. Invited Lecturer for Multimedia and GIS Workshop, sponsored by the National Center for Geographic Information (CNIG) and GIS Europe, Inc., Feb. 24-26, 1993, Lisbon, Portugal
302. Personalized System of Instruction. Presentation and moderating panel on Individualized Instruction: Past, Present, and Future, for LRC Workshop, Virginia Tech, Blacksburg, VA, Feb. 4, 1993.
303. Electronic Libraries. Invited presentation for session on Impacts of Application Environments on Hyperbase Systems, NSF Hyperbases Workshop, Oct. 14-16, 1992, Washington.
304. Project Envision & Information Retrieval Invited presentation for NSF Workshop on Electronic Libraries, July 20-21, 1992, Washington, D.C.
305. Digital Technology and Standards for Multimedia. Invited seminar for Informatik-Kolloquium and HypEdu (Swiss Special Interest Group for Hypermedia and Educational Computing and Software Ergonomics) at ETH (Eldgenossische Technische Hochschule) Zurich, Swiss Federal Institute of Technology Zurich, June 26, 1992.
306. Constructing and Distributing Information Retrieval Databases. Invited presentation for "Databases for Research and Testing in Document Analysis and Information Retrieval" panel at Symposium on Document Analysis and Information Retrieval, March 16-18, 1992, Tropicana Hotel, Las Vegas, Nevada.
307. Developing Interactive Digital Multimedia Applications and Archives. Invited seminar for Graduate Center, CUNY, NY, Feb. 6, 1992. Approaches to Improving Information Access. Invited seminar for AT&T Bell Laboratories Murray Hill, NJ, Feb. 6, 1992.
308. Developing Interactive Digital Multimedia Applications and Archives. Invited presentation for Bell Communications Research, Morristown, NJ, Dec. 23, 1991.
309. Building an Archive of Computer Science Literature. Invited seminar for Department of Computer Science, The Ohio State University, Nov. 5, 1991.
310. Extended Boolean. Invited presentation for "Full Text: From Tutorial to Innovations, Part 2. Beyond Boolean" panel at ASIS '91, American Society for Information Science Annual Meeting, Oct. 28-30, 1991, Washington, D.C. Abstract on conf. proceedings p. 370.

311. New Developments in Information Retrieval Systems and Technology. Invited presentation as part of "The SMART Project in Automatic Document Retrieval" Panel Session at SIGIR '91, 14th Intern'l Conf. on R & D in Information Retrieval, October 13-16, 1991, Chicago.
312. Intelligent Information Retrieval. Invited presentation for Librarian Ex Machine 5.0 session sponsored by LITA, American Library Association 1991 Annual Conf., Atlanta, GA, July 2, 1991.
313. Multimedia Databases. Invited presentation for 1991 NCR Imaging/Multimedia TIES, June 25, 1991, Kitchener, Canada.
314. The Emergence of Digital Video in Interactive Computing Systems. Invited presentation for 1990 NCR Multimedia TIES, October 29-31, 1990, Atlanta, GA.
315. Interactive Digital Video. Invited presentation for Data Processing Management Association, New River Valley Chapter, Sept. 13, 1990, Blacksburg, VA.
316. DVI: A Tool for Tomorrow. Invited demonstration, presentation for Extension Technology Conference: Applied Technology, May 30 - June 2, 1990, Blacksburg, VA.
317. Electronic Publishing and Information Retrieval Research. Invited presentation for joint meeting of Southern and Central Ohio Chapters of ASIS, March 7, 1990, Dayton, OH.
318. Managing Large Data, Information, Knowledge Bases. Invited presentation for University-Wide Seminar on Engineering and Scientific Computing, sponsored by Dept. of Mathematics, Feb. 13, 1990, VPI&SU, Blacksburg, VA.
319. Interactive Digital Video -- Technology and Applications for the Next Generation of Multimedia Systems. Invited presentation for seminar of Dept. of Computer Science, Dec. 13, 1989, Miami University of Ohio, Oxford, Ohio.
320. Advances in Retrieval Software and Database Design. Invited Presentation for Information Industry Association Impact Seminar "The Second Generation of CD-ROM" on Nov. 13, 1989, New York City.
321. Intelligent Handling of Naval Intelligence Messages: The CODER System. Invited presentation for Seventh Intelligence Community Artificial Intelligence / Advanced Computing Symposium, 4-6 October, 1989, Reston, VA.
322. The Lexicon and Information Retrieval panel presentation for SIGIR '89, 12th Int'l Conf. on R&D in Information Retrieval, June 25 - 28, 1989, Cambridge, MA.
323. Development of the CODER Distributed System. Invited presentation for Computer Science Colloquium, University of Utah, March 30, 1989, Salt Lake City, Utah.
324. Expert-Based Retrieval: The CODER Prototype. Invited presentation for "Intelligent Access to Information" session at University of Utah health sciences INFOFAIR, March 30, 1989, Salt Lake City, Utah.
325. Electronic Publishing panel presentation, for ACM Conf. on Document Processing Systems, Dec. 5-9, 1988, Santa Fe, NM.
326. The State of the Art in Information Retrieval: Background for Hypertext. Invited presentation for "The Range and Power of Standards & the Desktop," sponsored by Graphic Communications Assoc. and National Assoc. of Desktop Publishers, Nov. 16-18, 1988, Boston, MA.
327. Experimental Information Retrieval. Invited presentation, Center for Medical Informatics Seminar Series, Nov. 3, 1988, Columbia Univ., NY.
328. Medium and Message: The Future of the Electronic Book. Invited presentation for panel at Univ. of Waterloo Centre for the New Oxford English Dictionary 4th Annual Conf., Information in Text, Oct. 27-28, 1988, Waterloo, Canada.
329. Interfaces for User Modeling. Invited presentation for "Interface Issues in Interactive Information Retrieval," SIG UOI / ACM SIGIR joint session, at 51st American Society for Information Science Annual Mtg., Oct. 23-27, 1988, Atlanta, GA.
330. Hypertext and Hypermedia are Aspects of Information Storage and Retrieval. Invited presentation for NCR sponsored Hypertext Workshop, Oct. 5-7, 1988, Atlanta, GA.
331. Relating Information Retrieval Research to CD-ROM. Invited presentation for "Advanced Information Handling" session, at CD-ROM Expo, Sept. 26-29, 1988, Chicago, IL.
332. The Emergence of CD-ROM: Image vs. Information Access. Invited presentation for Old Dominion Chapter AIIM, Sept. 13, 1988, Richmond VA.

333. Automatic Methods for Information Retrieval. Invited presentation for Annual Meeting of American Association of Law Libraries, June 26-28, 1988, Atlanta, GA.
334. Information Retrieval Models, Methods and Merits. Invited presentation for Microsoft's Third Int'l CD-ROM Conf., March 1-3, 1988, Seattle, WA.
335. Advances in Information Retrieval. First seminar for Lotus Development Corp. Information Retrieval Series, October 5, 1987, Cambridge, MA.
336. UNIX Today. Seminar for Lynchburg Chapter of Data Processing Man. Assoc. Computer Show, September 4, 1987, Lynchburg, VA.
337. Experimental Systems for Advanced Retrieval. Seminar at AT&T Bell Laboratories, July 15, 1987, Holmdel, NJ.
338. Experimental Systems for Advanced Retrieval: SMART and CODER. Seminar for the School of Information Studies, Syracuse University, April 28, 1987, Syracuse, NY.
339. Experimental Systems for Testing the Applicability of Advanced Retrieval Methods: Lessons Learned from SMART and CODER. Seminar for Grad. Library School, Univ. of Chicago, April 9, 1987, Chicago, IL.
340. Why UNIX: A Personal Perspective. Presentation for Roanoke Chapter of Data Processing Management Assoc., March 19, 1987, Roanoke, VA.
341. The CODER System: A Testbed for AI Methods in Information Retrieval. Seminar on "Applications of Artificial Intelligence and Expert Systems to Database Production and Information Retrieval" at NFAIS (National Federation of Abstracting and Information Services) 29th Annual Conference, March 1-4, 1987, Washington, D.C.
342. Why UNIX. Presentation for New River Valley PC Users Group, December 17, 1986, Blacksburg, VA.
343. Effective Retrieval of Composite Documents. NSF Research Panel at American Society for Information Science 49th ASIS Annual Meeting, Sept. 28 - Oct. 2, 1986, Chicago, IL.
344. Information Retrieval Research Opportunities. Presentation for Westinghouse Electric Corp. Technologies Enterprises, June 24, 1986, Monroeville PA.
345. Overview of Information Retrieval Research. Presentation at AT&T Document Development Organization, March 24,1986, Winston- Salem, NC.
346. Research in Electronic Conferencing. Presentation for SIG/UOI Panel at American Society for Information Science 48th American Society for Information Science Annual Mtg., Oct. 20-24, 1985, Las Vegas, NE.
347. Overview of Information Retrieval Research. Seminar at Bell Communications Research, Aug. 7, 1985, Morristown, NJ.
348. Overview of Information Retrieval Research. Seminar at AT&T Bell Laboratories, Aug. 5, 1985, Murray Hill, NJ.
349. Theory and Applications for Extended Information Retrieval. Seminar at AT&T Bell Laboratories, July 30, 1985, Holmdel, NJ.
350. Composite Document Extended Retrieval: An Overview. Seminar at AT&T Bell Laboratories, July 23, 1985, Holmdel, NJ.
351. UNIX in Education. In seminar on "UNIX, Micro Computers, And You," sponsored by Virginia Western Community College and Sperry Corp., May 16, 1985, Roanoke, VA.
352. Improved Retrieval Using a Relational Thesaurus Expansion of Boolean Logic Queries. Presentation and paper for workshop on relational models of the lexicon, June 29, 1984, Stanford Univ., Palo Alto, CA.

TECHNICAL REPORTS, OTHER PUBLICATIONS:

1. Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavvaf, Edward A. Fox. Natural Language Processing Advancements By Deep Learning: A Survey, Mar. 2020, arXiv:2003.01200, https://arxiv.org/abs/2003.01200
2. Liuqing Li, Ziqian Song, Xuan Zhang, Edward A. Fox. A Hybrid Model for Role-related User Classification on Twitter, Nov. 2018, arXiv:1811.10202, https://arxiv.org/abs/1811.10202

3. Hamed Alhoori, Richard Furuta, Mohammed Samaka, Edward A. Fox. Anatomy of Scholarly Information Behavior Patterns in the Wake of Social Media. 23 Dec. 2016. CoRR arXiv:1612.07863 [cs.DL]. https://www.arxiv.org/abs/1612.07863
4. Zhang, Xuan; Qiao, Zhilei; Tang, Lijie; Fan, Patrick (Weiguo); Fox, Edward A.; Wang, Alan (Gang). Identifying Product Defects from User Complaints: A Probabilistic Defect Model. Virginia Tech Department of Computer Science Technical Report TR-16-01, 2016-03-02. http://hdl.handle.net/10919/64902
5. Kanan, Tarek; Ayoub, Souleiman; Saif, Eyad; Kanaan, Ghassan; Chandrasekarar, Prashant; Fox, Edward. Extracting Named Entities Using Named Entity Recognizer and Generating Topics Using Latent Dirichlet Allocation Algorithm for Arabic News Articles. Department of Computer Science Technical Report TR-15-2. Virginia Tech, Blacksburg VA. April 27, 2015. http://hdl.handle.net/10919/51822
6. Tarek Kanan and Edward A. Fox. Automated Arabic Text Classification with P-Stemmer, Machine Learning, and a Tailored News Article Taxonomy. Department of Computer Science Technical Report TR-15-1. Virginia Tech, Blacksburg VA. January 22, 2015. http://hdl.handle.net/10919/51269
7. S.M.Shamimul Hasan, Sandeep Gupta, Edward A. Fox, Keith Bisset, Madhav V. Marathe. Unified Digital Library and Data Access Framework For Heterogeneous and Federated Computational Epidemiology Datasets. NDSSL Technical Report, Network Dynamics and Simulation Science Laboratory Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA. April 2, 2014.
8. Kavanaugh, Andrea, Yang, Seungwon, Sheetz, Steven, Li, Lin Tzy, Fox, Edward (2011) Between a Rock and a Cell Phone: Social Media Use during Mass Protests in Iran, Tunisia and Egypt. Technical Report TR-11-10, Computer Science, Virginia Tech. July 2011. http://eprints.cs.vt.edu/archive/00001149/
9. Kavanaugh, Andrea, Fox, Edward, Sheetz, Steven, Yang, Seungwon, Li, Lin Tzy, Whalen, Travis, Shoemaker, Donald, Nastev, Apostel, Xie, Lexing (2011) Social Media for Cities, Counties and Communities. Technical Report TR-11-09, Computer Science, Virginia Tech. July 2011. http://eprints.cs.vt.edu/archive/00001148/
10. N. Short, A. L. Abbott, M. Hsiao, and E. Fox "A Bayesian Approach to Fingerprint Minutia Localization and Quality Assessment using Adaptable Templates," CESCA Technical Report, CESCA-2011-002, Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, June 2, 2011. http://www.cesca.centers.vt.edu/research/technical_report/TechReport_2011_002.pdf
11. K. Hoyle, N. Short, M. S. Hsiao, and A. L. Abbott, E. A. Fox, "Minutiae + Friction Ridges = Triplet-Based Features for Determining Sufficiency in Fingerprints," CESCA Technical Report, CESCA-2011-001, Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, June 2, 2011. http://www.cesca.centers.vt.edu/research/technical_report/fingerprint_sufficiency.pdf
12. Murthy, Uma, Li, Lin Tzy, Hallerman, Eric, Fox, Edward, Perez-Quinones, Manuel, Delcambre, Lois, Torres, Ricardo (2011) Use of Subimages in Fish Species Identification: A Qualitative Study. Technical Report TR-11-02, Computer Science, Virginia Tech. http://eprints.cs.vt.edu/archive/00001142/
13. Kavanaugh, A., Fox, E., Sheetz, S., Shoemaker, D., Natsev, A., Xie, L. Social Media for Cities, Counties and Communities. Final Grant Report, submitted to Virginia Tech Center for Community Security and Resilience (CCSR), March 2011.
14. Nadia P. Kozievitch, Edward Fox, and Ricardo da S. Torres. Analyzing Compound Object Technologies from the 5S Perspective. University of Campinas (UNICAMP) Instituto de Computacao, Technical Report IC-11-01, Brazil, Jan. 2011, 26 pages, http://www.ic.unicamp.br/~reltech/2011/11-01.pdf
15. N. Short, A. L. Abbott, M. Hsiao, and E. Fox "Extraction of Hierarchical Extended Features in Fingerprint Images," CESCA Technical Report, CESCA-2010-001, Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, November 12, 2010. http://www.cesca.centers.vt.edu/research/technical_report/TechReport_2010_001.pdf
16. Nadia P. Kozievitch, Sherley Codio, Jennifer A. Francois, Edward Fox, and Ricardo da S. Torres. Exploring CBIR concepts in the CTRnet Project. University of Campinas (UNICAMP) Instituto de Computacao, Technical Report IC-10-32, Brazil, November 2010, 20 pages, http://www.ic.unicamp.br/~reltech/2010/10-32.pdf

17. Murthy, U., Kozievitch, N. P., Leidig, J., Torres, R. da S., Yang, S., Goncalves, M., Delcambre, L., Archer, D., and Fox, E. A. Extending the 5S Framework of Digital Libraries to Support Complex Objects, Superimposed Information, and Content-Based Image Retrieval Services. Technical Report TR-10-05, Computer Science, Virginia Tech, April 2010, http://eprints.cs.vt.edu/archive/00001114/.
18. Nadia P. Kozievitch, Ricardo da Silva Torres, Thiago Falcao, Evandro Ramos, Felipe Andrade, Silmara Marques Allegretti, Marlene Tiduko Ueta, Rubens Riscala Madi, Uma Murthy, Edward A. Fox, Yinlin Chen, and Eric Hallerman. Evaluation of a Tablet PC image annotation and retrieval tool in the parasitology domain. University of Campinas (UNICAMP) Instituto de Computacao, Technical Report IC-09-23, Brazil, July 2009, 16 pages, http://www.ic.unicamp.br/~reltech/2009/09-23.pdf
19. Tripathi, Jyotirmaya. ETD Collection and Preservation from India. Independent Study Project Report. Computer Science, Virginia Tech, Blacksburg, VA, July 2008, http://pubs.dlib.vt.edu:9090/193/
20. Volpe, James, McMillan, Gail and Fox, Edward A. Integration of VT ETD-db with Banner. Technical Report TR-08-04, Virginia Tech Department of Computer Science, Jan. 2008, http://eprints.cs.vt.edu/archive/00001018/
21. Andrews, Nicholas O. and Fox, Edward A. Recent Developments in Document Clustering. Technical Report TR-07-35, Virginia Tech Department of Computer Science, Oct. 2007, http://eprints.cs.vt.edu/archive/00001000/
22. Andrews, Nicholas O. and Fox, Edward A. Clustering for Data Reduction: A Divide and Conquer Approach. Technical Report TR-07-36, Virginia Tech Department of Computer Science, Oct. 2007, http://eprints.cs.vt.edu/archive/00000999/
23. Kim, Seonho, Fox, Edward A., Fan, Weiguo, North, Chris, Tatar, Deborah, and Torres, Ricardo da Silva. Design and Evaluation of Techniques to Utilize Implicit Rating Data in Complex Information Systems. Technical Report TR-07-20, Virginia Tech Department of Computer Science, May 2007, http://eprints.cs.vt.edu/archive/00000980/
24. Pomerantz, Jeffrey, Oh, Sanghee, Wildemuth, Barbara M., Yang, Seungwon, and Fox, Edward A. (2007) Digital Library Education in Computer Science Programs. Technical Report TR-07-09, Virginia Tech Department of Computer Science, Feb. 2007, http://eprints.cs.vt.edu/archive/00000945/
25. Matthew Phillips, Edward A. Fox, and Fernando das Neves. Finding Computer Science Syllabi on the World Wide Web. Virginia Tech Dept. of Computer Science Technical Report TR-06-25, Oct. 2006, http://eprints.cs.vt.edu/archive/00000930/
26. Ananth Raghavan, Divya Rangarajan, Rao Shen, Marcos Andre Goncalves, Naga Srinivas Vemuri, Weiguo Fan, Edward A. Fox. Schema Mapper: A Visualization Tool for Digital Library Integration. TCDL Bulletin, 2(1), 2005, http://www.ieee-tcdl.org/Bulletin/v2n1/raghavan/raghavan.html
27. Shen, Rao, Vemuri, Naga Srinivas, Vijayaraghavan, Vidhya, Fan, Weiguo, and Fox, Edward A. EtanaViz: A Visual User Interface to Archaeological Digital Libraries. Technical Report TR-05-14, Computer Science, Virginia Tech, 24 October 2005, http://eprints.cs.vt.edu/archive/00000725/
28. Rangarajan, Divya. SchemaMapper: A tool for visualization of schema mapping. Technical Report TR-05-09, Computer Science, Virginia Tech, 26 August 2005, http://eprints.cs.vt.edu/archive/00000679/
29. Rao Shen, Marcos Andre Goncalves, Weiguo Fan, Edward A. Fox. Requirements Gathering and Modeling of Domain-Specific Digital Libraries with the 5S Framework: An Archaeological Case Study with ETANA. Technical Report TR-05-07, Computer Science, Virginia Tech, 13 April 2005, http://eprints.cs.vt.edu/archive/00000713/
30. Ananth Raghavan, Naga Srinivas Vemuri, Rao Shen, Marcos A. Goncalves, Weiguo Fan, Edward A. Fox. Incremental, Semi-automatic, Mapping-Based Integration of Heterogeneous Collections into Archaeological Digital Libraries: Megiddo Case Study. Technical Report TR-05-06, Computer Science, Virginia Tech, 13 April 2005, http://eprints.cs.vt.edu/archive/00000712/
31. Ananth Raghavan, Divya Rangarajan, Rao Shen, Marcos Andre Goncalves, Naga Srinivas Vemuri, Weiguo Fan, Edward A. Fox. Schema Mapper: A Visualization Tool for DL Integration. Technical Report TR-05-05, Computer Science, Virginia Tech. 13 April 2005, http://eprints.cs.vt.edu/archive/00000711/

32. Rao Shen, Naga Srinivas Vemuri, Ananth Raghavan, Marcos Andre Goncalves, Divya Rangarajan, Weiguo Fan, Edward A. Fox. Integration of Heterogeneous Digital Libraries with Semi-automatic Mapping and Browsing: From Formalization to Specification to Visualization. Technical Report TR-05-04, Computer Science, Virginia Tech, 13 April 2005, http://eprints.cs.vt.edu/archive/00000710/.
33. Xi, Mr. Wensi and Zhang, Mr. Benyu and Fox, Dr. Edward A. (2004) SimFusion: A Unified Similarity Measurement Algorithm for Multi-Type Interrelated Web Objects. Technical Report TR-04-19, Computer Science, Virginia Polytechnic Institute and State University, http://eprints.cs.vt.edu:8000/archive/00000773/
34. Zhang, Baoping and Goncalves, Marcos Andre and Fan, Weiguo and Chen, Yuxin and Fox, Edward A. and Calado, Pavel and Cristo, Marco. Intelligent Fusion of Structural and Citation-Based Evidence for Text Classification. Technical Report TR-04-16, Computer Science, Virginia Tech, 30 August 2004, http://eprints.cs.vt.edu:8000/archive/00000758/
35. Kanade, Abhijat. Project AlcoZone - Independent Study Report supervised. Technical Report TR-04-11, Computer Science, Virginia Tech, 14 May 2004, http://eprints.cs.vt.edu:8000/archive/00000748/
36. Kampanya, Nithiwat. User Study of the Digital Library Clustering Result Visualization. Technical Report TR-04-10 supervised, Computer Science, Virginia Tech, 28 April 2004, http://eprints.cs.vt.edu:8000/archive/00000747/.
37. Scarborough, M.; Fox, E.; and Arnold, L. Search Tool Implementation for Historical Archive. Technical Report TR-04-09, Computer Science, Virginia Tech, 7 June 2004, http://eprints.cs.vt.edu:8000/archive/00000746/.
38. Ravindranathan, Unni and Shen, Rao and Goncalves, Marcos and Fan, Weiguo and Fox, Edward and Flanagan, James (2004) ETANA-DL: Managing Complex Information Applications - An Archaeology Digital Library. Technical Report TR-04-05, Computer Science, Virginia Tech, Feb. 2004, http://eprints.cs.vt.edu/archive/00000684/
39. Ravindranathan, Unni and Shen, Rao and Goncalves, Marcos and Fan, Weiguo and Fox, Edward and Flanagan, James. ETANA-DL: A Digital Library for Integrated Handling of Heterogeneous Archaeological Data. Technical Report TR-04-04, Computer Science, Virginia Tech, Feb. 2004, http://eprints.cs.vt.edu/archive/00000683/
40. Torres, Ricardo and Medeiros, Claudia and Goncalves, Marcos Andre and Fox, Edward. A Digital Library Framework for Biodiversity Information Systems. Technical Report TR-04-01b, Computer Science, Virginia Tech, 26 August 2005, http://eprints.cs.vt.edu/archive/00000681/
41. Perugini, Saverio and McDevitt, Kathleen and Richardson, Ryan and Perez-Quinones, Manuel and Shen, Rao and Ramakrishnan, Naren and Williams, Chris and Fox, Edward A. Enhancing Usability in CITIDEL: Multimodal, Multilingual, and Interactive Visualization Interfaces. Feb. 2004, http://eprints.cs.vt.edu:8000/archive/00000713/. Also in Proceedings Fourth ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL2004), Tucson, AZ, 10 pages. http://eprints.cs.vt.edu:8000/archive/00000713/
42. Krowne, Aaron and Fox, Edward A. (2003) An Architecture for Multischeming in Digital Libraries. Technical Report TR-03-25, Computer Science, Virginia Tech, Sep. 2003, 17 pages. http://eprints.cs.vt.edu:8000/archive/00000692/
43. Goncalves, Marcos and Zhu, Qinwei and Kelapure, Rohit and Fox, Edward (2003) Rapid Modeling, Prototyping, and Generation of Digital Libraries- A Theory-Based Approach. Technical Report TR-03-16, Computer Science, Virginia Tech, June 2003, 11 pages. http://eprints.cs.vt.edu:8000/archive/00000676/
44. Fox, Edward and Carroll, John and Fan, Patrick and Cassel, Lillian and Zubair, Mohammed and Maly, Kurt and McMillan, Gail and Ramakrishnan, Naren and Halbert, Martin (2003) Science of Digital Libraries(SciDL). Technical Report TR-03-13, CS, Virginia Tech, June 2003, 17 pages. http://eprints.cs.vt.edu:8000/archive/00000671/
45. Fox, Dr. Edward and Garach, Kunal (2003) CITIDEL Collection Building. Technical Report TR-03-14, Computer Science, Virginia Polytechnic Institute and State University, May 2003, 34 pages. http://eprints.cs.vt.edu:8000/archive/00000663/
46. Suleman, Senior Lecturer in Computer Science Hussein and Fox, Professor of Computer Science Edward and Krowne, Graduate Student Aaron and Luo, Graduate Student Ming and Kelapure, Graduate Student Rohit (2003) Building Digital Libraries from Simple Building Blocks. Technical Report TR-03-09, Computer Science, Virginia Tech, April 2003, 17 pages. http://eprints.cs.vt.edu:8000/archive/00000656/

47. Goncalves, Marcos Andre and Fox, Edward A. and Watson, Layne T. and Kipp, Neill A. (2003) Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. Technical Report TR-03-04, Computer Science, Virginia Tech. http://eprints.cs.vt.edu/archive/00000653/
48. Bazaz, Anil and Fox, Edward A. (2003) PRESERVATION OF ETDs ON NDLTD Version 1.0. Technical Report TR-03-02, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000640/
49. Saverio Perugini, Marcos Andre Goncalves, and Edward A. Fox. A Connection-Centric Survey of Recommender Systems Research, May 2002, arXiv:cs/0205059, https://arxiv.org/abs/cs/0205059
50. Prabhune, Aniket and Fox, Edward (2002) XML for ETDs. Technical Report TR-02-28, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000635/
51. Suleman, Hussein and Fox, Edward A. (2002) Beyond Harvesting: Digital Library Components as OAI Extensions. Technical Report TR-02-25, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000625/
52. Fox, Edward A. (2002) Overview of a Guide for Electronic Theses and Dissertations. Technical Report TR-02-24, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000624/
53. Fox, Edward A. (2002) Overview of Digital Library Components and Developments. Technical Report TR-02-23, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000623/
54. Koneru, Srikanth and Fox, Edward (2002) Open Peer to Peer Technologies. Technical Report TR-02-19, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000613/
55. Wang, Li and Fox, Edward A. (2002) Crawling on the World Wide Web. Technical Report TR-02-10, Computer Science, Virginia Tech. http://eprints.cs.vt.edu:8000/archive/00000572/
56. Anthony Atkins, Edward Fox, Robert France, Hussein Suleman. ETD-MS: An Interoperability Metadata Standard for Electronic Theses and Dissertations - version 1.00, revision 2, http://www.ndltd.org/standards/metadata/ETD-MS-v1.00-rev2.html, http://www.ndltd.org/standards/metadata/current.html
57. M. Goncalves, E. Fox, L. Watson, and N. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. Virginia Tech Department of Computer Science Technical Report TR-01-12, July 2001. http://eprints.cs.vt.edu/archive/00000536/
58. E. Fox. 1999 NSF Information and Data Management Workshop: Research Agenda for the 21st Century - IDM'99, entered in NSF's FastLane 6/9/2001
59. Neill Kipp, Edward A. Fox, Gail McMillan and John L. Eaton. FIPSE Final Report, 11/30/99, in PDF (http://www.ndltd.org/pubs/FIPSEfr.pdf) and Word (http://www.ndltd.org/pubs/FIPSEfr.doc) formats.
60. Robert K. France, Lucy Terry Nowell, Edward A. Fox, Rani A. Saad, Jianxin Zhao. Use and usability in a digital library search system, Feb. 1999, arXiv:cs/9902013, https://arxiv.org/abs/cs/9902013
61. B. Liu, G. Abdulla, and E. Fox. Web Traffic Latency: Characteristics and Implications. Virginia Tech Dept. of Computer Science Technical Report, July 1998.
62. O. Balci, C. Ulusarac, P. Shah and E. Fox. A Library of Reusable Model Components for Visual Simulation of the NCSTRL System. TR-98-02, Virginia Tech Dept. of Computer Science Technical Report, Jan. 1998. URN: ncstrl.vatech_cs/TR-98-02
63. G. Abdulla, A.H. Nayfeh and E. Fox. Modeling Correlated Proxy Web Traffic Using Fourier Analysis. TR-97-19, Virginia Tech Dept. of Computer Science Technical Report, Nov. 1997. URN: ncstrl.vatech_cs/TR-97-19
64. G. Abdulla, B. Liu, R. Saad and E. Fox. Characterizing World Wide Web Queries. TR-97-04, Virginia Tech Dept. of Computer Science Technical Report, March 1997. URN: ncstrl.vatech_cs/TR-97-04
65. G. Abdulla, E. Fox, M. Abrams and S. Williams. WWW Proxy Traffic Characterization with Application to Caching. TR-97-03, Virginia Tech Dept. of Computer Science Technical Report, March 1997. URN: ncstrl.vatech_cs/TR-97-03

66. E. Fox. Section in "Report on Building and Using Test Collections Panel, SIGIR 1996", ed. Donna Harman. ACM SIGIR Forum, Fall 1996, 30(2): 8.
67. G. Abdulla, M. Abrams and E. Fox. Scaling the World-Wide Web. TR-96-06, Virginia Tech Dept. of Computer Science Technical Report, March, 1996.
68. E. Fox. Virginia Tech Department of Computer Science Information Access Laboratory. ACM SIGIR Forum, Lab Report Special Section, 30(1), Spring 1996.
69. H. Gladney, Z. Ahmed, R. Ashany, N. Belkin, E. Fox and M. Zemankova. Digital Library: Gross Structure and Requirements (Report from a Workshop). IBM Research Report RJ9840, IBM Almaden Research Center, May, 1994. Virginia Tech Dept. of Computer Science Technical Report 94-25, June, 1994.
70. K. Maly, J. French, A. Selman and E. Fox. Wide Area Technical Report Service, TR_94_13, Old Dominion Univ. Dept. of Computer Science, June 1994.
71. L. Nowell, E. Fox, L. Heath, D. Hix, W. Wake and E. Labow. Seeing Things Your Way: Information Visualization for a User-Centered Database of Computer Science Literature, TR-94-06, VPI&SU Computer Science Dept., Jan. 1994, Blacksburg, VA.
72. K. Dalal and E. Fox. Document Translation: Dissertations and Technical Reports, TR-93-31, VPI&SU Computer Science Dept., Sep. 21, 1993, Blacksburg, VA.
73. J. Leggett, J. Schnase, J. Smith, E. Fox, eds. Final Report of the NSF Workshop on Hyperbase Systems. Texas A&M Univ. Dept. of Computer Science Hypermedia Research Lab Report TAMU-HRL 93-002, July 1993. For workshop on Oct. 15-16, 1992 in Washington, D.C.
74. E. Fox, R. France, E. Sahle, A. Daoud, and B. Cline. Development of a Modern OPAC: From REVTOLC to MARIAN, TR-93-06, VPI&SU Computer Science Dept., Feb. 17, 1993, Blacksburg, VA.
75. S. Betrabet, E. Fox, and Q. Chen. A Query Language for Information Graphs, TR-93-03, VPI&SU Computer Science Dept., Jan. 1993, Blacksburg, VA.
76. K. Maly, E. Fox, J. French, and A. Selman. Wide Area Technical Report Service, TR_92_44, Old Dominion Univ. Dept. of Computer Science, Dec.1992
77. E. Fox. Book review for Managing Interactive Video/MultimediaProjects, by Bergman & Moore, ACM Computing Reviews, April 1992.
78. S. Patel, G. Abdulla, M. Abrams, and E. Fox. NMFS: Network Multimedia File System Protocol, TR-92-54, VPI&SU Computer Science Dept., Nov. 1992, Blacksburg, VA.
79. D. Brueni, E. Fox, L. Heath, D. Hix, L. Nowell, and W. Wake. What if there were Desktop Access to the Computer Science Literature?, TR-92-42, VPI&SU Computer Science Dept., Aug. 12, 1992, Blacksburg, VA.
80. E. A. Fox, Qi Fan Chen, and L. S. Heath. LEND and Faster Algorithms for Constructing Minimal Perfect Hash Functions, TR-92-02, VPI&SU Computer Science Dept., Feb. 1992, Blacksburg, VA.
81. E. A. Fox and L. Wilson. Comparison of Advanced Retrieval Approaches for Online Catalog Access. Final Report, Virginia Polytechnic Institute and State University, Blacksburg, Univ. Lib., ERIC Clearinghouse on Information Resources, ED 338 262, 1991, 21 pages.
82. C.A. Shaffer, L.W. Carstensen, V.F. Miranda, S.K. Kriss, R.W. Morrill, and E.A. Fox. Project GeoSim: the first two modules, TR 91-26, VPI&SU Comp. Sci. Dept., Oct 1991.
83. E. A. Fox, M. Prabhakar Koushik, Qi Fan Chen, and Robert K. France. Integrated access to a large medical literature database, TR-91-15, VPI&SU Computer Science Dept., May 1991, Blacksburg, VA.
84. E. A. Fox and W. C. Lee. FAST-INV: A Fast Algorithm for Building Large Inverted Files, TR-91-10, VPI&SU Computer Science Dept., April 1991, Blacksburg, VA.
85. E. Fox, Q.-F. Chen, A. Daoud, and L. Heath. Order Preserving Minimal Perfect Hash Functions and Information Retrieval, TR-91-1, VPI&SU Computer Science Dept., Feb. 1991, Blacksburg, VA.

86. E. Fox. Final Report on NSF Grant IRI-8703580: Organizing Lexical Knowledge for Information Retrieval. VPI&SU Dept. of Computer Science, Oct. 1990, Blacksburg, VA.
87. E. A. Fox. Advanced Retrieval Methods for Online Catalogs. Report under "External and Collaborative Research" section of Annual Review of OCLC Research, July 1989 - June 1990, OCLC Online Computer Library Center, Inc., Dublin, OH, 32-34.
88. E. A. Fox, L. S. Heath, Q.-F. Chen, and A. M. Daoud. Practical Minimal Perfect Hash Functions for Large Databases, TR-90-41, VPI&SU Computer Science Dept., Aug. 1990, Blacksburg, VA.
89. J.T. Nutter, E.A. Fox, and M.W. Evans. Building a Lexicon from Machine-Readable Dictionaries for Improved Information Retrieval, TR-90-17, VPI&SU Computer Science Dept., 1990, Blacksburg, VA.
90. E.A. Fox, D. Hix, E.E. Schwartz, A. Siochi, P. Koushik, and D. Inman. Interactive Digital Video Authoring and Prototyping, TR-90-13, VPI&SU Computer Science Dept., Dec. 1989, Blacksburg, VA.
91. R.B. Quizon and E. Fox. Virginia Disc 2: Preparing, Presenting and Retrieving MARC Standard Library Data on a CDROM, TR-90-5, VPI&SU Computer Science Dept., Feb. 1990, Blacksburg, VA.
92. E. Fox, L. Heath, and Q.-F. Chen. An O(n log n) Algorithm for Finding Minimal Perfect Hash Functions, TR-89-10, VPI&SU Comp. Sci. Dept., April 1989, Blacksburg, VA.
93. S. Datta and E. Fox. Implementation of a Perfect Hash Function Scheme, TR-89-9, VPI&SU Computer Sci. Dept., April 1989, Blacksburg, VA.
94. P. M. Koushik and E. A. Fox. CD-ROM: The New Wave in Information Storage. In CS Bits & Bytes, 1(3): 1,4; VPI&SU Comp. Sci. Dept., Feb. 15, 1989, Blacksburg, VA.
95. E. Fox, Q.-F. Chen, L. Heath, and S. Datta. A More Cost Effective Algorithm for Finding Perfect Hash Functions, TR-88-30, VPI&SU CS Dept., Sept. 1988, Blacksburg, VA.
96. W. Lee and E. Fox. Experimental Comparison of Schemes for Interpreting Boolean Queries, TR-88-27, VPI&SU Comp. Science Dept., September 1988, Blacksburg, VA.
97. M. Weaver, R. France, Q. Chen, E. Fox. A Frame-Based Language in Information Retrieval, TR-88-25, VPI&SU Comp. Science Dept., Aug. 1988, Blacksburg, VA.
98. E. Fox. A Review of Publishing and Access Issues for Optical Discs and CD-ROMs, TR-88-22, VPI&SU Computer Science Dept., August 1988, Blacksburg, VA.
99. R. C. Wohlwend and E. Fox. Creation of a Prolog Fact Base from the Collins English Dictionary, M.S. thesis, TR-88-24, VPI&SU Comp.Sci. Dept., 1988, Blacksburg, VA.
100. R. K. France and E. Fox. An Artificial Intelligence Environment for Information Retrieval Research, M.S. thesis, TR-88-10, VPI&SU Comp.Sci. Dept., 1988, Blacksburg, VA.
101. M. Weaver and E. Fox. Implementing an Intelligent Retrieval System: The CODER System, Version 1.0, TR-88-6, VPI&SU Comp.Sci. Dept., Feb.1988, Blacksburg, VA.
102. E. Fox. Final Report on NSF Grant IST-8418877: Effective Retrieval of Composite Documents. VPI&SU Dept. of Computer Science, Oct. 1987, Blacksburg, VA.
103. E. Fox. Final Report on CIT Grant INF-85-016: Improved Automatic Analysis, Indexing, Storage and Retrieval of Corporate Documentation and Network Information. VPI&SU Dept. of Computer Science, Oct. 1987, Blacksburg, VA.
104. E. Fox. Workshop on Distributed Expert-Based Information Systems: A Perspective. ACM SIGIR Forum, Spring/Summer 1987, 21(3-4):18-20.
105. E. Fox. Development of the CODER System: A Test-Bed for Artificial Intelligence Methods in Information Retrieval, TR-86-40, VPI&SU Computer Science Dept., December 1986, Blacksburg, VA.
106. E. Fox, R. Wohlwend, P. Sheldon, Q. Chen, and R. France. Building the CODER Lexicon: The Collins English Dictionary and Its Adverb Definitions, TR-86-23, VPI&SU Computer Science Dept., October 1986, Blacksburg, VA.

107. E. Fox. A Call for Integrating Advanced Information Retrieval Models with CD-ROM / Microcomputer Systems, TR-86-14, VPI&SU Computer Science Dept., June 1986, Blacksburg, VA.
108. E. Fox. Expert Retrieval for Computer Message Systems, TR-86-13, VPI&SU Computer Science Dept., June 1986, Blacksburg, VA.
109. E. Fox and S. Birch. A UNIX Requirement for Computer Science Majors, TR-86-11, VPI&SU Computer Sci. Dept., May 1986, Blacksburg, VA.
110. E. Fox and R. France. Architecture of an Object-Oriented Expert System for Composite Document Analysis, Representation, and Retrieval, TR-86- 10, VPI&SU Computer Sci. Dept., April 1986, Blacksburg, VA.
111. E. Fox and R. France. A Knowledge-Based System for Composite Document Analysis and Retrieval: Design Issues in the CODER Project, TR-86-6, VPI&SU Computer Science Dept., March 1986, Blacksburg, VA.
112. E. Fox and S. Sharan. A Comparison of Two Methods for Soft Boolean Operator Interpretation in Information Retrieval, TR-86-1, VPI&SU Computer Science Dept., Jan. 1986, Blacksburg, VA.
113. E. Fox and J. Mindel. Communication and Information Sharing in Virginia, Final Report for Contract C-85-026, Virginia Center for Innovative Technology, Jan. 1986.
114. E. Fox. Development of a Prototype Electronic Mail and Bulletin Board Distributed System to Facilitate C.I.T. Operations, Supplemental Report for Contract C-85-026, Virginia Center for Innovative Technology, May 1986.
115. E. Fox and J. Mindel. Summary Report on the Virginia Networking and Telecommunications Planning Conference (Oct. 15/16, 1985, Fredericksburg, VA), Virginia Center for Innovative Technology, Jan. 1986
116. E. Fox. Book Review of Representation and Exchange of Knowledge as a Basis of Information Processes, H. J. Dietschmann (ed.). In Information Processing & Management, 1985, 21(5): 465-466.
117. E. Fox. Information Storage and Retrieval of Composite Documents: A User Oriented Model of Computer Message Systems. TR-85-34, VPI&SU Computer Science Dept., Aug. 1985, Blacksburg, VA.
118. E. Fox. Composite Document Extended Retrieval: An Overview.TR-85-1, VPI&SU Computer Science Dept., Jan. 1985, Blacksburg, VA.
119. E. Fox. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts. TR 83-561, Cornell Univ. Dept. of Computer Science, Sept. 1983, Ithaca, NY.
120. E. Fox. Some Considerations for Implementing the SMART Information Retrieval System under UNIX. TR 83-560, Cornell Univ. Dept. of Comp. Science, Sept. 1983, Ithaca, NY.
121. E. Fox. Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Cornell University dissertation, Aug. 1983. Available from: University Microfilms International, Ann Arbor, Michigan.
122. G. Salton, E. Fox and E. Voorhees. Advanced Feedback Methods in Information Retrieval. TR 83-570. Cornell Univ. Dept. of Computer Science, Aug. 1983, Ithaca, NY.
123. G. Salton, E. Fox and E. Voorhees. A Comparison of Two Methods for Boolean Query Relevance Feedback. TR 83-564, Cornell Univ. Dept. of Computer Science, July 1983, Ithaca, NY.
124. G. Salton, C. Buckley and E. Fox. Boolean Query Formulation with Relevance Feedback. TR 83-539, Cornell Univ. Dept. of Computer Science, Jan. 1983, Ithaca, NY.
125. E. Fox. Lexical Relations: Enhancing Effectiveness of Information Retrieval Systems. ACM SIGIR Forum, Winter 1980, 15(3): 5-36.

ACM SIGIR FORUM CHAIRMAN STATEMENTS:

1. E. Fox. ACM SIGIR Forum, Fall 1994, 28(2): 1-3
2. E. Fox. ACM SIGIR Forum, Spring 1994, 28(1): 1-2.

3. E. Fox. ACM SIGIR Forum, Fall 1993, 27(3): 1-3.
4. E. Fox. ACM SIGIR Forum, Spring 1993, 27(1): 1-2.
5. E. Fox. ACM SIGIR Forum, Fall 1992, 26(2): 2-3.
6. E. Fox. ACM SIGIR Forum, Spring 1992, 26(1): 1.
7. E. Fox. ACM SIGIR Forum, Fall 1991, 25(2): 2-3.

REVIEWS:

- E. Fox. Review of "Digital Libraries" by William Y. Arms, MIT Press, 2000, Kluwer Journal "Information Retrieval", 2001.

THESIS AND DISSERTATION ONLINE DOCUMENTS (and Reports, from Selected Advisees):

1. Prashant Chandrasekar. Continuously Extensible Information Systems: Extending the 5S Framework by Integrating UX and Workflows. Doctoral dissertation to appear in 2020 in https://vtechworks.lib.vt.edu. (2020). Virginia Tech, Computer Science
2. Saurabh Chakravarty, Summarizing Legal Depositions. Doctoral dissertation to appear in 2020 in https://vtechworks.lib.vt.edu. (2020). Virginia Tech, Computer Science
3. Amirsina Torfi, Privacy-Preserving Deep Learning to Generate Synthetic Healthcare Records. Doctoral dissertation to appear in 2020 in https://vtechworks.lib.vt.edu. (2020). Virginia Tech, Computer Science
4. Sampanna Yashwant Kahu, "Figure extraction from scanned electronic theses and dissertations", August 2020, MS thesis, Electrical and Computer Engineering, to appear
5. Ziqian Song, The Impact of Operational Crisis on Firm Equity Value: An Event-driven Approach. Doctoral dissertation to appear in 2020 in https://vtechworks.lib.vt.edu. (2020). Virginia Tech, Computer Science
6. Maanav Mehrotra, "Generating Canonical Sentences from Question-Answer Pairs of Deposition Transcripts", July 2020, MS thesis, Computer Science, to appear
7. Palakh Mignonne Jude, "Increasing Accessibility of Electronic Theses and Dissertations (ETDs) Through Chapter-level Classification", June 2020, MS thesis, Computer Science
8. Harinni Kodur Kumar, "ACM Venue Recommendation System", May 2020, MS thesis, Computer Science, http://hdl.handle.net/10919/96211
9. Farnaz Khaghani, "A deep learning approach to predict accident occurrence based on the traffic dynamics", May 2020, MS thesis, Computer Science, http://hdl.handle.net/10919/98801
10. Liuqing Li, "Event-related Collections Understanding and Services", final Ph.D. dissertation defense 2/18/2020, Virginia Tech, Computer Science, http://hdl.handle.net/10919/97365
11. Ashish Malpani, "Tweets Clustering and Visualization", ECE 5904 MS Project and Report, June 2019
12. Pranavi Rambhakta, "A Large Collection Learning Optimizer Framework", ECE 5904 MS Project and Report, June 2019
13. Ashish Baghudana, "A Web Framework for Text Extraction, Segmentation, and Summarization of Long Documents", CS5974 MS Project and Report, June 2019, Computer Science
14. Adithya Upadhya, "A general web platform for summarizing text and documents", CS5974 MS Project and Report, June 2019, Computer Science
15. Supritha Patil, "Analysis of moving events using tweets", June 2019, MS thesis, Computer Science, http://hdl.handle.net/10919/90884

16. Abhinav Kumar, "SensAnalysis: A Big Data Platform for Vibration-Sensor Data Analysis", May 2019, MS thesis, Computer Science
17. Raja Venkata Satya Phanindra Chava, "Natural Language Processing techniques for comprehending legal depositions", ECE 5904 MS Project and Report, April 2019
18. Yufeng Ma, "Going Deeper with Images and Natural Language", final defense 2/20/2019, Ph.D. dissertation, Computer Science
19. Xuan Zhang, "Product Defect Discovery and Summarization from Online User Reviews", final defense 8/31/2018, Ph.D. dissertation, Computer Science, http://hdl.handle.net/10919/85581
20. Abigail Bartolome, "Describing Trail Cultures through Studying Trail Stakeholders and Analyzing their Tweets", June 2018, MS thesis, Computer Science, http://hdl.handle.net/10919/84528
21. Nikhil Komawar, "SOLR supported search on an OpenStack metadata service", August 2017, MS independent study report, Virginia Tech Dept. of Computer Science, DLRL Technical Report, http://hdl.handle.net/10919/78741
22. S.M.Shamimul Hasan, "A Semantic Web-Based Digital Library Infrastructure to Facilitate Computational Epidemiology", final defense Aug. 10, 2017, ETD approved 9/15/2017, Ph.D. dissertation, http://hdl.handle.net/10919/88386
23. Yinlin Chen, "A High-quality Digital Library Supporting Computing Education: The Ensemble Approach", final defense July 27, 2017, Ph.D. dissertation, http://hdl.handle.net/10919/78750
24. Shivam Maharshi, "Performance Measurement and Analysis of Transactional Web Archiving", May 2017, MS thesis, Computer Science, http://hdl.handle.net/10919/78371
25. Matthew Bock, "A Framework for Hadoop Based Digital Libraries of Tweets", May 2017, MS thesis, Computer Science, http://hdl.handle.net/10919/78351
26. Saurabh Chakravarty, "A Large Collection Learning Optimizer Framework", June 2017, MS thesis, Computer Science, http://hdl.handle.net/10919/78302
27. Saket Dilip Vishwasrao, "Performance Evaluation of Web Archiving Through In-Memory Page Cache", June 2017, MS thesis, Computer Engineering, http://hdl.handle.net/10919/78252
28. Sunshin Lee, "Geo-Locating Tweets with Latent Location Information", Ph.D. dissertation, defended December 2016, published 2017-02-13 in VTechWorks: http://hdl.handle.net/10919/75022
29. Andrej Galad, "ArchiveSpark - MS Independent Study Final Submission", December 2016, MS independent study report, Virginia Tech Dept. of Computer Science, DLRL Technical Report, http://hdl.handle.net/10919/77457
30. Mohamed Magdy Gharib Farag, "Intelligent Event Focused Crawling", Ph.D. dissertation, 23 September 2016, http://hdl.handle.net/10919/73035
31. Tarek Ghaze Kanan, "Arabic News Text Classification and Summarization: A Case of the Electronic Library Institute SeerQ (ELISQ)", June 2015, Ph.D. dissertation, published 2015-07-21 in VTechWorks: http://hdl.handle.net/10919/74272
32. Kiran Chitturi, "Building CTRnet Digital Library Services using Archive-It and LucidWorks Big Data Software", March 2014, MS thesis, http://hdl.handle.net/10919/46865
33. Seungwon Yang, "Automatic Identification of Topic Tags from Texts Based on Expansion-Extraction Approach", Jan. 2014, Ph.D. dissertation, http://hdl.handle.net/10919/25111
34. Monika Akbar, "Integrating Community with Collections in Educational Digital Libraries", Jan. 2014, Ph.D. dissertation, http://hdl.handle.net/10919/25139
35. Sai Tulasi Neppali, "Managing the Development of Digital Libraries, Aided by a Tool Based on 5S", Jan. 2014, MS thesis, http://hdl.handle.net/10919/25218
36. Sung Hee Park, "Discipline-Independent Text Information Extraction from Heterogeneous Styled References Using Knowledge from the Web", June 2013, Ph.D. dissertation, http://hdl.handle.net/10919/52860

37. Jonathan Paul Leidig, "Epidemiology Experimentation and Simulation Management through Scientific Digital Libraries" September 2012, Ph.D. dissertation, http://hdl.handle.net/10919/28759
38. Noha Ibrahim ElSherbiny, "Secure Digital Libraries", June 2011, MS thesis, http://hdl.handle.net/10919/33842
39. Uma Murthy, "Digital Libraries with Superimposed Information: Supporting Scholarly Tasks that Involve Fine Grain Information", April 2011, PhD dissertation, http://hdl.handle.net/10919/26866
40. Alexander Joel Dacara Alon, "The AlgoViz Project: Building an Algorithm Visualization Web Community", July 2010, MS thesis, http://hdl.handle.net/10919/34246
41. Logambigai Venkatachalam, "Scalability of Stepping Stones and Pathways", May 2008, MS thesis, http://hdl.handle.net/10919/32326
42. Seonho Kim, "Visualizing Users, User Communities, and Usage Trends in Complex Information Systems Using Implicit Rating Data", May 2008, PhD dissertation, http://hdl.handle.net/10919/27286
43. W. Ryan Richardson, "Using Concept Maps as a Tool for Cross-Language Relevance Determination", July 2007, PhD dissertation, http://hdl.handle.net/10919/28191
44. Douglas Gorton, "Practical Digital Library Generation into DSpace with the 5S Framework", April 2007, MS thesis, http://hdl.handle.net/10919/31914
45. Yuxin Chen, "A Novel Hybrid Focused Crawling Algorithm to Build Domain-Specific Collections", Feb. 2007, PhD dissertation, http://hdl.handle.net/10919/26220
46. Johnny L. Sam Rajkumar, "ETANA-CMV: A coordinated multiple view visual browsing interface for ETANA-DL", Dec. 2006, MS thesis, http://hdl.handle.net/10919/30817
47. Shahrooz Feizabadi, "Garbage Collection Scheduling for Utility Accrual Real-Time Systems", Dec. 2006, PhD dissertation, co-advisors Godmar Back and Edward A. Fox, http://hdl.handle.net/10919/30233
48. Vikram Raj Vidya Sagar, "A Digital Library Success Model for Computer Science Student Use of a Meta-Search System", Aug. 2006, MS thesis, http://hdl.handle.net/10919/30995
49. Baoping Zhang, "Intelligent Fusion of Evidence from Multiple Sources for Text Classification", June 2006, PhD dissertation, http://hdl.handle.net/10919/28198
50. Devdutta Bhosale, "AlcoZone: An Adaptive Hypermedia based Personalized Alcohol Education", May 2006, MS thesis, Fox as co-chair, committee chair Sandeep Shukla, http://hdl.handle.net/10919/32913
51. Rao Shen, "Applying the 5S Framework To Integrating Digital Libraries", April 2006, PhD dissertation, http://hdl.handle.net/10919/27099
52. Ananth Raghavan, "Schema Mapper: A Visualization Tool for Incremental Semi-automatic Mapping-based Integration of Heterogeneous Collections into Archaeological Digital Libraries: The ETANA-DL Case Study", May 2005, MS thesis, http://hdl.handle.net/10919/32950
53. Marcos Andre Goncalves, "Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications", Nov. 2004, PhD dissertation, http://hdl.handle.net/10919/29942
54. Fernando Adrian Das Neves, "Stepping Stones and Pathways: Improving Retrieval by Chains of Relationships between Documents", Sep. 2004, PhD dissertation, http://hdl.handle.net/10919/29419
55. Unnikrishnan Ravindranathan, "Prototyping Digital Libraries Handling Heterogeneous Data Sources - An ETANA-DL Case Study", April 2004, MS thesis, http://hdl.handle.net/10919/31938
56. Aaron Krowne, "An Architecture for Collaborative Math and Science Digital Libraries", July 2003, MS thesis, http://hdl.handle.net/10919/20137
57. Rohit Dilip Kelapure, "Scenario-Based Generation of Digital Library Services", June 2003, MS thesis, http://hdl.handle.net/10919/33634
58. Ye Zhou, "Reengineering PhysNet in the uPortal Framework", May 2003, MS thesis, http://hdl.handle.net/10919/33491
59. Kunal Garach, "CITIDEL Collection Building", May 2003, MS independent study report, Virginia Tech Dept. of Computer Science Technical Report TR-03-14, http://hdl.handle.net/10919/20130

60. Ganesh K. Panchanathan, "Digital library logging and analysis using XML", May 2003, MS independent study report, Virginia Tech Dept. of Computer Science Technical Report TR-03-17, http://hdl.handle.net/10919/20094
61. Aniket Prabhune, "XML for ETDs", Dec. 2002, MS independent study report, Virginia Tech Dept. of Computer Science Technical Report TR-02-28, http://hdl.handle.net/10919/20115
62. Hussein Suleman, "Open Digital Libraries", Nov. 2002, PhD dissertation, http://hdl.handle.net/10919/29712
63. Qinwei Zhu, "5SGraph: A Modeling Tool for Digital Libraries", Nov. 2002, MS thesis, http://hdl.handle.net/10919/35832
64. Srikanth Koneru, "Open Peer to Peer Technologies", Sept. 2002, MS independent study report, Virginia Tech Dept. of Computer Science Technical Report TR-02-09, http://hdl.handle.net/10919/20086
65. Li Wang, "Crawling on the World Wide Web", June 2002, MS independent study report, Virginia Tech Dept. of Computer Science Technical Report TR-02-10, http://hdl.handle.net/10919/20052
66. Jun Wang, "VIDI: A Lightweight Protocol Between Visualization Systems and Digital Libraries", May 2002, MS thesis, http://hdl.handle.net/10919/33845
67. Robert Karl France, "Effective, Efficient Retrieval in a Network of Digital Information Objects", Nov. 2001, PhD dissertation, http://hdl.handle.net/10919/29754
68. Ohm Sornil, "Parallel Inverted Index for Large-Scale, Dynamic Digital Libraries", Jan. 2001, PhD dissertation, http://hdl.handle.net/10919/26131
69. Jianxin Zhao, "Making Digital Libraries Flexible, Scalable and Reliable: Reengineering the MARIAN System in JAVA", June 1999, MS thesis, http://hdl.handle.net/10919/78146
70. Ghaleb Abdulla, "Analysis and Modeling of World Wide Web Traffic", May 1998, PhD dissertation, http://hdl.handle.net/10919/30470
71. Binzhang Liu, "Characterizing Web Response Time", April 1998, MS thesis, http://hdl.handle.net/10919/36741
72. David R. DeVaux, "A Tutorial on Authorware", April 1996, MS project report, http://hdl.handle.net/10919/37163
73. Lucio Cunha Tinoco, "Online Evaluation in WWW-based Courseware: The QUIZIT System", Jan. 1996, MS thesis, http://hdl.handle.net/10919/36881
74. Kaushal R. Dalal, "Database manager for Envision", Aug. 1994, MS thesis, http://hdl.handle.net/10919/42322
75. Sangita Betrabet, "A query language for information graphs", Dec. 1993, MS thesis, http://hdl.handle.net/10919/45365
76. Ghaleb Abdulla, "An image processing tool for cropping and enhancing images", Dec. 1993, MS thesis, http://hdl.handle.net/10919/46362
77. Amjad M. Daoud, "Efficient data structures for information retrieval", Aug. 1993, PhD dissertation, http://hdl.handle.net/10919/40031
78. Eskinder Sahle, "Development of a user interface for MARIAN and CODER systems", 1993, MS thesis, http://hdl.handle.net/10919/42318
79. Richard Dee Barnhart, "The naval message analyzer", Dec. 1992, MS thesis, http://hdl.handle.net/10919/41159
80. John D. Bozarth, "Requirements of the Navy's Tomahawk Theater Mission Planning system relating to object-oriented technology", Nov. 1992, MS thesis, http://hdl.handle.net/10919/41823
81. James Robert Johnson, "Interface design for an audio based information retrieval system", July 1992, MS thesis, http://hdl.handle.net/10919/42448
82. QiFan Chen, "An object-oriented database system for efficient information retrieval applications", March 1992, PhD dissertation, http://hdl.handle.net/10919/27976
83. Arun Narasimhan, "Design of the integrator to work with HyTime", 1992, MS thesis, http://hdl.handle.net/10919/42616
84. Sanjeev Datta, "Implementation of a Perfect Hash Function Scheme", March 1988, MS project report, released as Computer Science Technical Report TR 89-9, March 1989, Virginia Tech, Blacksburg, VA, http://hdl.handle.net/10919/19495
85. Marybeth Therese Weaver, "Implementing an intelligent information retrieval system: the CODER system, version 1.0", Feb. 1988, MS thesis, http://hdl.handle.net/10919/44097
86. Whay C. Lee, "Experimental comparison of schemes for interpreting Boolean queries", 1988, MS thesis, http://hdl.handle.net/10919/80010

87. R. C. Wohlwend and E. Fox, "Creation of a Prolog Fact Base from the Collins English Dictionary", M.S. report, TR-88-24, VPI&SU Comp.Sci. Dept., 1988, Blacksburg, VA, http://hdl.handle.net/10919/19817
88. Sheila G. Winett, "FOCES: An experimental expert system to select appropriate foster care homes for children", 1987, MS thesis, http://hdl.handle.net/10919/80055

Last updated 7/24/2020

Fox Patent Consulting Summary List – 8/2/2020 – CONFIDENTIAL

A. Cases with trial or deposition

Case: Impact Engine, Inc. v. Google LLC, C.A. No. 19-cv-1301 (S.D. Cal.)
Attorney: Antonio Sistos of Quinn Emanuel Urquhart Oliver & Hedges, LLP
Patents: 7,870,497; 8,356,253; 8,930,832; 9,361,632; 9,805,393; 10,068,253; 10,565,618; 10,572,898
Consulting starting June 2020, for: Defendant (i.e., Google)
Status: report filed 7/24/2020; deposition requested for August 2020

Case: Uniloc 2017 LLC v. Google LLC, transferred June 2020 to N. District Cal.
Attorney: Antonio Sistos of Quinn Emanuel Urquhart Oliver & Hedges, LLP
Patents: 6,366,908
Consulting starting March 2019, for: Defendant (i.e., Google)
Status: helped with preliminary invalidity contentions, non-infringement report due after transfer completed

Case: Arendi S.A.R.L. v. Google LLC, C.A. No. 13-319-LPS (D. Del.)
Motorola Mobility LLC f/k/a Motorola Mobility, Inc., C.A. No. 12-160-LPS (D. Del.)
Attorney: Robert Laurenzi of Paul Hastings LLP
Patents: 7,496,854 etc.
Consulting starting March 2019, for: Defendants (i.e., Google, Motorola)
Status: declaration in June 2019 led to some claims being dropped; invalidity report due 8/7/2020 with deposition to follow

Case: Cherwell Software, LLC v. BMC Software, Inc., No. 1:17-cv-01399 (D. Colo.); BMC Software, Inc. v. Cherwell Software, LLC et al., No. 2:17-cv-374 (E.D. Tex.); BMC Software, Inc. v. Cherwell Software, LLC et al., No. 3:17-cv-01369 (N.D. Tex.), BMC Software, Inc. v. Cherwell Software, LLC, No. 1:17-cv-03127-MSK-STV (D. Colo.)
Attorney: Sharif Jacob of Keker, Van Nest & Peters LLP
Patents: 8,082,222 and 9,239,857
Consulting starting Aug. 2017, for: Defendants (i.e., Cherwell)
Status: advising regarding invalidity and non-infringement and IPR; settled

Case: USDC, Eastern District of Texas, Tyler Div., C.A. No. 6:15-cv-01039-RWS; Eolas Technologies Inc. vs. Google, Inc.; transferred to N.D. Cal., Case No. 3:2017cv01138
Attorney: David Perlson of Quinn Emanuel Urquhart Oliver & Hedges, LLP
Patent: US 9,195,507
Consulting starting Jan. 2017, for: Defendants (Google)
Trial: Judge Robert W. Schroeder III in E.D. Texas, transferred to Jon S. Tigar in N.D. Cal.
Status: non-infringement report due probably by Nov. 2020

Case: USDC, District of Delaware, C.A. No. 09-525 (LPS);
Personalized User Model, LLP vs. Google, Inc.
Attorney: David Perlson of Quinn Emanuel Urquhart Oliver & Hedges, LLP
Patents: US 6,981,040 and 7,685,276
Consulting starting Aug. 2010, for: Defendants (on non-infringement, including expert report, deposition completed Nov. 20, 2012, and testifying in trial)
Trial: Judge Stark, Wilmington DE, 3/10/2014-3/20/2014
Status: defendant not infringing, patents invalid, plaintiff guilty in counter-suit

Case: Personalized Media Communications v. Zynga, Case No. 2:12-cv-68-JRG
Zynga Attorneys: Jones Day: Louis Touton, Krista Schwartz, David Wu
Other Attorneys: Akin Gump Strauss Hauer & Feld: Fred Williams
Consulting starting March 2012, for: Defendants (two invalidity reports filed, offer of proof filed for possible bench trial)
Trial: Judge Payne, Marshall, TX during Nov. 12-15, 2013
(attended for invalidity, not argued)
Status: defendant not infringing; case closed

Case: Bright Response, LLC v. Google, Inc., et al., also called
POLARIS IP, LLC v. GOOGLE, INC., et al.
Civil Case No. 2:07-CV-00371-CE
Attorney: David Perlson of Quinn Emanuel Urquhart Oliver & Hedges, LLP
Consulting for Defendants: Google, AOL (including expert report, deposition, and serving as testifying expert on non-infringement in trial)
Status: 2010 jury trial completed with Judge Charles Everingham IV presiding, Eastern District of Texas, Marshall Division; litigation concluded; patent invalid and defendants not infringing

Case: Constellation IP, LLC v. Avis Budget Group, Inc., et al., Civil Action No. 5:07-cv-00038-LED-CMC
Attorney: Jeffrey Berkowitz with
Finnegan, Henderson, Farabow, Garrett & Dunner (representing FedEx)
11955 Freedom Drive, Reston, VA 20190 USA
Patent: 6,453,302
Consulting for: Defendants (on invalidity, including expert report and deposition)
Status: settled in 2008

Case: USDC, Eastern District of Texas, Tyler Division, 6:06-CV-208-LED, Forgent Networks, Inc. vs. Echostar Communications Corp., et al. (including DIRECTV, Time Warner Cable, CoxCom, Comcast, ...)
Attorneys: Louis Touton with Jones Day, and team with Morrison & Foerster LLP (Charles Barquist), Alston & Bird LLP, Duane Morris LLP
Patent: US 6,285,746
Consulting starting 2006 for: Defendants (including expert report, deposition, and serving as testifying expert witness in May 2007 trial, on invalidity)

Status: jury trial with Judge Davis completed and litigation concluded; patent invalid and defendants not infringing

B. Patents

Co-inventor on: US patent no. 7346621, filed May 14, 2004, issued March 18, 2008, Method and System for Ranking Objects Based on Intra-type and Inter-type Relationships (Benyu Zhang, Hua-Jun-Zeng, Wei-Ying Ma, Wensi Xi, Zheng Chen, and Edward A. Fox)

Co-inventor on: US Provisional Patent Application 62/945,202; filed December 8, 2019; Methods and Systems for Generating Declarative Statements Given Documents with Questions and Answers (Edward A. Fox, Saurabh Chakravarty, Raja Venkata Satya Phanindra Chava, and Maanav Mehrotra)

Co-inventor on: US Provisional Patent Application 63/039,725; filed June 16, 2020; Methods and Systems for Generating Summaries Given Documents with Questions and Answers (Edward A. Fox, Saurabh Chakravarty, Maanav Mehrotra, Satvik Chekuri, and Aarohi Sumant)

C. Inter partes review declarations

Volkswagen Group of America, Inc. submitting IPR on U.S. Pat. No. 6,408,296 assigned to Sound View Innovations, LLC
Working with: Sterne, Kessler, Goldstein & Fox LLC (Ryan C. Richardson, Todd Thurheimer)
Consulting starting: March 2019
Status: drafted declaration related to petition; case no longer active (settled)

Comcast submitting IPRs on U.S. Pat. Nos. 7,895,218, 8,122,034, and 8,433,696; then on 7,779,011 and 9,668,014
assigned to Veveo, Inc., a wholly-owned entity of Rovi
Working with: Banner & Witcoff, Ltd. (Matthew P. Becker, Michael Cuviello, et al.)
Consulting starting: August 2016
Status: (a) Six declarations (2 per patent) filed early 2017, institution ordered on '034 and '218, deposition for each of those, supplemental reports for each of those, 8,433,696 claims unpatentable in Final Written Decision; (b) 3 declarations filed for '011 and '394 in Nov. 2018; (c) 4 declarations filed for '014 in Jan. 2019; (d) deposition taken for '011/'394 on 8/26 and for '014 on 10/14/2019; (e) reply declarations for '011/'394 and '014 filed in Feb. 2020; (f) deposition for '011/'394 on 2/28/2020; (g) '014 determining 7/21/2020 by PTAB: all challenged claims unpatentable

Oracle Corporation v. Selene Communication Technologies LLC,

Inter Partes Review of US Patent 6,363,377; aka Thomson Reuters v. Selene IPR2014-01411
Working with: James M. Heintz, Timothy Lohse, and others from DLA Piper, as part of the joint defense group
Consulting starting: June 2014
Status: Declaration submitted August 2014, instituted on most claims, supplementary declaration filed 3/6/2015, case settled with dismissal 6/10/2015

King & Spalding LLP, starting April 2013,
(Software Rights Archive, LLC v. LinkedIn Corporation,
No. 12-cv-03971-RMW (N.D. Cal.);
Software Rights Archive, LLC v. Twitter, Inc.,
No. 12-cv-03972-RMW (N.D. Cal.))
Worked with K&S staff: Ketan Pastakia, . . .
Consulting on invalidity
Shift: defense shifted to Keker & Van Nest, LLP, in June 2013; also Cooley LLP
Worked with David Silbert, Philip Tassin, Sharif Jacob; Heidi Keefe, Lowell Mead
Activities: Declaration submitted July 29, 2013 in support of
Keker & Van Nest, LLP (KVN) filing of Inter Partes Review of
U.S. Patent Nos. 5,544,352 and 5,832,494 and 6,233,571; deposition 8/5 – 8/6/2014; second declaration filed 9/5/2014
Status: IPRs generally successful (Final Written Decision 35 U.S.C. 318(a) and 37 C.F.R. 42.73 on Case IPR2013-00478 on Patent 5,544,352 ordered claims 26, 28-30, 32, 34, 39 to be unpatentable; Final Written Decision 35 U.S.C. 318(a) and 37 C.F.R. 42.73 on Case IPR2013-00479 on Patent 5,832,494 ordered claims 18-20, 45, 48, 49, 51, 54 to be unpatentable; Final Written Decision 35 U.S.C. 318(a) and 37 C.F.R. 42.73 on Case IPR2013-00481 on Patent 6,233,571 ordered claims 12 and 22 to be unpatentable); decision sustained by Fed. Circuit

Clear With Computers, LLC v. Hyundai Motor America, Inc.
C.A. No. 12-cv-00077-LED (E.D. Tex.)
Consulting starting 2012 for: defendant Hyundai Motor America, Inc.
Attorneys: Winston & Strawn LLP
Completed: 2 reports to accompany IPR petitions filed early 2013
Status: settled

D. Some other cases

Impact Engine, Inc. v. Google LLC, No. 19-cv-1301 (S.D. Cal.)
Retained by: Quinn Emanuel Urquhart & Sullivan, LLP
Patents: 7,870,497; 8,356,253; 8,930,832; 9,361,632; 9,805,393; 10,068,253; 10,565,618; 10,572,898
Consulting starting June 2020 for: defendant Google
Status: Ongoing work

IXI Mobile and IXI IP v. BlackBerry Ltd. and Corp. et al.; Case No. 2:15-cv-01883-JRG-RSP, filed in the US District Court for the Eastern District of Texas, Marshall Division
Retained by: Quinn Emanuel Urquhart & Sullivan, LLP
Patent: 7,552,124
Consulting starting July 2016 for: defendant Google
Status: Ongoing work; case stayed Oct. 2016 pending IPR

Case: Word to Info, Inc. v. Facebook, Inc. et al.; Civil Action No. 3:14-CV-4387-K, in Northern District of Texas, Dallas Division; transferred to 3:15-cv-03485-WHO in N. District of California, Judge William H Orrick
Patents: 5,715,468; 6,138,087; 6,609,091; 7,349,840; 7,873,509; 8,326,603; 8,688,436
Attorney: Lowell Mead of Cooley, LLP
Consulting starting: July 2015, representing the defendant (on invalidity, claim construction, etc.)
Working with: Cooley LLP: Lowell D. Mead
Status: WTI agreed to stipulate to non-infringement under the Court's constructions (11/16/2016). The case is pending on appeal at the Federal Circuit.

Brite Smart Corp. v. Google Inc., Case 2:14-cv-760-JRG-RSP, filed in the US District Court for the Eastern District of Texas, Marshall Division
Serving as: expert consultant for defendant since June 2015
Retained by: Quinn Emanuel Urquhart & Sullivan, LLP
Patents: 7,249,104; 7,953,667; 8,326,763; 8,671,057
Status: Work completed June 2016

Dragon Intellectual Property, LLC v. DIRECTV, LLC, Case No. 1:13-cv-02065-RGA, in the United States District Court for the District of Delaware
Attorneys: Akin Gump Strauss Hauer & Feld LLP: John Wittenzellner
Patent: 5,930,444
Consulting starting June 2015 for: defendant DIRECTV
Activities: Work completed Sept. 2015

Custom Media Technologies LLC v. DIRECTV, Case No. 1:13-cv-01423-LPS
Attorneys: Akin Gump Strauss Hauer & Feld LLP: Brock F. Wilson, Kevin G. McBride
Patent: 6,269,275
Consulting starting 2015 for: defendant DIRECTV
Activities: Declaration filed regarding claim construction

PMC v. Amazon.com, Inc. and Amazon Web Services, LLC, Case No. 1:13-cv-1608-RGA, in the United States District Court for the District of Delaware
Attorneys: Knobbe, Martens, Olson & Bear, L.L.P.: Colin Heideman
Patents: 5,887,243; 7,801,304; 7,805,749; 8,046,791; 7,864,956; 7,827,587;

7,783,252
Consulting starting 2015 for: defendant Amazon
Status: Defendant won case, but that was appealed

TLI Communications LLC v. AV Automotive, LLC, Max Media LLC, and Google Inc., Civil Action No. 1:14-CV-00139(TSE/IDD), filed in the US District Court for the Eastern District of Virginia, Alexandria Division
Serving as: expert consultant for defendants since May 2014
Retained by: Williams and Connolly LLP
Status: Court dismissed case on 101 and 112 grounds on 2/9/2015

Rockstar Consortium US LP et al. v. Google Inc., Case 2:13-cv-00893, filed in the US District Court for the Eastern District of Texas, Marshall Division
Serving as: expert consultant for defendant since May 2014
Retained by: Quinn Emanuel Urquhart & Sullivan, LLP
Status: Plaintiff transferred patents so case over in late Dec. 2014

Spark Networks USA, LLC v. Humor Rainbow, Inc. et al., Case No. 2:11-cv-01430 (C.D. Cal.)
Serving as expert consultant in 2011 for defendants:
- Match.com, represented by Quinn Emanuel Urquhart & Sullivan, LLP
  -- working with Mark Baker
- Zoosk, Inc., represented by Wilson Sonsini Goodrich & Rosati
  -- working with Stefani E. Shanberg

Status: settled

Content Delivery Solutions LLC v. Akamai Techs., Inc., et al., Case No. 1:11-cv-00216-LY, United States District Court for the Western District of Texas (Austin Division)
Consulting in 2011, retained by:
Wilson Sonsini Goodrich & Rosati ("WSGR"), attorneys for Google Inc.
Status: settled

King & Spalding LLP, starting Sept. 2010, for Google, Software Rights Archive, LLC v. Google Inc. et al, No. 07-cv-0511 (E.D. Tex.); Google Inc. et al v. L. Daniel Egger et al, No. 5:08-cv-3172 (N.D. Cal.)
Worked with: Jennifer J. Schmidt, Alex Skucas
Consulting on claim construction and invalidity through Sept. 2011
Status: settled

Quinn Emanuel Urquhart Oliver & Hedges, LLP, San Francisco, CA, representing Google starting Jan. 2011, for
Xerox Corporation v. Google Inc. et al., Case No. 10-136-LPS [District of Delaware] (Another document shows 10-136-JJF-MPT)
Status: settled

Quinn Emanuel Urquhart Oliver & Hedges, LLP, San Francisco, CA,
representing IAC Search & Media, Inc.
Attorneys: Mark Baker, Kristin Madigan
starting 2008, settled by 2010, for
Software Rights Archive v. Google et al., No. 07-cv-0511 (E.D. Tex.)

PA ADVISORS, LLC v. GOOGLE INC., ET AL.,
CASE NO. 2-07-CV-480-RRR (E.D. Tex., Marshall Division)
Retained by Quinn Emanuel Urquhart Oliver & Hedges, LLP, counsel
for Defendant Google Inc. ("Google")
Attorney: Brian Cannon
Status: report on invalidity prepared 1/4/2010,
summary judgment ordered 3/11/2010

Case: USDC, Eastern District of Texas, Beaumont Division, Civil Action
No. 1:05-CV-0264; Finisar Corporation v. The DIRECTV Group, Inc.;
DIRECTV Holdings, LLC; DIRECTV Enterprises, LLC; DIRECTV Operations,
LLC; DIRECTV, Inc.; and Hughes Network Systems, Inc.
Attorney: Louis Touton with Jones Day
Patent: US 5,404,505
Consulting starting 2008 for: Defendants (including expert report)
Status: Judge Ron Clark granted defendants' motion for summary
judgment on 5/19/2009

Irell & Manella LLP for
USDC, Eastern District of Virginia (Alexandria Division);
2:07CV03; Lycos Inc. vs. Tivo, Inc, Netflix, Inc. and Blockbuster, Inc.
Consulting for: Defendents (TiVo)
Agreement: 8/2007 (retained but no work requested)

Morrison & Foerster LLP for
US District Court, Southern District of Florida, Case
07-20624-CIV, Nissim Corp. vs. Time Warner Inc. et al.
Consulting for: Defendants
Agreement: 6/2007; my work completed in 2007

Covington & Burling LLP for
potential IP dispute between Netratings Inc. and comScore Networks, Inc.
Patents: US 6,115,680 and 5,675,510
Consulting for: comScore Networks, Inc.
Agreement: 7/2006 (retained but no work requested)

E. Yet other experience

Case: USDC (C.D. Cal.) No. CV 04-7456-JFW; Altnet, Inc. et al. v. Recording Industry Association of America, et al.
Attorney: Gibson, Dunn & Crutcher LLP
Patents: US 5,978,791 and 6,415,280
Consulting for: Defendants (RIAA)
Agreement: 3/2005

Case: USDC (Western District of Virginia) No. 7:04-00628; DE Technologies, Inc. vs. Dell Inc.
Attorney: Merchant & Gould
Patents: US 6,460,020 and 6,845,364
Consulting for: Plaintiff
Agreement: 2/2005

Case: USDC (N.D. Cal.) Case Nos. 04-2927 JW and 04-2928 JW; Teleshuttle Technologies Inc. et al. v. Microsoft Corporation
Attorney: Klarquist Sparkman, LLP
Patents: US 6,125,388; 6,611,862; and 6,658,464; 6,557,054 and 6,769,009
Consulting for: Defendent
Agreement: 2/2005

Case: USDC (C.D. CA) No. 00-13378 SVW (RNBx); NetZero, Inc. v. Juno Online Services, Inc.
Attorney: Kenyon & Kenyon
Patent: US 6,157,946
Consulting for: Defendent
Agreement: 1/2001

Case: C.A. No. 00-073-SLR (D. Del.) filed 2/4/2000; IBM Corp. v. Informix Corp.
Attorney: Cravath, Swaine & Moore
Patents: US 5,995,974; 5,978,793; 5,873,074; and 5,701,453
Consulting for: Plaintiff
Agreement: 4/2000

In the 1970s I gave a deposition as a employee and fact witness on a case involving Vulcraft, Division of NUCOR Corporation. I do not remember further details, except that there was some discussion of a competitor getting listings of our computer programs from the trash and stealing secrets of our methods; I ran the computer operations at the plant in Florence, SC.

Exhibit D

Claim Chart Applying CyberDesk Against the 7,917,843 Patent

CyberDesk was known and/or publicly used in the United States at least by 1997 as evidenced by the publications below. It therefore constitutes prior art under at least pre-AIA 35 U.S.C. §§ 102(a), 102(b), and 103(a). As shown below, CyberDesk anticipates and/or renders obvious claims 1, 8, 13, 15, 17, 18, 19, 23, and 30 of the '843 patent, at least as CyberDesk was known, used, and described in the following publications:

- Dey, Anind et al., *CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services*, Technical Report, GVU Center, Georgia Institute of Technology, GIT-GVU-97-10, June 1997 ("CyberDesk Technical Report");
- Dey, Anind et al., *CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services*, UIST 97, ACM 0-89791-881-9/97/10 ("UIST Article");
- Wood, Andrew et al., *CyberDesk: Automated Integration of Desktop and Network Services*, CHI 97, Atlanta GA, Mar. 22-27, 1997, ACM 0-89791-802-9/97/03 ("CyberDesk Technical Note").
- Abowd, Gregory et al., *Context-Awareness in Wearable and Ubiquitous Computing*, GVU Technical Report, GIT-GVU-97-11, Submitted to the 1st International Symposium on Wearable Computers, 1997, Abbreviated Version to be Presented as a Poster. ("Context-Awareness Technical Report")
- Dey, Anind et al., *CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services*, to appear in proceedings of IUI '98. ("IUI Proceedings")
- Dey, Anind et al., *CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services*, Knowledge-Based Systems, Volume 11, Received May 19, 1998; Accepted June 1, 1998. ("Knowledge-Based Systems")
- Dey, Anind et al., *Context-Aware Computing: The CyberDesk Project*; AAAI 1998 Spring Symposium, Stanford University, March 23-25, 1998. ("AAAI Symposium")
- "Ubiquitous Computing: Extending Access To Mobile Data", Michael David Pinkerton, May 1997, Georgia Institute of Technology, MS thesis ("Pinkerton Thesis"), with accompanying thesis notes ("Pinkerton Thesis Notes")

- "Context-Awareness in Wearable and Ubiquitous Computing," which was published in 1999 in the Virtual Reality Society International Journal 3 (1999) ("VRSIJ Article")
- Context-Awareness in Wearable and Ubiquitous Computing," which was presented at the Proceedings of 1st International Symposium on Wearable Computers in October, 1997 ("CyberDesk Poster")

CyberDesk was further described in the November 12, 2019 deposition of Anind Dey ("Dey Depo.").

CyberDesk, including its integration with Newton devices, was further described in a March 20, 2019 discussion with Mike Pinkerton and myself.

"Obviousness Statement" - To the extent that the Judge or Jury finds that CyberDesk does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my

Exhibit D

Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, CyberDesk discloses the preamble. See, e.g., CyberDesk Technical Report at 1, col. 1: "Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviours and leave little flexibility to the user. In this paper, we discuss CyberDesk, a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., CyberDesk Technical Report at 1, col. 1: "In response, software companies have been adopting the notion of component software: using small software modules as building blocks for a larger application. While there are many competing standards (OLE [11], Active X [10], Java Beans [6], OpenDoc [1]), the prevailing view is to provide a framework which programmers and sophisticated users can build upon to create desired application suites." See, e.g., CyberDesk Technical Report at 1, col. 2: "In this paper, we present CyberDesk system, a component software framework that relieves most of the burden of integrating services from both the designer of individual services and the end user, provides greater flexibility to the user, and automatically suggests how independent services can be integrated in interesting ways." See, e.g., CyberDesk Technical Report at 1, col. 2: "CyberDesk is a component-based framework written in Java, that supports automatic integration of desktop and network services [16]. The framework is flexible, and can be easily customized and extended." See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he

Exhibit D

	wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at 4, cols. 1-2: "All of the desktop applets currently being used in CyberDesk (2 e-mail browsers, contact manager, 2 calendar managers/schedulers, scratchpad) were previously written by other Georgia Tech students." See, e.g., CyberDesk Technical Report at 5, col. 2—6, col. 1 (including fig. 3): "The example below is the wrapper for the Contact Manager (see Figure 3), and it extends the ContactApplet class (the original application class). . . . Lookup an entry for the name in the ContactManager." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in

Exhibit D

	it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stored on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes it simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the

Exhibit D

	WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Website address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map

Exhibit D

	services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See., e.g. VRSIF Article at 5: "The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and event sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the event sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. These five components are discussed in greater detail elsewhere [6]." CyberDesk Poster at 2-3: "In the CyberDesk project [3], we use information context to aid in the integration of user services. Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that a user is attending to. Examples of user services are an e-mail browser, Web-based map service, or contact manager. CyberDesk uses informational context to change the set of resident services and offer relevant suggestions to the user. For example, as shown in Figure 2, a user can be reading an e-mail message which has information (Web address and e-mail address) on a new book written by a favorite author. The user highlights the e-mail address (a), explicitly announcing the context. The system gives him some

Exhibit D

	suggestions (b) on what he can do: search on the author's name, save the contact information, call the author, or send an e-mail. The user chooses the first two options (c and d) and saves the e-mail." See, e.g., Pinkerton Thesis at 19-20: "The second application extends CyberDesk [8], a separate FCE research project for allowing users to access information through simple 'agents' which search multiple information spaces (the Internet, local data, etc.) based on context. The CyberDesk infrastructure simplifies the automatic integration of applications by providing services (network or local) utilized by applications in the CyberDesk environment without the application itself having to be explicitly aware of them. We have added several services to CyberDesk which locate and present mobile information (such as that on a Newton) to the user, accessible by clicking a single button." See, e.g., Pinkerton Thesis at 29-32: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9).

Exhibit D

	Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it. What do the above applications demonstrate? Most importantly, they demonstrate how powerful removing the barrier between mobile information and the rest of a user's information can be. Once a user's computing environment is opened up to include all the information to which they have physical access, they are empowered to use all the tools and devices available to complete their task, whether it is making a phone call to ta colleague or writing a summary of their last meeting. Nobody doubts the overall gains which integrating Internet information into our desktops has provided, and mobile devices, which hold important information not available on the Internet, are no different." See, e.g., Pinkerton Thesis at Figs. 8-9:

Exhibit D

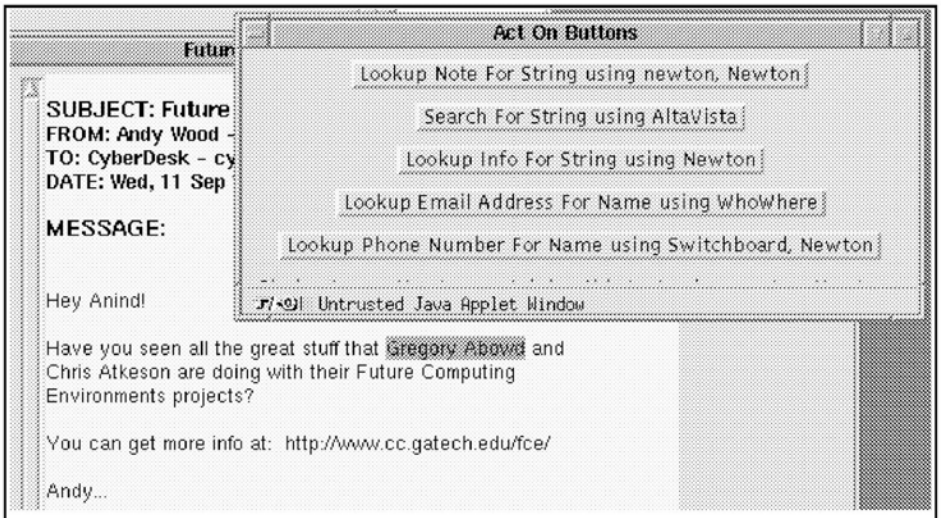


Figure 8 - CyberDesk ActOn window

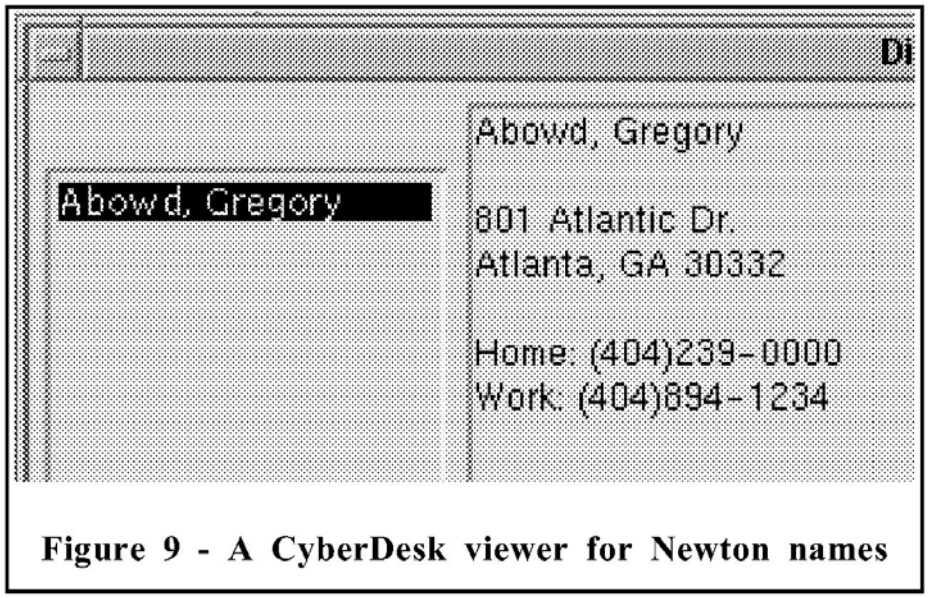


Figure 9 - A CyberDesk viewer for Newton names

See, e.g., Pinkerton Thesis at 84-85:

"In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add teh following:

- Extending the components to allow modifying mobile information and then storing it back to the Newton.
- Allow users to generate content from scratch on the desktop side and save it on the Newton.

Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names'

Exhibit D

	information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton. ● An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier. ● CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services." 'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9. 'And the information that's being displayed includes the name Gregory D. Abowd, correct? A. Correct. Q. And it also includes a mailing address that's in Atlanta, Georgia, correct? A. Correct. Q. And it also includes a phone number? A. Correct." Dey Depo. at 61:17-25. 'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information." They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would

Exhibit D

	suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. 'So the e-mail address abowd@cc.gatech.edu and the sane Gregory D. Abowd would get picked up, the name CyberDesk would get picked and the e-mail address cyberdesk@cc.gatech.edu would get picked up. This date and timestamp will be picked up as a date, and then John Doe would be picked up as a name, the URL, newbooks.com/DOE would get picked up as a URL, the e-mail address DOE@newbooks.com would get picked up. And then all the rest of the content would just be picked up as text." Dey Depo. at 86:2-11. 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'if John Doe existed in the contact book, it would populate the rest of the interface with information about John doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. 'Q. In the CyberDesk system that you were describing, was the intent to have the ability for the user actually to make a phone call? A. Yes." Dey Depo. at 92:6-9. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. 'So these were fully -- full-fledged, real applications built in Java. So that's a calendaring system, and e-mail system, a second e-mail system, and a contact manager. And so those were applications where I was able

Exhibit D

	to modify the source code to essentially speak the CyberDesk internal language, and so that they could tell CyberDesk, "This is the kind of information I can consume and run services on." Dey Depo. at 148:9-16. 'And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. 'So that menu of the ActOn buttons would always be displayed? A. That's correct. Q. Okay. Then the window in the top left, this shows an e-mail, correct? A. Correct. Q. Where is the content that's being displayed in this e-mail coming from? A. This was coming from my -- my e-mail -- the e-mail server that Georgia Tech ran. Q. And in terms of a user who's using CyberDesk at this time, what e-mail – how would this e-mail content pop up into the CyberDesk window? A. So there was no -- there's no CyberDesk window. So just – the – when you went to the CyberDesk Web page, it would -- it would launch an e-mail tool. We're seeing one window of that e-mail tool, so you'll -- you'll -- the window that we're not seeing the e-mail tool would be, you know, the sort of subject line and the date and the to and the from, it's the sort of the summary information. Q. Okay. A. If you double=clicked on this, on one of those messages ,this is what you would get. So that's what's being shown here. Q. Okay. So this is -- the e-mail tool would pop up and it would be like an e-mail inbox? A. That's right. Q. Okay. And then if a user selected an e-mail within that inbox -- A. They'd see the message, yeah. Q. They'd see the message. Okay. And then if the user highlighted a word within this -- within that window -- A. Yes. Q. -- then CyberDesk would recognize that the user had highlighted information?

Exhibit D

	A. That's correct. Q. Okay. Did CyberDesk analyze any information other than the highlighted content? MR. HICKS: Objection. Form. A. Not in the version that was published at the CHI 1997. Q. (BY MR. DIEHL) Okay. A. Venue. Q. Okay. Did it, in a subsequent version, recognize information in the e-mail other than the highlighted content? A. Yes -- yes, it did. Q. Which version was that? A. What we've been referring to so far as the second version of CyberDesk. Q. Okay. Okay. In this version that did not incorporate that aspect, if the user highlighted information, would CyberDesk analyze the highlighted span of text for data structures within that highlighted span? A. That's correct." Dey Depo. at 194:9-196:16. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	CyberDesk discloses this element. See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up

Exhibit D

	the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some

Exhibit D

	suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Website address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057.

Exhibit D

"After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105.

See, e.g., Pinkerton Thesis at Fig. 8:

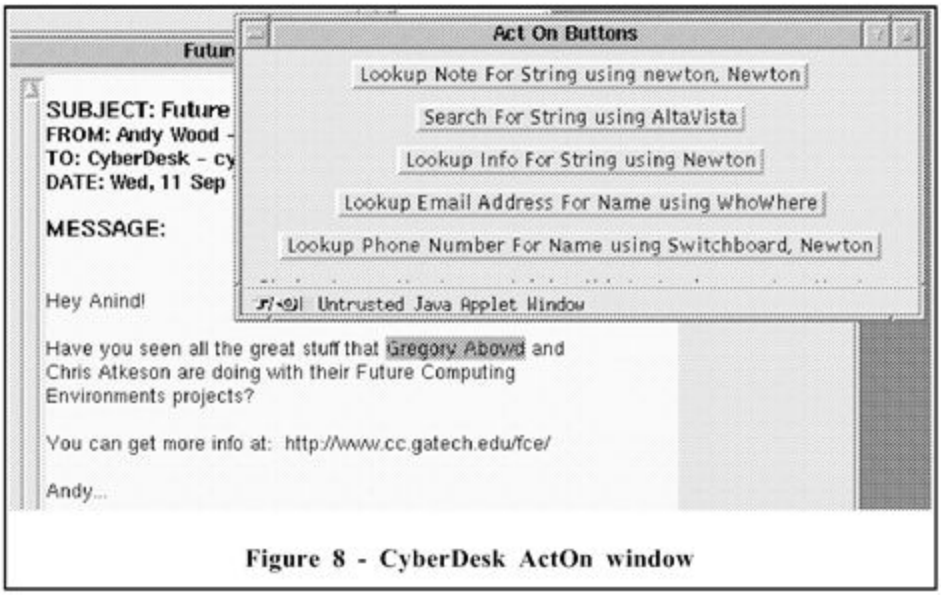


Figure 8 - CyberDesk ActOn window

'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information."

They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader.

Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type.

So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was

Exhibit D

	--pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. "So that menu of the ActOn buttons would always be displayed? A. That's correct. Q. Okay. Then the window in the top left, this shows an e-mail, correct? A. Correct. Q. Where is the content that's being displayed in this e-mail coming from? A. This was coming from my -- my e-mail -- the e-mail server that Georgia Tech ran. Q. And in terms of a user who's using CyberDesk at this time, what e-mail – how would this e-mail content pop up into the CyberDesk window? A. So there was no -- there's no CyberDesk window. So just – the – when you went to the CyberDesk Web page, it would -- it would launch an e-mail tool. We're seeing one window of that e-mail tool, so you'll -- you'll -- the window that we're not seeing the e-mail tool would be, you know, the sort of subject line and the date and the to and the from, it's the sort of the summary information. Q. Okay. A. If you double=clicked on this, on one of those messages ,this is what you would get. So that's what's being shown here. Q. Okay. So this is -- the e-mail tool would pop up and it would be like an e-mail inbox? A. That's right. Q. Okay. And then if a user selected an e-mail within that inbox -- A. They'd see the message, yeah." Dey Depo. at 194:9-195:14. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer	CyberDesk discloses this element.

Exhibit D

process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	See, e.g., CyberDesk Technical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 4, col. 2: "Data typing is used extensively in the interface declarations of the event sources and sinks that applications provide. The property field that corresponds to each interface declares the datatype/event that a component is interested in or can provide. The CyberDesk system takes advantage of the Java type system to do the data typing." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CyberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to convert the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the

Exhibit D

	IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 5, col. 1: "Currently the list of CyberDesk types include: Date PhoneNumber, Mailing Address, Name, URL, and EmailAddress. If any of the conversions can be made, then the converter generates a second, but related, selection event containing the newly typed data and sends it to observing entities." See, e.g., CyberDesk Technical Report at 5, cols. 1-2: "The CyberDesk framework was designed to be easily extensible. Simple extensions to CyberDesk include adding additional types, type converters, desktop services and network services. The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behaviour of CyberDesk to individual use and creating more interesting integrating behaviour." See, e.g., CyberDesk Technical Report at 5, col. 2: "An example converter is the StringToEmailAddress converter, which is a subclass of the ConversionApplet class. The code for this component and all components described in the paper can be viewed at http://www.cc.gatech.edu/fce/cyberdesk/samples. This converter looks at traditional ways of writing an e-mail address, and tries to map selected data to one of these ways. If it is successful, it returns an EmailAddress object. The ConversionApplet object is responsible for handling the ties to the CyberDesk framework." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Note at 553, col. 1 (including fig. 3): "When the user selects information displayed by one service, say some

Exhibit D

	text from the e-mail message, the type converters try recursively to see if the data can be converted to other types used in the system (e.g., a name in Figure 3). In the case of plain text, this could be done by comparing the string to common formats for representing the various types; for names you might use title firstname lastname, and similar patterns can be used for dates, URLs, e-mail and mailing addresses." See, e.g., CyberDesk Technical Report at 8, cols. 1-2: "CyberDesk contains some simple notions of context. It knows the application a user is working with and the data (both type and content) the user is interested in (via explicit selection with the mouse). But CyberDesk has shown the potential for supporting higher level context. For example, if an e-mail message contains information about a meeting, and the user selects the message content, a type converter could potentially convert the text to a Meeting object to be inserted in the user's Calendar Manager. Of course, retrieving context from arbitrary text is a very difficult problem being investigated by the AI learning community. But the power of CyberDesk supports the ability to use this higher level context, if available." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered dtatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them

Exhibit D

	back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or

Exhibit D

	send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk,

Exhibit D

	search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 5: "Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures." See, e.g., Pinkerton Thesis at 29-30:

Exhibit D

"Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services."

See, e.g., Pinkerton Thesis at Fig. 8:

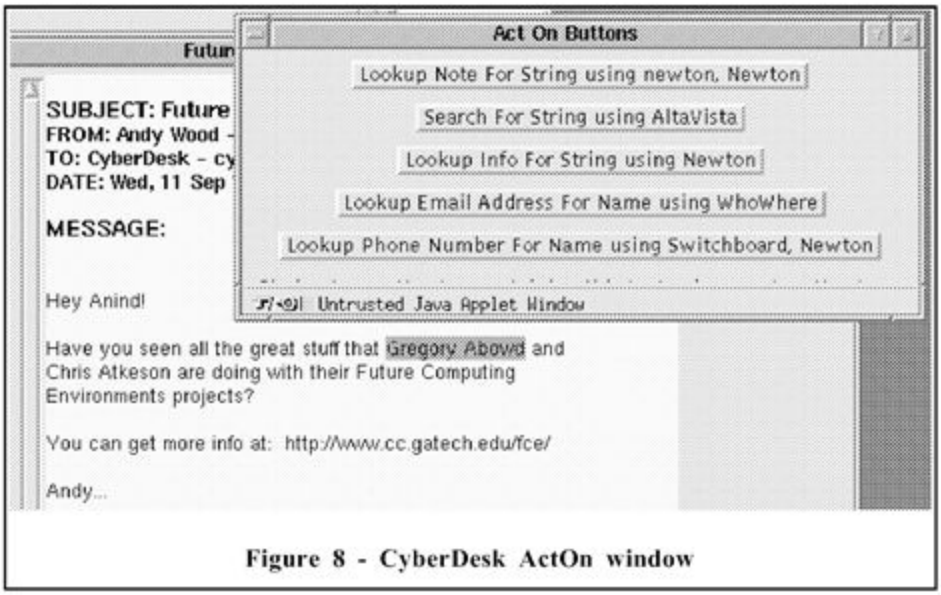


Figure 8 - CyberDesk ActOn window

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
"manipulate( ) called when user clicks button
- get data that is selected
- grab data
- create
* * *
- once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email).
- conversion process creates the new object and creates the button
- when user clicks button, calls objects manipulate( )"

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024:

Exhibit D

	"Cyberdesk Activities * * * Each of these is a "service" that can make use of existing types - manipulate( ) routine to do the work" 'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information." They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. 'So the e-mail address abowd@cc.gatech.edu and the sane Gregory D. Abowd would get picked up, the name CyberDesk would get picked and the e-mail address cyberdesk@cc.gatech.edu would get picked up. This date and timestamp will be picked up as a date, and then John Doe would be picked up as a name, the URL, newbooks.com/DOE would get picked up as a URL, the e-mail address DOE@newbooks.com would get picked up. And then all the rest of the content would just be picked up as text." Dey Depo. at 86:2-11. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first

Exhibit D

	name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. 'So these were fully -- full-fledged, real applications built in Java. So that's a calendaring system, and e-mail system, a second e-mail system, and a contact manager. And so those were applications where I was able to modify the source code to essentially speak the CyberDesk internal language, and so that they could tell CyberDesk, "This is the kind of information I can consume and run services on." Dey Depo. at 148:9-16. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	CyberDesk discloses this element. See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)."

Exhibit D

	See, e.g., CyberDesk Technical Report at 3, col. 2: "The previous three components discussed provide the core functionality of CyberDesk. Regardless of what tools the user wants to use, these three components are required. The fourth type of component, desktop and network services, are the actual services the user wants to access. Desktop services include e-mail browsers, contact managers, and schedulers. Network services include web search engines, telephone directories, and map retrieval tools. . . . One of the services available in CyberDesk is a gateway to the AltaVista search engine available on the web." See, e.g., CyberDesk Technical Report at 4, col. 1: "The wrapper must have 'hooks' into the original application code to intercept and broadcast the appropriate data selection events (for the 'select' interfaces), and to execute a service on data passed to it (for the 'method' interfaces). At the time of developments, there were three ways to approach this problem for the 'select' interface. First, we could modify the original application's event processing loop to broadcast events in the CyberDesk fashion. Second, we could modify the original application code to make calls to a notification routine in the wrapper when data is selected. Third, we could rely on the original application have a suitable API for retrieving those events." See, e.g., CyberDesk Technical Report at 7, col. 1 (including fig. 4): "There were two reasons for integrating CyberDesk with LlamaShare. First, we wanted to illustrate the platform-neutrality and language-neutrality of the LlamaServer, which CyberDesk allows us to do. More importantly, however, CyberDesk's vision of ubiquitous information access was the deciding factor. While LlamaShare provides a concrete, visible object to represent the data on a mobile device, CyberDesk takes the approach that information is distributed throughout a rather nebulous information space (consisting of Internet, desktop, and mobile data) that can be retrieved at any moment depending on the context in which the user is currently working. This new metaphor of seamless integration between mobile data and Internet (remote) data was too good to pass up." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stored on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes it simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: • You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text

Exhibit D

	and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manager, search name on the WWW, etc. However, by using chaining, more powerful suggestions can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the

Exhibit D

	information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057.

Exhibit D

	"After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as

Exhibit D

input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures."

See, e.g., Pinkerton Thesis at 29-30:

"Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services."

See, e.g., Pinkerton Thesis at Fig. 8:

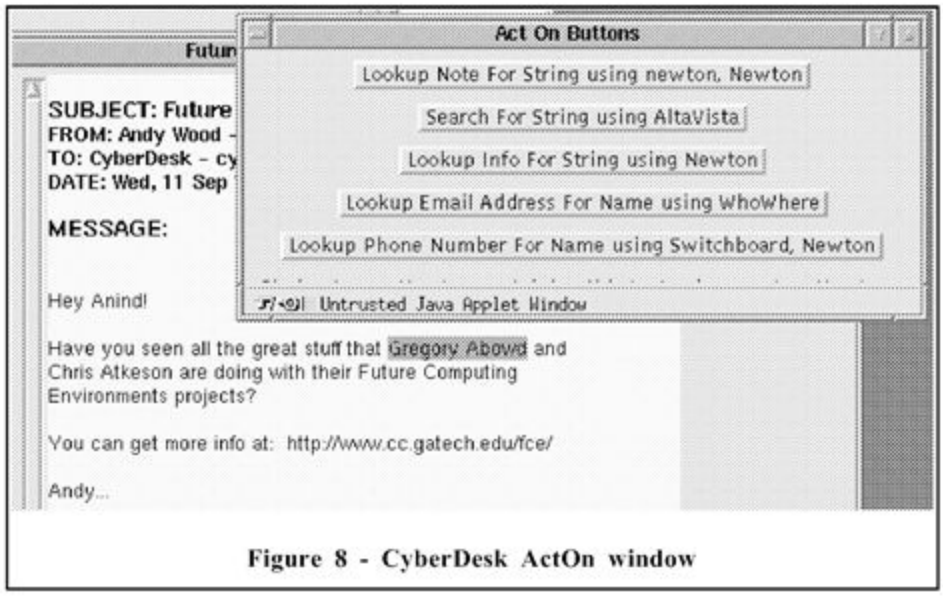


Figure 8 - CyberDesk ActOn window

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
"manipulate( ) called when user clicks button
- get data that is selected

Exhibit D

	- grab data - create" 'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9. 'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information." They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. 'So the e-mail address abowd@cc.gatech.edu and the sane Gregory D. Abowd would get picked up, the name CyberDesk would get picked and the e-mail address cyberdesk@cc.gatech.edu would get picked up. This date and timestamp will be picked up as a date, and then John Doe would be picked up as a name, the URL, newbooks.com/DOE would get picked up as a URL, the e-mail address DOE@newbooks.com would get picked up. And then all the rest of the content would just be picked up as text." Dey Depo. at 86:2-11. 'I don't know about the phone and the tax, because what you can see

Exhibit D

	also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	CyberDesk discloses this element. See, e.g., CyberDesk Technical Report at 1, col. 2: "The components in CyberDesk treat all data uniformly, regardless of whether the data came from a locally running application or from a service running on the World Wide Web (WWW). The services and applications themselves can be running anywhere, meeting CyberDesk's goal of providing ubiquitous access to services." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar,

Exhibit D

	the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 3, col. 2: "The ActOn Button Bar, as described before, is simply the user interface for the integrating IntelliButton. We chose to keep the interface separate from the actual integrating functionality to all easier experimentation with alternative interfaces. Currently, the interface is very simplistic. It is a dynamically generated list of buttons, where each button corresponds to a particular service that can be executed based on an user-generated event and its corresponding data. The list of buttons is provided by the IntelliButton. . . . For example: Send e-mail to this EmailAddress using Netscape; Search for a string on the Web using AltaVista."

Exhibit D

	See, e.g., CyberDesk Technical Report at 4, col. 1: "With this interface, this search would be suggested by the IntelliButton whenever a text string is the target of a selection (assuming the component in which the text selection is done, supports the 'select' interface)." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CYberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to conver the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 6, col. 2: "The following is an example network service: WhoWhereEmail. It is a gateway to the WhoWhere network service available on the web. When a name is input into the web service, a list of possible e-mail addresses corresponding to that name is returned. The CyberDesk network

Exhibit D

	service takes a Name object, inputs it into the service and displays the results in a web browser." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 9, col. 1: "Perhaps the biggest limitation of the system is the user interface implemented by the ActOn Button Bar. It consists of a window that displays a long list of suggested user actions. It is clear that the number of possible suggestions could quickly become overwhelming to the user. We are currently looking at different ways to adapt the interface to initially show actions that the user is likely to take, but provide a way for the user to see other possible actions as well. We are also looking at different presentation methods for the suggestions, including pop-up hierarchical menus and document lenses [4]." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered datatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See, e.g., CyberDesk Technical Report at 1, col. 2: "The components in CyberDesk treat all data uniformly, regardless of whether the data came from a locally running application or from a service running on the World Wide Web (WWW). The services and applications themselves can be running anywhere, meeting CyberDesk's goal of providing ubiquitous access to services." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a

Exhibit D

	separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them."

Exhibit D

	See, e.g., CyberDesk Technical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 3, col. 2: "The ActOn Button Bar, as described before, is simply the user interface for the integrating IntelliButton. We chose to keep the interface separate from the actual integrating functionality to all easier experimentation with alternative interfaces. Currently, the interface is very simplistic. It is a dynamically generated list of buttons, where each button corresponds to a particular service that can be executed based on an user-generated event and its corresponding data. The list of buttons is provided by the IntelliButton. . . . For example: Send e-mail to this EmailAddress using Netscape; Search for a string on the Web using AltaVista." See, e.g., CyberDesk Technical Report at 4, col. 1: "With this interface, this search would be suggested by the IntelliButton whenever a text string is the target of a selection (assuming the component in which the text selection is done, supports the 'select' interface)." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CYberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and

Exhibit D

	created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to convert the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 6, col. 2: "The following is an example network service: WhoWhereEmail. It is a gateway to the WhoWhere network service available on the web. When a name is input into the web service, a list of possible e-mail addresses corresponding to that name is returned. The CyberDesk network service takes a Name object, inputs it into the service and displays the results in a web browser." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 9, col. 1: "Perhaps the biggest limitation of the system is the user interface implemented by the ActOn Button Bar. It consists of a window that displays a long list of suggested user actions. It is clear that the number of possible suggestions could quickly become overwhelming to the user. We are currently looking at different ways to adapt the interface to initially show actions that the user is likely to take, but provide a way for the user to see other possible actions as well. We are also looking at different presentation methods for the suggestions, including pop-up hierarchical menus and document lenses [4]." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the

Exhibit D

	same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered datatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of

Exhibit D

	network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures." See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows

Exhibit D

	user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it." See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The

Exhibit D

LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following:

- Extending the components to allow modifying mobile information and then storing it back to the Newton.
- Allow users to generate content from scratch on the desktop side and save it on the Newton.

Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop:

- A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton.
- An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier.
- CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services."

See, e.g., Pinkerton Thesis at Figs. 8-9:

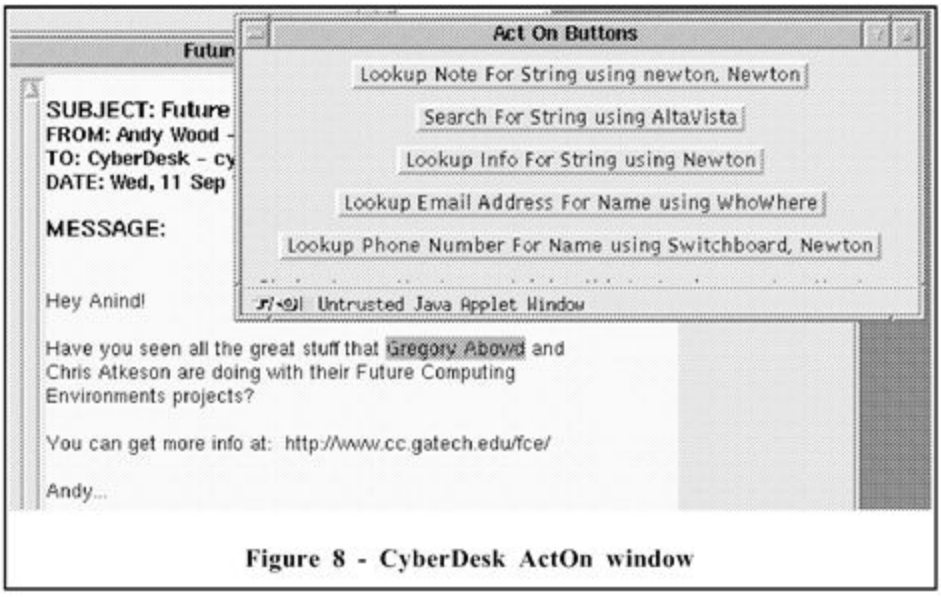


Figure 8 - CyberDesk ActOn window

Exhibit D

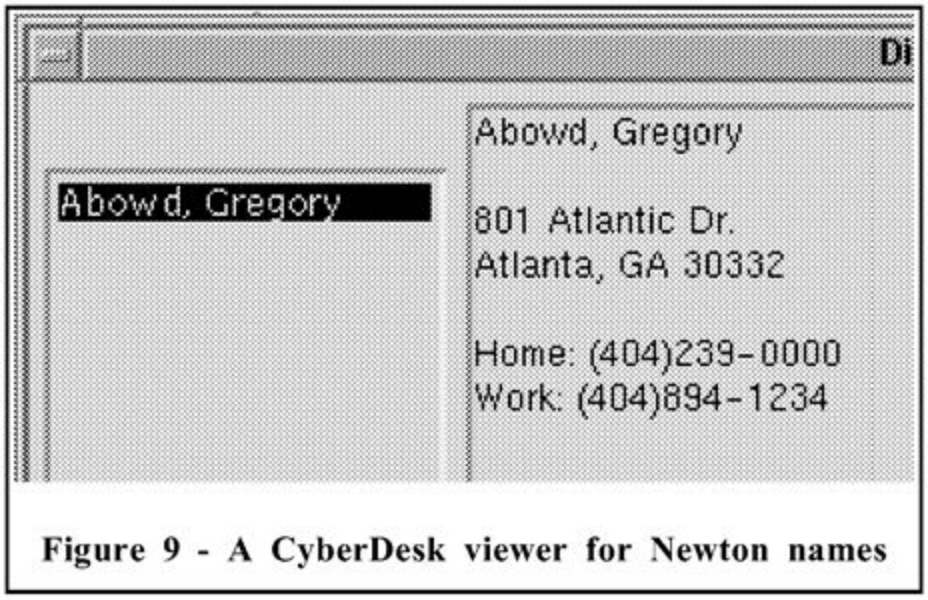


Figure 9 - A CyberDesk viewer for Newton names

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
"manipulate( ) called when user clicks button
- get data that is selected
- grab data
- create
* * *
- once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email).
- conversion process creates the new object and creates the button
- when user clicks button, calls objects manipulate( )"

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024:
"Cyberdesk Activities
* * *
Each of these is a "service" that can make use of existing types
- manipulate( ) routine to do the work"

'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9.

'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information."
They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader.

Exhibit D

	Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. 'So the e-mail address abowd@cc.gatech.edu and the sane Gregory D. Abowd would get picked up, the name CyberDesk would get picked and the e-mail address cyberdesk@cc.gatech.edu would get picked up. This date and timestamp will be picked up as a date, and then John Doe would be picked up as a name, the URL, newbooks.com/DOE would get picked up as a URL, the e-mail address DOE@newbooks.com would get picked up. And then all the rest of the content would just be picked up as text." Dey Depo. at 86:2-11. 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. 'Q. In the CyberDesk system that you were describing, was the intent to have the ability for the user actually to make a phone call? A. Yes." Dey Depo. at 92:6-9. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the

Exhibit D

	'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. 'And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	CyberDesk discloses this element. See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number

Exhibit D

	for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at 1, col. 2: "The components in CyberDesk treat all data uniformly, regardless of whether the data came from a locally running application or from a service running on the World Wide Web (WWW). The services and applications themselves can be running anywhere, meeting CyberDesk's goal of providing ubiquitous access to services." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves,

Exhibit D

	and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 3, col. 2: "The ActOn Button Bar, as described before, is simply the user interface for the integrating IntelliButton. We chose to keep the interface separate from the actual integrating functionality to all easier experimentation with alternative interfaces. Currently, the interface is very simplistic. It is a dynamically generated list of buttons, where each button corresponds to a particular service that can be executed based on an user-generated event and its corresponding data. The list of buttons is provided by the IntelliButton. . . . For example: Send e-mail to this EmailAddress using Netscape; Search for a string on the Web using AltaVista." See, e.g., CyberDesk Technical Report at 4, col. 1: "With this interface, this search would be suggested by the IntelliButton whenever a text string is the target of a selection (assuming the component in which the text selection is done, supports the 'select' interface)." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For

Exhibit D

	example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CyberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to convert the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 6, col. 2: "The following is an example network service: WhoWhereEmail. It is a gateway to the WhoWhere network service available on the web. When a name is input into the web service, a list of possible e-mail addresses corresponding to that name is returned. The CyberDesk network service takes a Name object, inputs it into the service and displays the results in a web browser." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes

Exhibit D

	from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 9, col. 1: "Perhaps the biggest limitation of the system is the user interface implemented by the ActOn Button Bar. It consists of a window that displays a long list of suggested user actions. It is clear that the number of possible suggestions could quickly become overwhelming to the user. We are currently looking at different ways to adapt the interface to initially show actions that the user is likely to take, but provide a way for the user to see other possible actions as well. We are also looking at different presentation methods for the suggestions, including pop-up hierarchical menus and document lenses [4]." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered datatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stored on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that

Exhibit D

	number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manager, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Website address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can

Exhibit D

	be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured

Exhibit D

	context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures."

Exhibit D

	See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as

Exhibit D

looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the infomation's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it."

See, e.g., Pinkerton Thesis at Figs. 8-9:

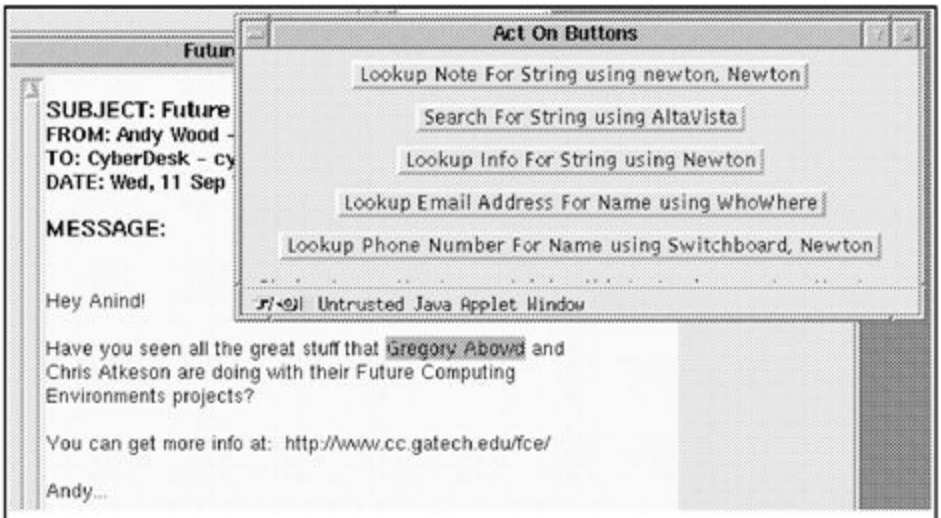


Figure 8 - CyberDesk ActOn window

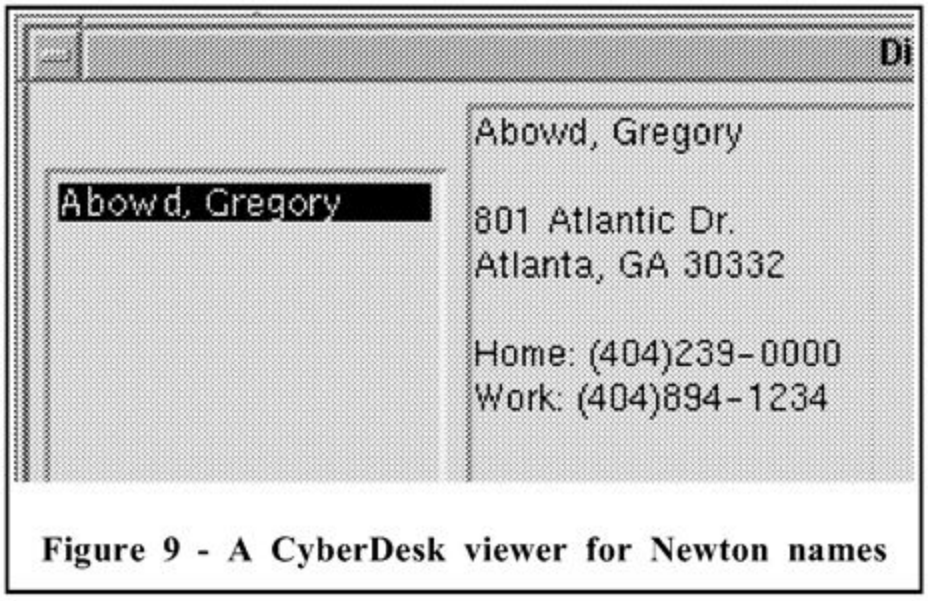


Figure 9 - A CyberDesk viewer for Newton names

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023: "manipulate( ) called when user clicks button

Exhibit D

	- get data that is selected - grab data - create * * * - once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email). - conversion process creates the new object and creates the button - when user clicks button, calls objects manipulate( )" See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024: "Cyberdesk Activities * * * Each of these is a "service" that can make use of existing types - manipulate( ) routine to do the work" 'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information." They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. "I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8.

Exhibit D

	'if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. "Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	CyberDesk discloses this element. See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact

Exhibit D

	information in the contact manager (d)." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are

Exhibit D

	displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CYberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Tehnical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 4, col. 2: "Data typing is used extensively in the interface declarations of the event sources and sinks that applications provide. The property field that corresponds to each interface declares the datatype/event that a component is interested in or can provide. The CyberDesk system takes advantage of the Java type system to do the data typing."

Exhibit D

	See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CYberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to conver the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 5, col. 1: "Currently the list of CyberDesk types include: Date PhoneNumber, Mailing Address, Name, URL, and EmailAddress. If any of the conversions can be made, then the converter generates a second, but related, selection event containing the newly typed data and sends it to observing entities." See, e.g., CyberDesk Technical Report at 5, cols. 1-2: "The CyberDesk framework was designed to be easily extensible. Simple extensions to CyberDesk include adding additional types, type converters, desktop services and network services. The real advantages with CyberDesk can be seen with more complex extensions that include adapting the

Exhibit D

	behaviour of CyberDesk to individual use and creating more interesting integrating behaviour." See, e.g., CyberDesk Technical Report at 5, col. 2: "An example converter is the StringToEmailAddress converter, which is a subclass of the ConversionApplet class. The code for this component and all components described in the paper can be viewed at http://www.cc.gatech.edu/fce/cyberdesk/samples. This converter looks at traditional ways of writing an e-mail address, and tries to map selected data to one of these ways. If it is successful, it returns an EmailAddress object. The ConversionApplet object is responsible for handling the ties to the CyberDesk framework." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Note at 553, col. 1 (including fig. 3): "When the user selects information displayed by one service, say some text from the e-mail message, the type converters try recursively to see if the data can be converted to other types used in the system (e.g., a name in Figure 3). In the case of plain text, this could be done by comparing the string to common formats for representing the various types; for names you might use title firstname lastname, and similar patterns can be used for dates, URLs, e-mail and mailing addresses." See, e.g., CyberDesk Technical Report at 8, cols. 1-2: "CyberDesk contains some simple notions of context. It knows the application a user is working with and the data (both type and content) the user is interested in (via explicit selection with the mouse). But CyberDesk has shown the potential for supporting higher level context. For example, if an e-mail message contains information about a meeting, and the user selects the message content, a type converter could potentially convert the text to a Meeting object to be inserted in the user's Calendar Manager. Of course, retrieving context from arbitrary text is a very difficult problem being investigated by the AI learning community. But the power of CyberDesk supports the ability to use this higher level

Exhibit D

	context, if available." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered datatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Tehnical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CYberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection

Exhibit D

	events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to convert the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stored on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called

Exhibit D

	while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at

Exhibit D

	ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage

Exhibit D

	of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of

Exhibit D

	information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it." See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components

Exhibit D

used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following:

- Extending the components to allow modifying mobile information and then storing it back to the Newton.
- Allow users to generate content from scratch on the desktop side and save it on the Newton.

Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop:

- A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton.
- An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier.
- CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services."

See, e.g., Pinkerton Thesis at Figs. 8-9:

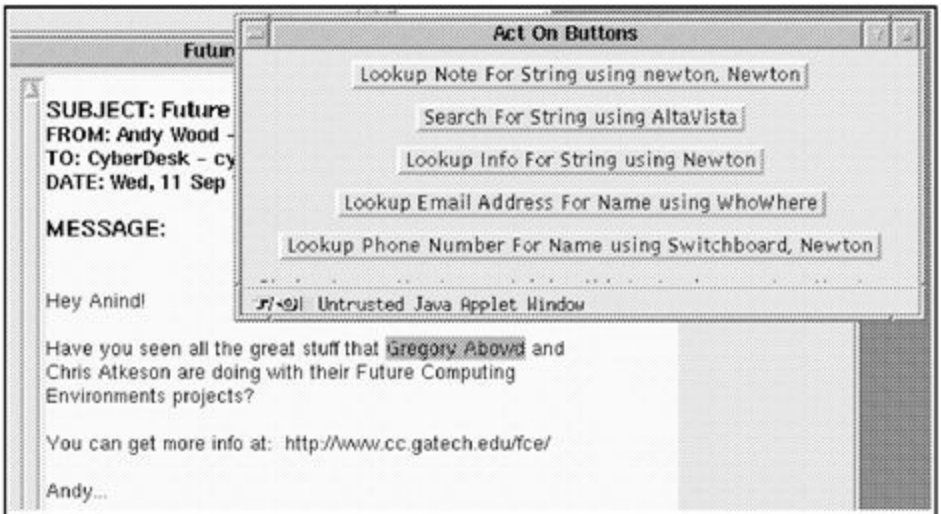


Figure 8 - CyberDesk ActOn window

Exhibit D

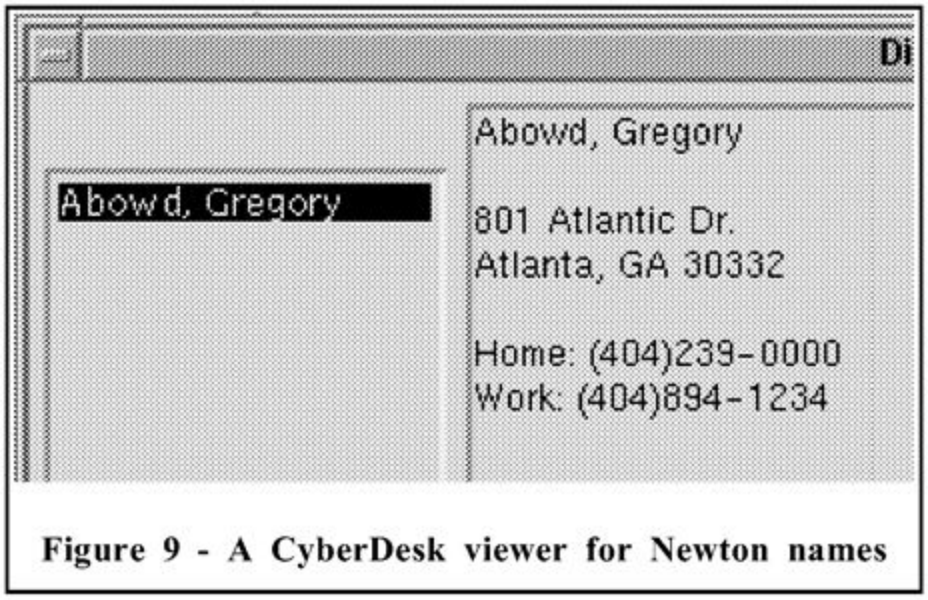


Figure 9 - A CyberDesk viewer for Newton names

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
"manipulate( ) called when user clicks button
- get data that is selected
- grab data
- create
* * *
- once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email).
- conversion process creates the new object and creates the button
- when user clicks button, calls objects manipulate( )"

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024:
"Cyberdesk Activities
* * *
Each of these is a "service" that can make use of existing types
- manipulate( ) routine to do the work"

'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9.

'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information."
They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader.

Exhibit D

	Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. "I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. 'Q. In the CyberDesk system that you were describing, was the intent to have the ability for the user actually to make a phone call? A. Yes." Dey Depo. at 92:6-9. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. 'And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct.

Exhibit D

	Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element. See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Report at 8, col. 1: "Along the same line of thought, chaining can be used along with the concept of 'combining' to make services more powerful. . . . Combining in

Exhibit D

	CyberDesk terms, is the ability to collect multiple data types for a piece of selected data, and bind them together, as needed, to create multiple meta-objects. These meta-objects could be used to perform substantially more powerful actions. Taking the above example of a user reading an appointment in her calendar, assume that there are multiple services capable of chaining, so a URL selection event is created, a Date selection event is created, a MailingAddress selection event is created, etc. There is potentially enough information to create a new entry in the Contact Manager. The ability to assimilate this widespread information in a compact entity is very powerful for a user. It's this same ability that would allow us to combine time and a name to search a friend's schedule as seen in the user scenario." See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., CyberDesk Technical Report at 4, cols. 1-2: "All of the desktop applets currently being used in CyberDesk (2 e-mail browsers, contact manager, 2 calendar managers/schedulers, scratchpad) were previously written by other Georgia Tech students." See, e.g., CyberDesk Technical Report at 5, col. 2—6, col. 1 (including fig. 3): "The example below is the wrapper for the Contact Manager (see Figure 3), and it extends the ContactApplet class (the original application class). . . . Lookup an entry for the name in the ContactManager." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)."

Exhibit D

	See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 7, col. 2—8, col. 1: "The extensions described so far are fairly simplistic. They deal with adding more suggestions for the user. More interesting are extensions which add more powerful suggestions for the user. Chaining is an example of this type of extension. It extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behaviour. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented with a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manager, search name on the WWW, etc. However, by using chaining, more powerful suggestions can be had. The WhoWhereEmail network described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one) that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the

Exhibit D

	message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVista, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is given a name, it returns a phone number and mailing address. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105.

Exhibit D

	See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following: ● Extending the components to allow modifying mobile information and then storing it back to the Newton. ● Allow users to generate content from scratch on the desktop side and save it on the Newton. Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton. ● An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier. ● CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services." 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 5. For example, CyberDesk and the prior art references and systems set forth in Exhibit U, Table 5 (e.g., Miller, Pandit) relate to managing information in an address book/contact database. A POSITA would have reasonable expectation of success to achieve predictable results in combining CyberDesk and the prior art references and systems in Exhibit U. A POSITA would have recognized advantages and benefits to have CyberDesk provide a prompt for updating the information source to include the first information.
Claim 13	

Exhibit D

A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element. See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at 2, cols. 1-2 (including figures): "The CyberDesk system has a simple but innovative architecture. It is based on an event-driven model, where components act as even sources and/or event sinks. Events, in this current version, are generated from explicit user interaction with the system. The system consists of five core components: the Locator, the IntelliButton, the ActOn Button Bar, the desktop and network services, and the type converters. The Locator maintains the registry of event sources and sinks. This allows the IntelliButton to automatically find matches between event sources and even sinks based on a given input event, a task normally required of the system or service designer. The IntelliButton displays the matches in the form of suggestions to the user, via the ActOn Button Bar. It is through the ActOn Button Bar that the user accesses the integrating functionality of CyberDesk. The services are the even sources and sinks themselves, and are the tools the user ultimately wants to use. The type converters provide more powerful integrating behavior by converting given events into other events, allowing for a greater number of matches. The five components are discussed in greater detail below." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call

Exhibit D

	upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Note at 553, cols. 1-2: "Finally, the user's selection, plus any extra type information generated by the conversion process, is observed by the ActOn integrating applet and a set of potential actions for that data is suggested. For example, a name is accepted as input by the Switchboard service, and so ActOn creates a button that suggests that integrating behaviour. Clicking on the button invokes the Switchboard service, completing the integration without requiring any change to the functionality of either service and without an programming effort from the user." See, e.g., CyberDesk Technical Report at 3, col. 2: "The ActOn Button Bar, as described before, is simply the user interface for the integrating IntelliButton. We chose to keep the interface separate from the actual integrating functionality to all easier experimentation with alternative interfaces. Currently, the interface is very simplistic. It is a dynamically generated list of buttons, where each button corresponds to a particular service that can be executed based on an user-generated event and its corresponding data. The list of buttons is provided by the IntelliButton. . . . For example: Send e-mail to this EmailAddress using Netscape; Search for a string on the Web using AltaVista." See, e.g., CyberDesk Technical Report at 4, col. 1: "With this interface, this search would be suggested by the IntelliButton whenever a text string is the target of a selection (assuming the component in which the text selection is done, supports the 'select' interface)." See, e.g., CyberDesk Technical Report at 4, col. 2: "Initially, we hardcoded applications to generate events for different data types. For example, the e-mail browser declares that it can generate String selection events when text is highlighted, but also EmailAddress selection events when the 'To:' or 'From:' field in an e-mail message is selected. When EmailAddress selection events were generated, they were passed through the CyberDesk system, as described before, to the ActOn Button Bar, which displayed services that could consume EmailAddress selection events (e.g., Send an E-mail to this E-mail Address using Netscape). . . . Consequently, we chose to use type converters. Using simple heuristics, it is possible to identify potential text strings that might be e-mail addresses. It would have been desirable to

Exhibit D

	augment our e-mail browser with this capability, so that any time text was selected in it, it would try to convert the text to an EmailAddress object and create an EmailAddress selection event rather than just a String selection event. But, rather than just giving this type conversion capability to the e-mail browser, we wanted to add that ability to the system once, and all it to be used in every application where e-mail addresses might appear. We took the type detection ability out of the individual applications and created type converters, an independent and extensible layer in the architecture." See, e.g., CyberDesk Technical Report at 5, col. 1: "So, when any component generates a String selection even, the StringToEmailAddress converter (and any other observers) are notified, and the converter attempts to conver the given String object to an EmailAddress object (while other converters attempt to convert the object to another CyberDesk type). In the above scenario, this conversion was done when the user selected the e-mail address. The system initially saw the selected data as a String but with this converter, it also saw it as an EmailAddress. This results in two related data selection events to arrive at the IntelliButton: one containing a string and one containing an EmailAddress. The IntelliButton will therefore seek integrating behaviour for both these types, allowing the user to access EmailAddress-relevant services where originally they wouldn't have had the option." See, e.g., CyberDesk Technical Report at 6, col. 2: "The following is an example network service: WhoWhereEmail. It is a gateway to the WhoWhere network service available on the web. When a name is input into the web service, a list of possible e-mail addresses corresponding to that name is returned. The CyberDesk network service takes a Name object, inputs it into the service and displays the results in a web browser." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 9, col. 1: "Perhaps the biggest limitation of the system is the user interface implemented by the ActOn Button Bar. It consists of a window that displays a long list of suggested user actions. It is clear that the number of possible suggestions could quickly become overwhelming to the user. We are currently looking at different ways to adapt the interface to initially show

Exhibit D

	actions that the user is likely to take, but provide a way for the user to see other possible actions as well. We are also looking at different presentation methods for the suggestions, including pop-up hierarchical menus and document lenses [4]." See, e.g., CyberDesk Technical Report at 9, col. 2: "Related Work. Pandit and Kalbags Selection Recognition Agent [12] attempts to address the same issues as CyberDesk. Unlike CyberDesk, it uses a fixed datatype-action pair, allowing for only one possible action for each datatype recognized. . . . Apple Data Detectors [2] is another component architecture that supports automatic integration of tools. It works at the operating system level, using the selection mechanism that most Apple applications support. It allows the selection of a large area of text and recognizes all user-registered datatypes in that selection. Users view suggested actions by pressing a modifier key and the mouse button. Like CyberDesk, it supports an arbitrary number of actions for each datatype." See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The

Exhibit D

	typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures." See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of

Exhibit D

	the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it." See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023: "manipulate( ) called when user clicks button - get data that is selected - grab data - create * * * - once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email). - conversion process creates the new object and creates the button - when user clicks button, calls objects manipulate( )" See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024: "Cyberdesk Activities * * * Each of these is a "service" that can make use of existing types - manipulate( ) routine to do the work" 'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9. 'So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information."

Exhibit D

	They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element.

Exhibit D

instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021072 (incl. fig. 1.): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery store and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an e-mail saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail . . . which as information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), saves the e-mail, and heads home.)" See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications

Exhibit D

	that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., UIST Article at p. 76, col. 1: "Chaining creates more interesting suggestions for the user, as is shown below, the WhoWhere network service can take a name as input and return a WWW browser showing a list of possible e-mail addresses corresponding to that name. By making the assumption that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related e-mail address selection event, and the user is supplied with all possible suggestions for

Exhibit D

	both a name and e-mail address." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed,

Exhibit D

	highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7 (*e.g.*, Luciw 735, Siitonen, Domini, and Schabes). A POSITA would have reasonable expectation of success to achieve predictable results in combining the CyberDesk and the prior art references and systems in Exhibit U, Table 7. For example, a POSITA would have recognized advantages and benefits to have the CyberDesk display the plurality of distinct instances of second information from search results to enable user selection of one of them for use in performing the action.
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element. See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of

Exhibit D

	one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021072 (incl. fig. 1.): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery store and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an e-mail saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail . . . which as information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), saves the e-mail, and heads home.)" See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk:

Exhibit D

	● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., UIST Article at p. 76, col. 1: "Chaining creates more interesting suggestions for the user, as is shown below, the WhoWhere network service can take a name as input and return a WWW browser showing a list of possible e-mail addresses corresponding to that name. By making the assumption that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related e-mail address selection event, and the user is supplied with all possible suggestions for both a name and e-mail address."

Exhibit D

	See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that

Exhibit D

	reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from

Exhibit D

	problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., VRSIF article at 5: "When integrating the behavior of separate applications, the programmer or the user needs to transfer data from one application to another. The typical scenario, as shown above, takes part of the output from one application (the name in the e-mail browser in Figure 2) and uses it as input to another application. In CyberDesk, after the user selects some string in one application, all possible type interpretations of that string are generated and announced (e.g., name, date, friend, etc.). Any resident service which can accept the announced data then becomes available to the user via a simple menu button. The context-awareness provides the ability to change the set of resident services and frees the user from having to remember any integration procedures." See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services.

Exhibit D

	Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in order to access it." See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following: ● Extending the components to allow modifying mobile information and then storing it back to the Newton. ● Allow users to generate content from scratch on the desktop side and save it on the Newton. Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk

Exhibit D

could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton.

- An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier.
- CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services."

See, e.g., Pinkerton Thesis at Fig. 8:

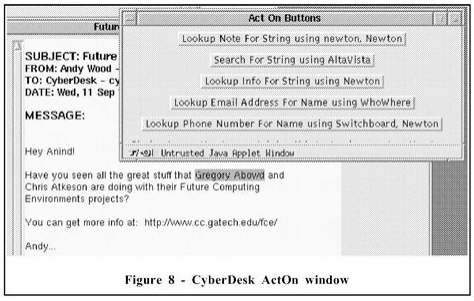


Figure 8 - CyberDesk ActOn window

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
" - once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email).
- conversion process creates the new object and creates the button
- when user clicks button, calls objects manipulate( )"

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024:
"Cyberdesk Activities
* * *
Each of these is a "service" that can make use of existing types
- manipulate( ) routine to do the work"

"I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8.

"if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6.

"Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]?
A. Yeah. So in this case, it would populate the -- it would populate the --

Exhibit D

	the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. "So these were fully -- full-fledged, real applications built in Java. So that's a calendaring system, and e-mail system, a second e-mail system, and a contact manager. And so those were applications where I was able to modify the source code to essentially speak the CyberDesk internal language, and so that they could tell CyberDesk, "This is the kind of information I can consume and run services on." Dey Depo. at 148:9-16. "And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element. See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021072 (incl. fig. 1.): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery store and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an e-mail saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail . . . which as information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the

Exhibit D

	system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), saves the e-mail, and heads home.)" See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption

Exhibit D

	(not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., UIST Article at p. 76, col. 1: "Chaining creates more interesting suggestions for the user, as is shown below, the WhoWhere network service can take a name as input and return a WWW browser showing a list of possible e-mail addresses corresponding to that name. By making the assumption that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related e-mail address selection event, and the user is supplied with all possible suggestions for both a name and e-mail address." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar."

Exhibit D

	See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a

	screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following: ● Extending the components to allow modifying mobile information and then storing it back to the Newton. ● Allow users to generate content from scratch on the desktop side and save it on the Newton. Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton. ● An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier. ● CyberDesk has many built-in types which we would like to

Exhibit D

	support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services." "Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9. "So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. "I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. "if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. "Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. "And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15.

Exhibit D

	For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	CyberDesk discloses claim 1. *See* claim 1 as charted above. CyberDesk further discloses this element. See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021072 (incl. fig. 1.): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery store and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an e-mail saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail . . . which as information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), saves the e-mail, and heads home.)" See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her

Exhibit D

	name to pull her name card from your Newton and display it on your workstation. • Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. • Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., UIST Article at p. 76, col. 1: "Chaining creates more interesting suggestions for the user, as is shown below, the WhoWhere network service can take a name as input and return a WWW browser showing a list of possible e-mail addresses corresponding to that name. By making the assumption that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related e-mail address selection event, and the user is supplied with all possible suggestions for both a name and e-mail address." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based

Exhibit D

	map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d), and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact

Exhibit D

	manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user."

Exhibit D

	See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following: ● Extending the components to allow modifying mobile information and then storing it back to the Newton. ● Allow users to generate content from scratch on the desktop side and save it on the Newton. Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton. ● An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier. ● CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services." 'Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9. 'So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. 'I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is

Exhibit D

	it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. 'if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. 'Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. 'And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	CyberDesk discloses this element. See, e.g., CyberDesk Technical Report at 1, col. 1: "Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviours and leave little flexibility to the user. In this paper, we discuss CyberDesk, a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user."

Exhibit D

	See, e.g., CyberDesk Technical Report at 1, col. 1: "In response, software companies have been adopting the notion of component software: using small software modules as building blocks for a larger application. While there are many competing standards (OLE [11], Active X [10], Java Beans [6], OpenDoc [1]), the prevailing view is to provide a framework which programmers and sophisticated users can build upon to create desired application suites." See, e.g., CyberDesk Technical Report at 1, col. 2: "In this paper, we present CyberDesk system, a component software framework that relieves most of the burden of integrating services from both the designer of individual services and the end user, provides greater flexibility to the user, and automatically suggests how independent services can be integrated in interesting ways." See, e.g., CyberDesk Technical Report at 1, col. 2: "CyberDesk is a component-based framework written in Java, that supports automatic integration of desktop and network services [16]. The framework is flexible, and can be easily customized and extended." See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact

Exhibit D

	information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at 4, cols. 1-2: "All of the desktop applets currently being used in CyberDesk (2 e-mail browsers, contact manager, 2 calendar managers/schedulers, scratchpad) were previously written by other Georgia Tech students." See, e.g., CyberDesk Technical Report at 5, col. 2—6, col. 1 (including fig. 3): "The example below is the wrapper for the Contact Manager (see Figure 3), and it extends the ContactApplet class (the original application class). . . . Lookup an entry for the name in the ContactManager." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)."' See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075: "A project currently under development in our research group is LlamaShare, an architecture and set of applications for providing users and programmers easy access to information stoed on mobile devices. There are two main goals for the Llamshare project, one of them coinciding with a goal of CyberDesk. The first is to create an infrastructure that makes is simple for programmers to take advantage of mobile data in their applications. The second is to provide applications that demonstrate ubiquitous access to information . . . Here are some examples of what we are doing with LlamaShare and CyberDesk: ● You're writing some e-mail and you know you scribbled down a note on your newton with some relevant text in it. Select text and search for all the notes on your Newton containing that text, then read them on your desktop machine. ● You get an e-mail from a colleague and you want to call them back. You have her phone number on your Newton. Select her

Exhibit D

	name to pull her name card from your Newton and display it on your workstation. ● Your secretary took down the number of someone who called while you were in a meeting, but you don't know who that number belongs to! Select the phone number and let the Newton search its names database and return the correct person or company matching that phone number. ● Your boss sends you e-mail asking you to schedule a meeting with two other people in your group. How do you find a time that everyone can meet? Select each of their names and let CyberDesk pull up their calendar's [sp] from their respective Newtons and display them. Schedule the meeting and make the change effective immediately in their Newton calendar." See, e.g., CyberDesk Technical Report at ARENDI-DEFS00021075-76: "Chaining . . . extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behavior. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manger, search name on the WWW, etc. However, by using chaining, more powerful suggestion can be had. The WhoWhereEmail network service described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one that the first e-mail address returns in the list is the correct one, we can now use the service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "In . . . the CyberDesk project, we use informational context to aid in the integration of desktop and network services Informational context refers to any artifact (e.g. words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map service, or a contact manager, interacting with one of these services might suggest some action to be invoked on another service. For example . . . the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Webs site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d),

	and saves the e-mail." See e.g., Context-Awareness Technical Report at ARENDI-DEFS00021061: "The real advantages with CyberDesk can be seen with more complex extensions that include adapting the behavior of CyberDesk to individual use and creating more interesting integrating behavior. Two examples we have implemented are chaining, in which the output of a service also acts as a type converter to announce higher level information to the system, and combining, using multiple pieces of information to invoke better services. For example, in chaining, a name selection might suggest an e-mail address lookup service. By executing this service automatically, the system can also suggest e-mail address-related services, such as searching newsgroups or sending an e-mail. In combining, using time with a name can allow the automatic search of a friend's calendar." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when

Exhibit D

	Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., VRSIJ article at 4-5: "In this second example, the CyberDesk project [18, 6], we use informational content to aid in the integration of desktop and network services. Informational context refers to any artifact (e.g., words on a screen or a picture at a museum) that the user is attending to. Examples of network and desktop services are an e-mail browser, a Web-based map services, or a contact manager. Interacting with one of these services might suggest some action to be invoked on another service. For example, as shown in Figure 2, the user can be reading an e-mail message which has information on a new book written by a favorite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a), thus explicitly announcing the information context, and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author. He chooses the first two options (c and d) and saves the e-mail. The behavior we are trying to provide in CyberDesk is *automatic* integration of separate services. Current software suites suffer from problems due to poor integration of their individual tools. They require the designer to think of all possible integrating behaviors and leave little flexibility to the user. CyberDesk is a component software framework that automatically integrates desktop and network services, requiring no integrating decisions to be made by the tool designers and giving total control to the user." See, e.g., Pinkerton Thesis at 29-31: "CyberDesk is a Java-based project developed by Wood, Dey, and Abowd at Georgia Tech [8]. One of the goals is to provide an architecture for application integration in which applications can automatically take advantage of the services provided by other applications in the environment. These services can make use of both local information, such as a calendar or a contact manager, or Internet information, such as searching for a phone number with SwitchBoard or

Exhibit D

	finding all web pages containing a text string with AltaVista. Unlike Cyberdog, which takes the approach of giving each piece of information a physical representation on the desktop, CyberDesk allows user to search a nebulous information pool of local and Internet information in response to queries based on the text selected in a CyberDesk-aware application, such as an email application. For example, when the user selects a person's name, they are provided the option to look up that person's phone number on SwitchBoard, find their email address on WhoWhatWhere, or perform a generic search for their name on AltaVista. The actions which a user may take are dependent on the type of data selected (name, URL, email address, etc.) and appear unobtrusively in a separate window called the "Act On" window, shown in Figure 8. In addition to searching the Internet, the user is given the option to display contact information from the locally-stored contact manager if a CyberDesk aware contact manager is also present. In this way, CyberDesk blurs the distinction between local and Internet information services. Since CyberDesk was such an open architecture, it was simple (an afternoon) to add a service which additionally searches a mobile device for the selected information. Two applets, one for displaying notes and another for displaying names, were written to allow users to browse information residing on a Newton (see Figure 9). Users now only need to learn one interaction model (select text, click ActOn button) to access information stored locally, on the Internet, or on a mobile device. This blurring of the boundaries, emphasized by the lack of physical representations which draw attention to the actual location of the data, simplifies the number of steps necessary to access information and makes the actual sources more ubiquitous to the user. Furthermore, since that information is retrieved based on the user's context during the completion of a task, it is seamlessly integrated into the task itself. In order to completely hide the location of the information from the user, we have proposed a change to the way CyberDesk creates the ActOn buttons (which is currently being implemented). In the existing system, CyberDesk creates one button in the ActOn window for every applicable service, even when different services produce similar results such as looking up a phone number. As a result, the user must manually choose between the actions based on where they think the information is located (for example, a local contact manager, the Internet service SwitchBoard, or the Newton). To achieve the goal of concealing the information's true location from the user, we propose to combine all action buttons which provide a similar service into a single button which, when clicked, searches all information spaces simultaneously. While the user may get multiple responses (another issue to be resolved in the future), they no longer need to be consciously aware of the location of the information in

Exhibit D

order to access it."

See, e.g., Pinkerton Thesis at 84-85:

"In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following:

- Extending the components to allow modifying mobile information and then storing it back to the Newton.
- Allow users to generate content from scratch on the desktop side and save it on the Newton.

Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop:

- A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton.
- An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier.
- CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services."

See, e.g., Pinkerton Thesis at Figs. 8-9:

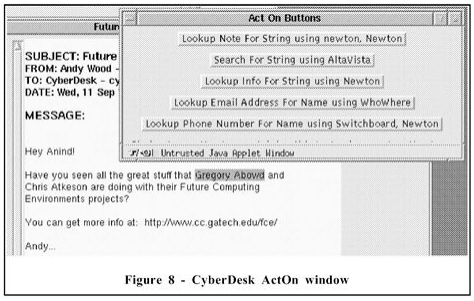


Figure 8 - CyberDesk ActOn window

Exhibit D

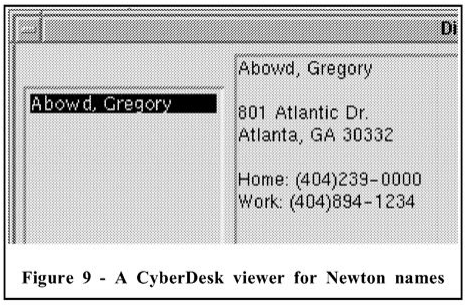


Figure 9 - A CyberDesk viewer for Newton names

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021023:
"manipulate( ) called when user clicks button
- get data that is selected
- grab data
- create
* * *
- once it finds a conversion that works (StringToemail), it goes over the list again looking for something that can take the new object (Email).
- conversion process creates the new object and creates the button
- when user clicks button, calls objects manipulate( )"

See, e.g., Pinkerton thesis notes at ARENDI-DEFS00021024:
"Cyberdesk Activities
* * *
Each of these is a "service" that can make use of existing types
- manipulate( ) routine to do the work"

"Together, we had this idea that we could build a system that could do what CyberDesk did, and so building off of his Cameo work, we put the first -- first version of CyberDesk together that allowed these different components to work together, one where you were able to select information from one application and have it be represented in some way in another application." Dey Depo. at 25:2-9.

"And the information that's being displayed includes the name Gregory D. Abowd, correct?

A. Correct.

Q. And it also includes a mailing address that's in Atlanta, Georgia, correct?

A. Correct.

Q. And it also includes a phone number?

A. Correct." Dey Depo. at 61:17-25.

Exhibit D

	"So within the – within the mail tool that's on the first page of Exhibit 8, you can have multiple different -- the user can select multiple different pieces of information." They can select someone's name, they can select text, they can select an e-mail address. And so first under the display portion of this figure, that's just information that's displayed in the personal information management tool. In this case, the mail reader. Once the user selects one of those, the CyberDesk system consumes that information, consumes what was selected and determines whether it can convert the chosen information to any other different type. So the two examples given here are: If an e-mail was selected, that e-mail address could be changed to a piece of text. And if a text was --pieces of text was created, it had the opportunity to be converted to a name. Based on the conversions that occurred, the ActOn button would suggest -- so now we're in the suggest portion of this image -- it would suggest, via the buttons that were -- that it was populated with, the kinds of actions that a user could perform. And then they would select one of those actions. And depending on the action that was selected, it would send the appropriate information to the appropriate service which was – is delineated quite well on -- on the buttons themselves." Dey Depo. at 65:18-66:21. "So you could have some -- a name that could go to -- sorry, a text that could be converted to a name, that name to a phone number, that phone number to an address, that address to and da-da-da-da, ad infinitum." Dey Depo. at 82:9-12. "So the e-mail address abowd@cc.gatech.edu and the sane Gregory D. Abowd would get picked up, the name CyberDesk would get picked and the e-mail address cyberdesk@cc.gatech.edu would get picked up. This date and timestamp will be picked up as a date, and then John Doe would be picked up as a name, the URL, newbooks.com/DOE would get picked up as a URL, the e-mail address DOE@newbooks.com would get picked up. And then all the rest of the content would just be picked up as text." Dey Depo. at 86:2-11.

Exhibit D

	"I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. "if John Doe existed in the contact book, it would populate the rest of the interface with information about John Doe and then it would add to it the information that CyberDesk had found." Dey Depo. at 90:2-6. "Q. In the CyberDesk system that you were describing, was the intent to have the ability for the user actually to make a phone call? A. Yes." Dey Depo. at 92:6-9. "Q. Again, can you describe for me what would happen if a user were to actually click ['Lookup Name Using Contact Manager]? A. Yeah. So in this case, it would populate the -- it would populate the -- the contact manager was already launched. It would populate that -- the 'find' text box with 'Wood, Andy,' because it went last name, first name. And then it would automatically search and then it populated the bottom-half or bottom two-thirds of the screen with information that you had about Andy Wood." Dey Depo. at 95:10-19. "So these were fully -- full-fledged, real applications built in Java. So that's a calendaring system, and e-mail system, a second e-mail system, and a contact manager. And so those were applications where I was able to modify the source code to essentially speak the CyberDesk internal language, and so that they could tell CyberDesk, "This is the kind of information I can consume and run services on." Dey Depo. at 148:9-16. "And so for example, the information that was in that contact book entry, the user, if they wanted to, could cut and paste that into the e-mail that they were writing . . . correct?" A. That is correct. Q. Did you ever come up with a version of CyberDesk that allowed the user actually to automatically insert any of the text that was found by

Exhibit D

	CyberDesk? A. Into an e-mail address you're asking me? Q. [] Yes, into a working document. A. Oh, sorry. Into a working document. Absolutely. Absolutely. The -- the example that comes to my mind is a text editor." Dey Depo. at 175:23-176:15. "So that menu of the ActOn buttons would always be displayed? A. That's correct. Q. Okay. Then the window in the top left, this shows an e-mail, correct? A. Correct. Q. Where is the content that's being displayed in this e-mail coming from? A. This was coming from my -- my e-mail -- the e-mail server that Georgia Tech ran. Q. And in terms of a user who's using CyberDesk at this time, what e-mail – how would this e-mail content pop up into the CyberDesk window? A. So there was no -- there's no CyberDesk window. So just – the – when you went to the CyberDesk Web page, it would -- it would launch an e-mail tool. We're seeing one window of that e-mail tool, so you'll -- you'll -- the window that we're not seeing the e-mail tool would be, you know, the sort of subject line and the date and the to and the from, it's the sort of the summary information. Q. Okay. A. If you double=clicked on this, on one of those messages ,this is what you would get. So that's what's being shown here. Q. Okay. So this is -- the e-mail tool would pop up and it would be like an e-mail inbox? A. That's right. Q. Okay. And then if a user selected an e-mail within that inbox --

Exhibit D

	A. They'd see the message, yeah. Q. They'd see the message. Okay. And then if the user highlighted a word within this -- within that window -- A. Yes. Q. -- then CyberDesk would recognize that the user had highlighted information? A. That's correct. Q. Okay. Did CyberDesk analyze any information other than the highlighted content? MR. HICKS: Objection. Form. A. Not in the version that was published at the CHI 1997. Q. (BY MR. DIEHL) Okay. A. Venue. Q. Okay. Did it, in a subsequent version, recognize information in the e-mail other than the highlighted content? A. Yes -- yes, it did. Q. Which version was that? A. What we've been referring to so far as the second version of CyberDesk. Q. Okay. Okay. In this version that did not incorporate that aspect, if the user highlighted information, would CyberDesk analyze the highlighted span of text for data structures within that highlighted span? A. That's correct." Dey Depo. at 194:9-196:16 *See also* preamble in Claim 1.
displaying the document electronically using the first computer program;	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.

Exhibit D

information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	
retrieving the first information;	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	CyberDesk discloses this element. *See* corresponding limitation in Claim 1.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the	CyberDesk discloses claim 23. *See* claim 23 as charted above.

Exhibit D

instructions establishing processes comprising:	
providing a prompt for updating the information source to include the first information.	CyberDesk discloses this element. *See* claim 8 above. See, e.g., UIST Article at 75 (including fig. 1): "The user receives an e-mail message (see Figure 1) with the name Andy Wood in it. She highlights the name with her mouse (a) and is shown a list of suggested actions she can perform (b). This list includes searching for the selected text using the AltaVista web search service, looking up a phone number for the selected name using the Switchboard web service, or looking up the selected name in the desktop contact manager. The user chooses the second option and retrieves Andy's phone number and mailing address from the web (c). She wants to update her contact information for Andy, so she chooses the last option which loads Andy Wood's contact information in the contact manager (d)." See, e.g., CyberDesk Technical Report at 3, cols. 1-2: "The IntelliButton component is really the core of the CyberDesk system, as it provides the automatic integrating behaviour. . . . So when a component generates an event, it notifies the IntelliButton and any other components that have expressed interest. . . . It uses simple type checking to identify potential services that the user may wish to call upon to operate on the data associated with the event. The matches are displayed to the user via the ActOn Button Bar, from which the user can select any or none of the integrating services suggested. If the user does choose one of the integrating services, the IntelliButton is notified and it access the correct service passing the associated data and event as parameters. In the above scenario, when the user highlighted the e-mail address, the IntelliButton used that even information to determine what services were available (send an e-mail, save the contact information, etc.) and suggested them." See, e.g., CyberDesk Technical Report at 8, col. 1: "Along the same line of thought, chaining can be used along with the concept of 'combining' to make services more powerful. . . . Combining in CyberDesk terms, is the ability to collect multiple data types for a piece of selected data, and bind them together, as needed, to create multiple meta-objects. These meta-objects could be used to perform substantially more powerful actions. Taking the above example of a user reading an appointment in her calendar, assume that there are multiple services capable of chaining, so a URL selection event is created, a Date selection event is created, a MailingAddress selection event is created, etc. There is potentially enough information to create a new entry in the Contact Manager. The ability to assimilate this widespread information in a compact entity is very powerful for a user.

Exhibit D

	It's this same ability that would allow us to combine time and a name to search a friend's schedule as seen in the user scenario." See, e.g., CyberDesk Technical Report at 1, col. 2—2, col. 1 (including figures): "The user walks to a grocery store, and the system asks if he wants to see his shopping list, get more information about the grocery store, or get directions to his house. The user chooses the grocery list and goes shopping. He walks to a friend's house but nobody is home. The system asks if he wants to check his friend's calendar, contact him via e-mail or phone, or get directions to go home. The user chooses the first option and the system tells him that his friend is at work. So, he chooses the second option, sends his friend an email saying that he stopped by, and starts walking home. On the way home, the system notifies him that he has received an e-mail from his friend. The user reads the e-mail (see Figure 1 below) which has information on a new book written by his favourite author. The e-mail contains a Web site address and an e-mail address for the author. The user highlights the e-mail address (a) and the system gives him some suggestions (b) on what he can do: search for more information on the author, put the author's contact information in the contact manager, call the author, or send an e-mail to the author." See, e.g., CyberDesk Technical Report at 4, cols. 1-2: "All of the desktop applets currently being used in CyberDesk (2 e-mail browsers, contact manager, 2 calendar managers/schedulers, scratchpad) were previously written by other Georgia Tech students." See, e.g., CyberDesk Technical Report at 5, col. 2—6, col. 1 (including fig. 3): "The example below is the wrapper for the Contact Manager (see Figure 3), and it extends the ContactApplet class (the original application class). . . . Lookup an entry for the name in the ContactManager." See, e.g., CyberDesk Technical Report at 6, col. 1 (figure 3): "Screenshot of contact manager being used with CyberDesk. The user selects the string 'Andy Wood' in the –mail tool (a). CyberDesk offers some suggestions (b): search using AltaVista, look up a phone number using Switchboard (c), and look up the name in the contact manager (d)." See, e.g., CyberDesk Technical Report at 7, cols. 1-2: "The next task involved adding the services that recognized the appropriate types and created 'ActOn' buttons for them. We added two services (NewtonNames and NewtonNotes) which, respectively, request contact information from the Newton about the selected name and request notes from the Newton containing the selected text in the body or title. Adding these services was quite simple (see Figure 4), requiring only

Exhibit D

	the implementation of the ServiceApplet methods described above." See, e.g., CyberDesk Technical Report at 7, col. 2—8, col. 1: "The extensions described so far are fairly simplistic. They deal with adding more suggestions for the user. More interesting are extensions which add more powerful suggestions for the user. Chaining is an example of this type of extension. It extends CyberDesk's type converting ability by using network services as type converters, to allow for increased integrating behaviour. For example, a user is reading an appointment in her calendar manager, and selects the name of the person she's supposed to be meeting. As an experienced user, she expects to be presented with a list of all possible services that can use a Name: search for a phone number, mailing address, look up in the contact manager, search name on the WWW, etc. However, by using chaining, more powerful suggestions can be had. The WhoWhereEmail network described earlier, takes a name as input and returns a WWW browser showing a list of possible e-mail addresses corresponding to that name. If we make the assumption (not always a good one) that the first e-mail address returned in the list is the correct one, we can now use this service to convert the name to an e-mail address. The service now creates a related EmailAddress selection event, and the user is supplied with all possible suggestions for both a Name and an EmailAddress." See, e.g., CyberDesk Technical Note at 552, col. 2—553, col. 1 (including figs. 1, 2): "The user highlights some text in the window of one service, and CyberDesk determines the type of the text to suggest how the user can invoke behavior in other services using that text. The suggestions made by CyberDesk appear as a dynamic button bar in a separate 'ActOn' window. For example, at the top left in Figure 1, is an e-mail message informing Anind about the great work going on in the Future Computing Environments group at Georgia Tech. Anind is intrigued and decides to investigate further. Highlight 'Gregory Abowd' causes the ActOn button bar to suggest some actions (a). One suggestion is to look up the name in an available contact manager (b). Anind discovers that he doesn't have Gregory's phone number, so he decides to follow another suggestion and initiates a search using the Switchboard Web service (c). Figure 2 continues the scenario. After speaking with Gregory, Anind wants to visit Georgia Tech, but first he will do some research. He selects the first part of the URL given in the message, and the ActOn buttons change (d). Anind decides to view the URL (e) and use AltaVista to retrieve a list of Web pages that reference URL (f)." See e.g., IUI Proceedings at ARENDI-DEFS00021064-65: "Desktop applications incorporated into CyberDesk include e-mail browsers, notepads, schedulers, and contact managers. Network applications include phone number lookups, e-mail writing, mailing address

Exhibit D

	lookups, Web searches, Usenet searches, e-mail address lookups, map lookups, and Web page browsing. PDA-based applications include contact managers and notepads. All applications make their services available to the user via a common interface. The services available at any particular time depend on the user's context at that time. By providing relevant suggestions and data to the user, the user receives useful and possibly unexpected, help in completing their task . . . To illustrate this behavior, an actual user experience follows . . . a user is checking his e-mail, and reads an e-mail from a friend about some interesting research. The user is interested in the researched discussed, highlights the URL in the message, and CyberDesk offers the following suggestions through its interface: search for the selected text using AltaVista, find pages that reference this URL, using AltaVista, and display the URL in Netscape. He chooses the last option and views the URL listed in the message. The user then selects the name of the person in charge of the research and is offered the following suggestions: search for the selected text using AltaVisata, search for a phone number and mailing address using Switchboard, lookup the name in the contact manager. The user wants to contact the researcher so he checks to see if the name is in his contact manager, but it isn't. So, he selects the phone number and mailing address lookup service. He then creates a new entry in the contact manager with this new information." *See also* AAAI Symposium at ARENDI-DEFS00021057. "After providing the system with these abilities, we made a simple extension to increase the functionality to the user. We noticed that many of the web-based services, which have been integrated into CyberDesk, search for type information that CyberDesk can use. For example, when Switchboard is geiven a name, it returns a phone number and mailing addres. When WhoWhere is given a name, it returns a corresponding e-mail address. The extension we made to CyberDesk takes advantage of these services, by automatically feeding these services with captured context and using the returned data as additional context . . . So, now when a user selects a name, not only do CyberDesk's suggestions deal with using name information but also deal with the automatically obtained e-mail address, phone number, mailing address, URL, etc. This provides the user with additional information to work with and potentially saves the user some effort I locating this information themselves." See AAAI Symposium at ARENDI-DEFS0002105. See, e.g., Pinkerton Thesis at 84-85: "In both the CyberLlama and CyberDesk applications, the components used to view the mobile information do only that -- view. The LlamaShare infrastructure supports writing changes back to the mobile device, but the applications themselves do not support it. We would like to add the following: ● Extending the components to allow modifying mobile information and then storing it back to the Newton.

Exhibit D

	● Allow users to generate content from scratch on the desktop side and save it on the Newton. Currently, the selection of viewers for mobile information on the desktop is rather limited. CyberLlama implements only a 'Notepad' viewer, and the CyberDesk applets can only display 'Notepad' and 'Names' information. The Newton, however, has many more data formats which users would benefit from being able to access and modify on the desktop: ● A 'Calendar' component which would allow users to view and modify the calendar information from a given Newton. CyberDesk could be expanded to show the calendar of the user whose name is selected in an email window by directly accessing the calendar information off of that Newton. ● An application to read the calendar information off of several Newtons and display an aggregation of everyone's calendar info in one place. This would make scheduling group meetings much easier. ● CyberDesk has many built-in types which we would like to support, such as dates, phone numbers, and email addresses. Writing services for CyberDesk which pull the appropriate information from the Newton would make the Newton services as complete as the Internet services." "I don't know about the phone and the tax, because what you can see also here is that John Doe already existed in this – existed in this person's contact manager. And so what CyberDesk would have done is it would have added all the fields that were missing or that were left blank." Dey Depo. at 89:3-8. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 5. For example, CyberDesk and the prior art references and systems set forth in Exhibit F, Table 5 (e.g., Miller, Pandit) relate to managing information in an address book/contact database. A POSITA would have reasonable expectation of success to achieve predictable results in combining CyberDesk and the prior art references and systems in Exhibit U. A POSITA would have recognized advantages and benefits to have CyberDesk provide a prompt for updating the information source to include the first information.

Exhibit E

Claim Chart Applying Apple Data Detectors System Against the '843 Patent

The Apple Data Detectors System is made up of and evidenced by several products, publications, and other sources of evidence, including Apple Internet Address Detectors, US Geographic Detectors, and the deposition of James Miller ("Miller Depo.").

The Apple Internet Address Detectors ("IAD") product, also referred to as "Data Detectors," was offered for sale, sold, publicly disseminated, and publicly used in the United States at least by September 8, 1997. It therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(a), (b) and (g).

An additional product named US Geographic Detectors 1.0 ("Geographic Detectors"), which utilized IAD, was offered for sale, sold, publicly disseminated, and publicly used in the United States around December 23, 1997 and therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(a) and (g).

As shown below, an Apple computer system running IAD and Simple Text and/or Claris Emailer, and for some elements Geographic Detectors, ("IAD System") anticipates and/or renders obvious claims 1, 8, 13, 15, 17-19, 23, and 30 of the '843 patent. The IAD System constitutes prior art under pre-AIA 35 U.S.C. § 102(a) and (g) with Geographic Detectors and 102(a), (b), and (g) without Geographic Detectors.

"Obviousness Statement" - To the extent that the Judge or Jury finds that the Apple Data Detectors System does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

Evidence of the availability of IAD, Geographic Detectors, and the IAD System include the following:

- "Apple Introduces Internet Address Detectors," September 8, 1997
- US Geographic Detectors Read Me file, containing metadata of December 23, 1997 ("US Geographic Detectors 1.0 Read Me file")

Evidence of the design and operation of IAD, Geographic Detectors, and the IAD System include the following:

- "Apple Introduces Internet Address Detectors," September 8, 1997

Exhibit E

- US Geographic Detectors Read Me file, containing metadata of December 23, 1997 ("US Geographic Detectors 1.0 Read Me file")
- Web page for "Apple Data Detectors," last updated December 30, 1996
- Apple Internet Address Detectors User Manual, August 28, 1997 ("User Manual")
- "Apple, StarNine updates in mail," February 23, 1998
- Nardi, B. A., Miller, J. R. & Wright, D. J. (1998). "Collaborative, Programmable Intelligent Agents." *Communications of the ACM*, Vol. 41 No. 3, March 1998 ("Nardi")
- "Claris Em@iler Getting Started"
- Source code for IAD, available for inspection at DLA Piper
- A system running IAD Version 1.0.1 (which is an example of an IAD System), available for inspection at DLA Piper
- A system running IAD Version 1.0.2 and US Geographic Detectors 1.0 (which is an example of an IAD System), available for inspection at DLA Piper
- Screenshots from the system running IAD Version 1.0.1 (which is an example of an IAD System), available for inspection at DLA Piper, as shown below
- Screenshots from the system running IAD Version 1.0.2 and US Geographic Detectors 1.0 (which is an example of an IAD System), available for inspection at DLA Piper, as shown below

- U.S. Patent 5,946,647 "System and Method for Performing an Action on a Structure in Computer-Generated Data", James R. Miller, Thomas Bonura, Bonnie Nardi, David Wright; filed Feb. 1, 1996 ("Miller")

- AppleScript code for "Write a letter, given an e-mail address," January 28, 1997 ("Write a Letter AppleScript Code")

- Worldwide Developers Conference presentation slides for Apple Data Detectives, May 1996 ("WWDC Presentation")

- May 15, 2009 Affidavit by James Miller ("Second Miller Affidavit")

A system running Apple Data Detectors, Now Contact, Claris Works, LiveDoc, and other software was publicly demonstrated at the MacWorld conference in the United States on August 7, 1996, as evidenced in the video produced with Bates number ARENDI-DEFS00000001 ("Video").[1] A system running Apple Data Detectors was further publicly demonstrated at the MacWorld conference in the United States on January 7, 1997, as evidenced in the video produced with Bates number ARENDI-DEFS00021315 ("MacWorld Expo 1997"). This system and the Video constitute prior art under pre-AIA 35 U.S.C. § 102(b) and anticipates and/or renders obvious claims 1, 8, 13, 15, 17, 18, 19, 23, and 30 of the '843 patent.

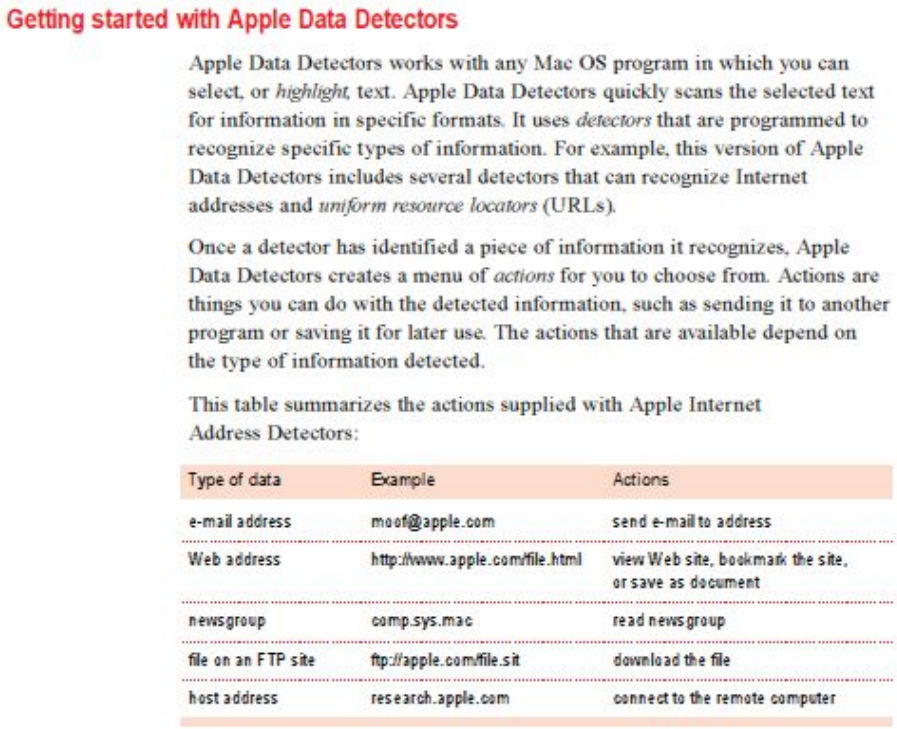
If the Judge or Jury finds that the Apple Data Detectors System does not anticipate a particular claim, then it still renders the claim obvious for the reasons discussed in Exhibit U. Evidence of the design and operation of Apple Data Detectors further includes the above and below-specified evidence pre-dating August 7, 1996 that is described in the claim chart served concurrently for the LiveDoc system as Exhibit H.

[1] To the extent that there is a dispute over the date of availability of any features discussed below, I offer my opinions below with respect to the Video both independently of and in conjunction with the other evidence listed.

Exhibit E

'843 Patent Claims	Disclosure
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, the Apple Data Detectors System discloses the preamble. The User Manual states at pp. 1-2: **Apple Internet Address Detectors User's Manual** Apple Internet Address Detectors utilizes a new Apple technology called *Data Detectors*. Data Detectors enables your computer to recognize and then act on certain types of information, or *data*, in your documents. Apple Data Detectors can recognize several different types of data, and soon software developers will extend the capabilities of Apple Data Detectors even further. Apple Internet Address Detectors for Mac OS 8 can recognize and act on data that's in the form of Internet addresses, which includes the following: ▪ e-mail addresses ▪ Web sites ▪ newsgroup names ▪ filenames on FTP (file transfer protocol) sites ▪ names of remote computers For example, if you have a word-processing document that contains several e-mail addresses, Apple Data Detectors can quickly scan the document, identify all the addresses, and then open a new e-mail message addressed to the one you select. Or, let's say that someone sends you a World Wide Web address in an e-mail message. You can use Apple Data Detectors to find the address within the message, then open the Web page in your favorite Web browser program. For each type of information that Apple Data Detectors identifies, you can select an action to perform with it. The actions available with Apple Internet Address Detectors include ▪ addressing a new e-mail message to the selected address ▪ opening a Web browser program and connecting to the selected Web site ▪ bookmarking the selected Web site in a Web browser program ▪ saving a Web document as a file on your hard disk ▪ downloading the selected file from an FTP site ▪ connecting to the selected remote computer ▪ opening a newsgroup with your news reader program The User Manual states at p. 4:

Exhibit E



Getting started with Apple Data Detectors

Apple Data Detectors works with any Mac OS program in which you can select, or *highlight*, text. Apple Data Detectors quickly scans the selected text for information in specific formats. It uses *detectors* that are programmed to recognize specific types of information. For example, this version of Apple Data Detectors includes several detectors that can recognize Internet addresses and *uniform resource locators* (URLs).

Once a detector has identified a piece of information it recognizes, Apple Data Detectors creates a menu of *actions* for you to choose from. Actions are things you can do with the detected information, such as sending it to another program or saving it for later use. The actions that are available depend on the type of information detected.

This table summarizes the actions supplied with Apple Internet Address Detectors:

Type of data	Example	Actions
e-mail address	moof@apple.com	send e-mail to address
Web address	http://www.apple.com/file.html	view Web site, bookmark the site, or save as document
newsgroup	comp.sys.mac	read newsgroup
file on an FTP site	ftp://apple.com/file.sit	download the file
host address	research.apple.com	connect to the remote computer

The User Manual states at pp. 5-6:

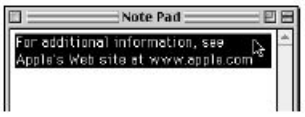
Exhibit E

Using Apple Data Detectors

To use Apple Data Detectors, follow these steps:

1 **Select some text in any application that allows you to highlight text.**

Make sure the selected text contains at least one type of data that Apple Data Detectors recognizes. (In this example, the selected text contains one complete Internet address.)



2 **Hold down the Control key, then press and hold down the mouse button.**

A "contextual menu" appears. It lists all the recognized data found in the selection.

Tip: If the contextual menu is empty, or the message "no structures found in selection" appears, the text you selected did not contain any recognizable data.

After the contextual menu appears, you can release the Control key. Be sure to keep holding down the mouse button or the menu will disappear.

3 **Choose a recognized data item from the contextual menu, then choose an action from the submenu that appears.**

IAD could operate on text entered by a user in a Simple Text file, a Claris Emailer email, or text entered using any other application.

See also Nardi at pp. 96-98 (including figs. 1, 2): "As Apple Computer researchers, we started from a simple but focused approach to agents: That they should have the ability to infer appropriate high-level goals from user actions and requests and take action to achieve these goals. Further, based on a study of reference librarians as exemplary human agents [9], we wanted to build a system in which the user would not have to state goals explicitly and in detail. We learned from librarians

Exhibit E

	that a large part of their value to clients is in working with imprecise requests. Beyond this concern, our general design strategy was to keep the most basic user question in front of us at all times: Will this software do something useful for users in an intelligent way that makes them more productive? The system we describe here—called Apple Data Detectors—meets our criteria of being unobtrusive, being able to infer user needs, and doing useful work. Apple Data Detectors shipped as a product in 1997. Earlier work on intelligent agents was multifaceted, to the point where it is difficult to find a consensus among researchers on exactly what constitutes an 'agent' or even 'intelligence.' However, in nearly all cases, systems described as 'agent-based' rely on some explicitly represented knowledge about relevant aspects of the world—the objects or concepts being addressed by the software, the tasks relevant to the user, and the user's own knowledge about the world. Researchers have used machine learning techniques to track user actions and construct models of user preferences [7], create explicit models of user knowledge and skill levels in an attempt to anticipate user actions, misconceptions, and information needs [2], and implement planning systems to leap from a user's stated intention to the specific actions required to achieve that intention [3]. The locality of agents also varies across different agent-based systems; some act only within one's own machine, find others autonomously crawl the Web, searching for interesting content [4]. We tried to find a middle ground by using explicit representations of user-relevant information as a means of identifying actions users might wish to take but to leave the choice of these actions to users." See also Nardi at fig. 4: "This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter." See also Nardi at 103-04: "Apple Data Detectors is a first step toward extracting semantics from everyday documents without asking users to create documents in new ways. Such an intelligent agent redefines 'document' from a stream of characters to a data structure containing specific, known kinds of structures that can play specific, known roles in user interactions. Such an approach can provide a foundation for more powerful analyses beyond our current recognition and parsing technology. Future work could explore the use of more sophisticated kinds of recognition and parsing, including those that rely on finite state technology and linguistically informed context analysis [5], as well as integration with statistical techniques of data analysis, such as

Exhibit E

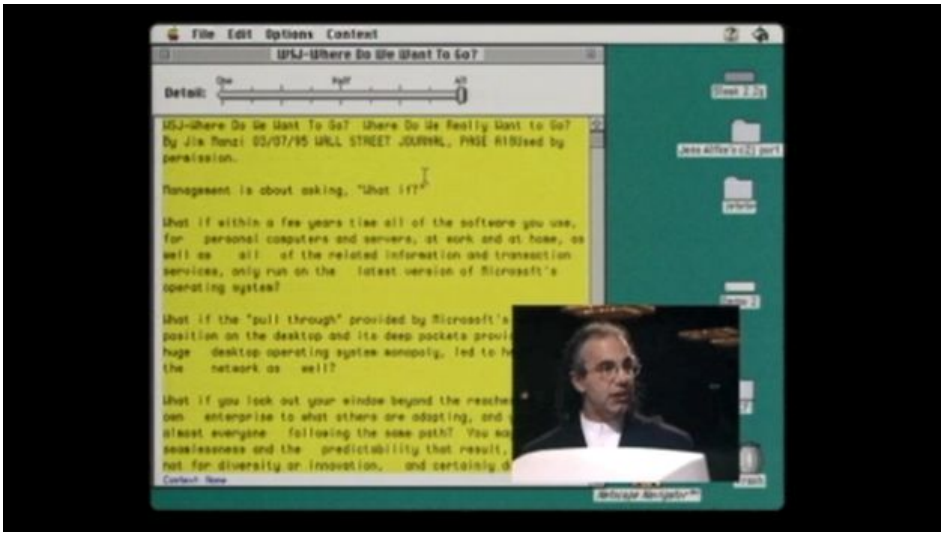
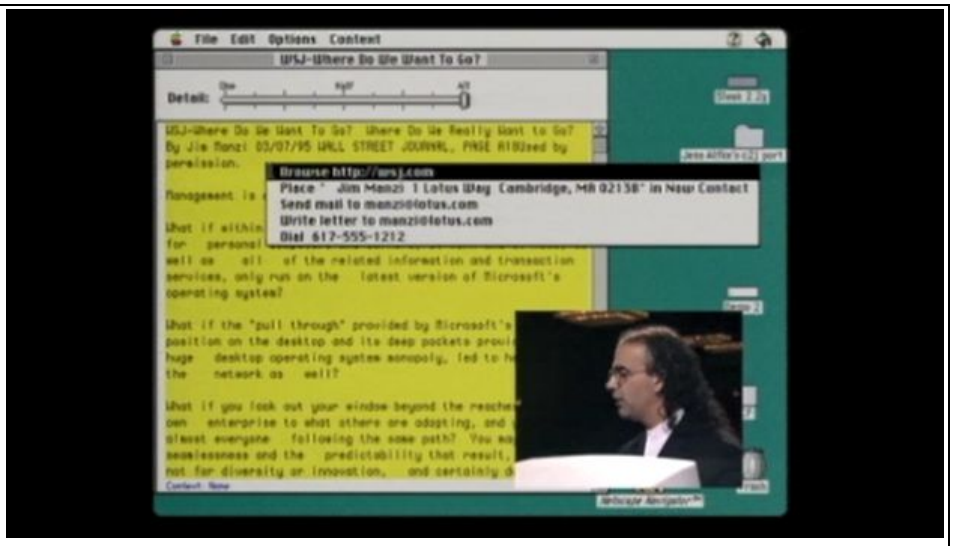
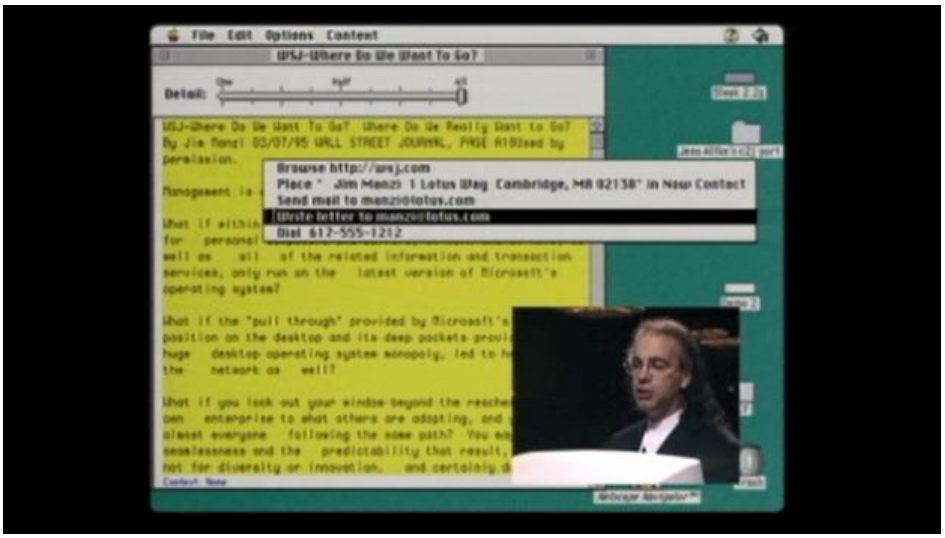
	relevance-based techniques. One important future step for research will be to build knowledge into the system about the structures being recognizing and how these structures are related to user goals and tasks. Doing so will provide the basis for more flexible and powerful task support. For instance, if we can attribute some reasonable set of email address semantics to the textual presentation of an email address in a document, the system can use the address as a pointer to the person with that address. We can then carry out system actions intended for people (such as 'Place a phone call to this person') on an email address and let the system figure out the person implied by the address. (This can already be done through Apple Data Detectors but requires writing a script, rather than relying on inferencing as suggested here.) Such interaction might require a different user interface from the one used today. One can certainly imagine many different kinds of user interfaces to the basic structure-detection technology underlying the current system." *See* Video at 0:15-1:40. *See also* MacWorld Expo 1997 at 1:11:00: 

Exhibit E


File Edit Options Context
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212


File Edit Options Context
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212

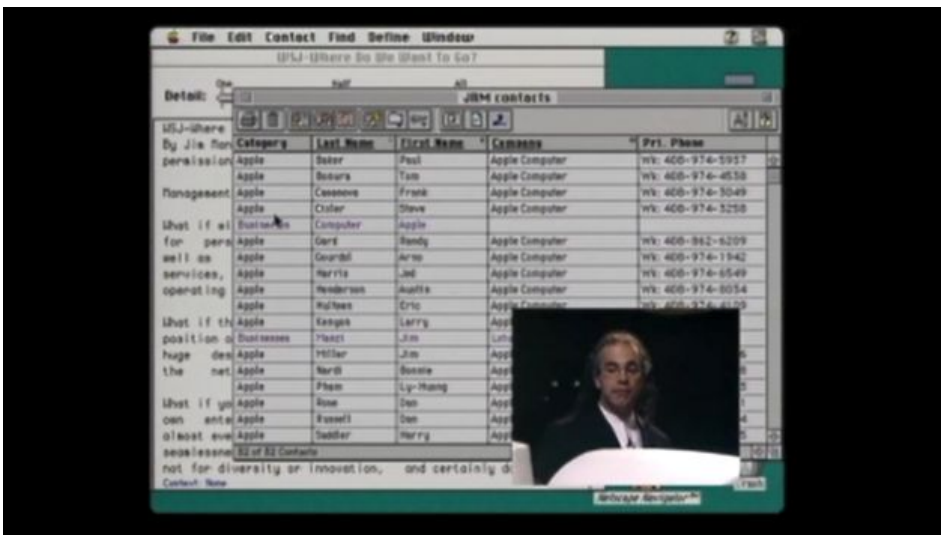
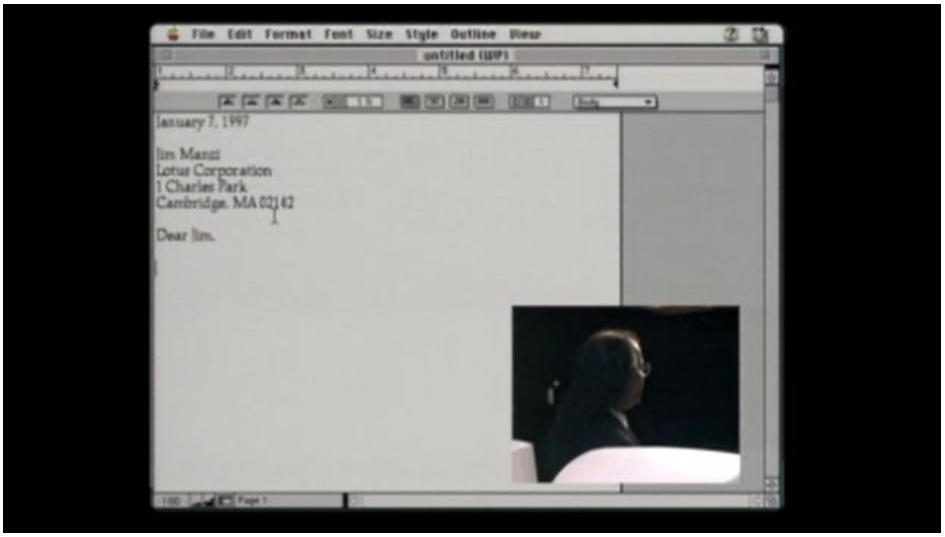

File Edit Contact Find Define Window
WSJ-Where Do We Want To Go?
Detail:
JBM contacts

Exhibit E

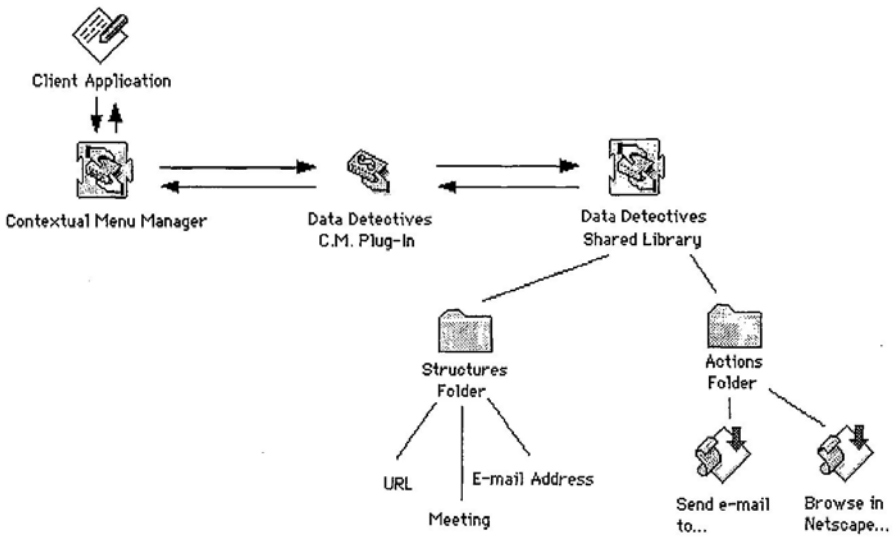


"Apple::Email Address^
Write a letter
Ger the mailing address of the person with this e-mail address, and prepare a piece of ClarisWorks stationery as a letter to them."
See Write a Letter AppleScript Code.

"Find inherently structured data, let user take action on data"
See WWDC Presentation at 2.

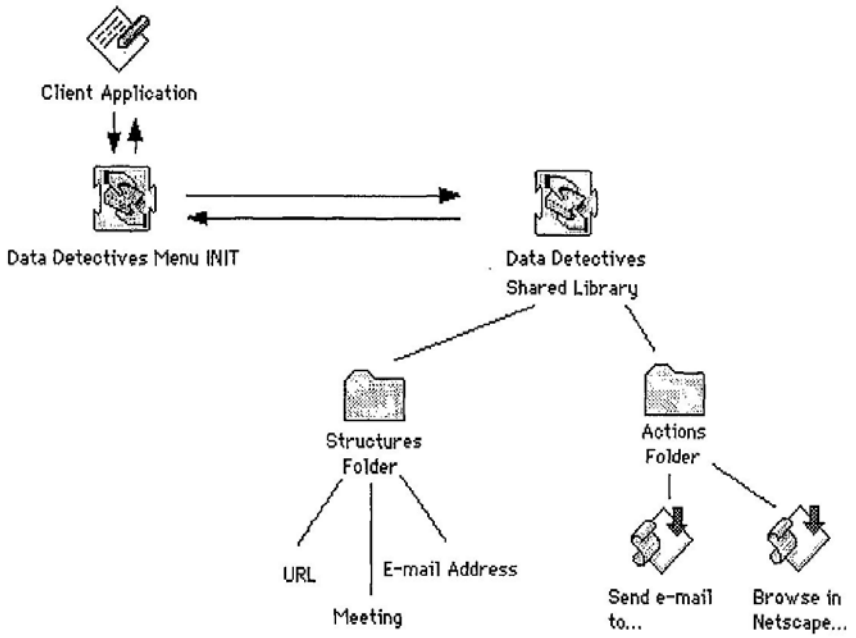
See WWDC Presentation at 4-5:

[Technical Overview - with Contextual Menus in Copland:]



[Technical Overview - Current]

Exhibit E



"A system and method causes a computer to detect and perform actions on structures identified in computer data."

Miller at Abstract

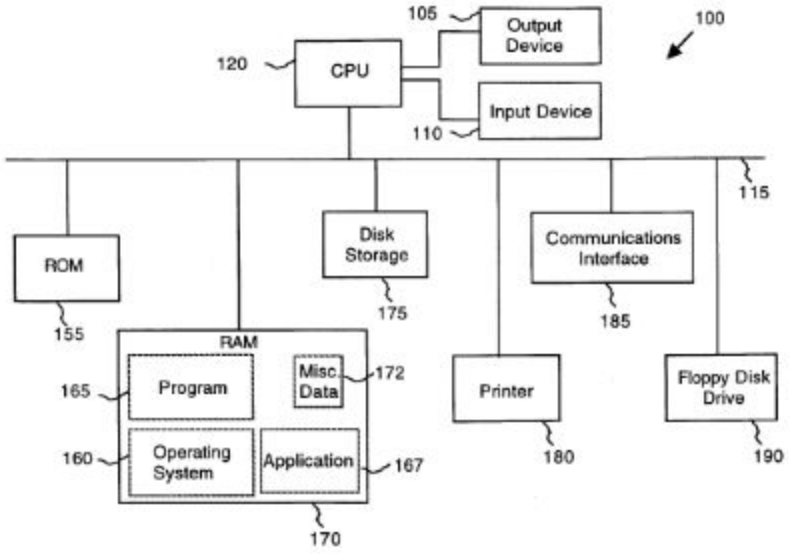


FIG. 1

Miller at Fig.1

Exhibit E

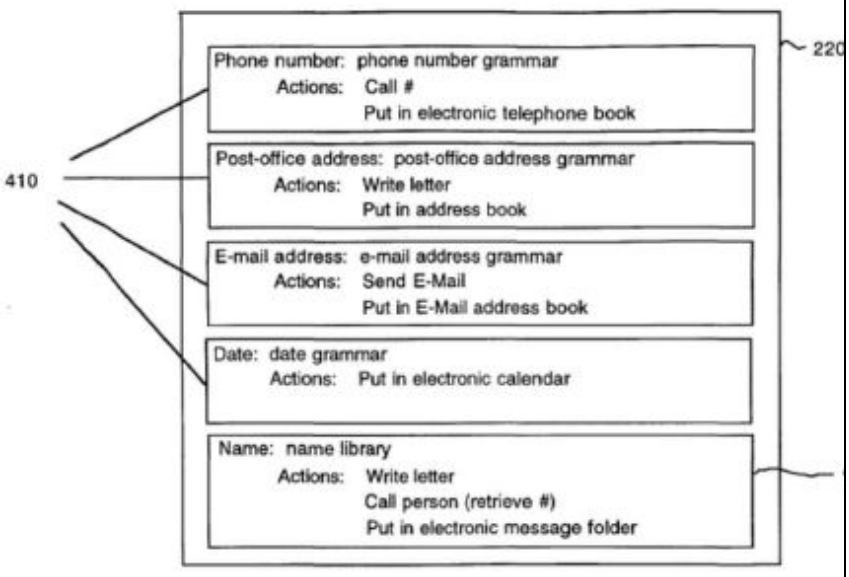
	See also Miller at FIG. 4 at 420 (“Write letter” and “retrieve #”); FIG. 4 at 410 (“Write letter” and “[p]ut in electronic calendar”); 4:58-6:21.  FIG. 4 "The analyzer server receives data from a document." Miller at 2:28. "Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." Miller at 2:42-46. “Application 167 is a program, such as a word-processor or e-mail program, that presents data on output device 105 to a user. The program 165 of the present invention…identif[ies] structures in the data presented by application 167, to associate actions with the structures identified in the data, to enable the user to select a structure and an action, and to automatically perform the selected action on the identified structure.”

Exhibit E

	Miller at 3:36-44. "Upon identification of a structure in the text, parser 310 links the actions associated with the grammar to the identified structure." Miller at 4:64-66. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the 'e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. **Architecture and implementation.** Apple Data Detectors is an open extensible system allowing the recognition and parsing of complex structures. (Nardi, page 99) "Q. Can you give me a general description of what it is that Apple Data Detectors did? A. The idea at the time was to identify bits of information in user documents. For instance, phone numbers or email addresses or, you know, other sorts of things that could be easily identified and make it very easy for people to carry out actions on them." Miller Depo. at 27:2-9. "And so we were looking for ways to find those things in user documents and then make it easy for people to carry out actions on them, to do things with those, let's say, email addresses that were found." Miller Depo. at 39:23-40:2. "The phrase 'your application' refers to the applications that the attendees of WWDC were building. And the point that we wanted to make was that what we were building could provide a way for users to be able to very easily interact with those applications without having to go through many of the details of pulling down menus off of the menu bar and clicking on things and

Exhibit E

	repositioning the cursor and typing stuff into text fields and clicking submit buttons and all of that. They could just interact directly with the content and then let the software go through all of the tedious parts of actually carrying out actions on that information. Q. And when you say 'input data more easily,' do you recall what that was referring to in your presentation? A. Well, that if we could identify, say, a very long email address and provide a way for someone to act directly on it, then the user would not have to open up a text box and remember and type in that email address, possibly making mistakes along the way. Q. So what sorts of things could be done with the email address like that using the Apple Data Detectives that you're describing in this presentation? A. Well, in particular you could create a new outgoing email message to that email address. You could also put it into an address book so that you could easily use it later. Q. How would that work? Using the Apple Data Detectors or Detectives as they are called in this presentation, how would you actually take an email address in a document and put it into an address book? A. There were application programmatic interfaces, APIs, that would receive requests from outside parts of the computer system. And so the way that this would ordinarily work is that our part of the system would identify this and then send a message through this API channel over to the application and say, 'Make a new email message addressed to this person.' Q. How about to save it in their address book, how would that work? A. There would be another kind of message that could be sent to the application. This one would be 'save this email address inside of your address book.' Q And did you actually build that kind of functionality into the Apple Data Detectors? A. Yes. We would work with applications that could accept those messages and then we would write the bits of code that would take the discovered pieces of information and send them to the application." Miller Depo. at 42:23-45:7. Q. … "Then the Data Detectives CM plug-in then exchanges information, it appears, with something called the Data Detectives shared library. Do you see that? A. Yes. Q. Then within that there are two folders, one called "structures folder" and one called "actions folder." Is that right? A. Yes.

Exhibit E

	Q. What is meant by "structures folder"? A. These were the patterns that would recognize different kinds of information. So there would be a component that was able to recognize URLs, http:// followed by a bunch of stuff, or an email address. Q. And I notice in the structures folder, in addition to URLs, it also indicates email address; is that right? A. Yes. Q. And is that something then that you had disclosed that the Apple Data Detectives would be able to recognize as a structure? A. Yes. Q. And I also notice meetings is included in the structure folder. Was that another structure that you disclosed that Apple Data Detectives would be able to identify? A. Yes. Q. Then there's the separate folder called "actions folder." What is that intended to indicate? A. This contains these small bits of code that are able to talk to the applications to carry out the actions. And that after a user had -- or after we had identified a piece of information like an email address, and the user had then indicated I would like to send an email message to that -- to that person, that little application icon would send email to, that would run accept the email address and tell its targeted application to create a new email message." Miller Depo. at 46:23-48:15. "Q. And again, what were you trying to accomplish with this presentation to developers? A. Well, this would point out to developers that their applications could be accessed directly from user information. As opposed to the user having to go back into the system finder and open up a sequence of folders until they finally got to the user's application and then launch that and then do activities inside of the application." Miller Depo. at 49:15-24. "And why was it important that – to allow users to do otherwise time-consuming actions on documents with a single click of the mouse? A. It makes their interaction with their information much easier." Miller Depo. at 70:14-19. "Am I correct that as we saw on the demo, you could, in fact, select the entire document to have Data Detectors review it? A. Yes. Q. And am I correct that Data Detectors would work even if a user did not specifically designate, for example, the particular phone number that they were interested in acting on? MS. FRANKLIN: Objection to form.

Exhibit E

	THE WITNESS: Yes. BY MR. UNIKEL: Q. Why is it that you wanted to allow users to be able to analyze as much as all of the text in a document? A. Well, it's much easier to just very quickly select a region of text on the screen and let the computer figure out what things can be acted on in it as opposed to having to go in, in this case with a mouse, and very carefully only select the characters that you're interested in. It just requires less precise action on the part of the user. Q. What happens if there were multiple different data objects within the text of the letter, let's say, an email address and a phone number and a map address? Could the Data Detectors detect all of those things? A. Yes, that's what is shown in the figure at the bottom of, I guess, Page 2 of Miller 7. There is a highlighted region of text in the window and the hierarchical menus showing up on the page are showing a phone number and an email address and a URL." Miller Depo. at 71:13-72:20. "Q. What about a word processing program, could Apple Data Detectors be used in conjunction with a word processing program at the time? A Yes. Q. What was an example of a word processing program available on Macintosh that Apple Data Detectors could work in conjunction with? A. ClarisWorks is one." Miller Depo. at 74:5-14. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of

Exhibit E

	the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4 "A. Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:4-16. "A. It was possible to add an address to some contact managers' address book. I believe it was Now Contact." Miller Depo. at 84:18-20. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. "Q. I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes.

Exhibit E

	Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "Q. So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15. "And so Data Detectors in that demonstration, were they being applied to only particular text or to the entire document? A. They were being applied to whatever part of the document had been selected. The yellow color here shows the text that has been selected by the user. I don't know exactly what selection took place, but it's certainly all of the text that is visible on the screen. Q. So everything that would be selected, if it was the entire document that was selected, the entire document would be analyzed? A. Yes. Q. And did the – does the user have to particularly designate somehow a particular piece of information like a telephone number that it wants to have detected? A. Well, Data Detectors will come back with that pop-up menu showing all of the structures that it found in the selected text." Miller Depo. at 119:14-120:8. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a

Exhibit E

	phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And so looking back at the screenshot that is Number 2 in this Miller Exhibit 14, am I correct that when Lotus – I'm sorry, when Apple Data Detectors are invoked on this document, the user is given, for this particular document, five different options of things that he or she can do? A. Yes. * * * Q. And the user can click any of these with a single click of the mouse? A. Yes." Miller Depo. at 129:17-130:18. "Q. Why is it that you separated the Apple Data Detector from the application? MR. WILLIAMS: Objection to form. THE WITNESS: We would not be able to build Apple Data Detectors into all of the applications that were out there. Those were other companies' applications with their own proprietary interests and their work. BY MR. UNIKEL: Q. And so why create it as a separate piece of code from the application itself? A. So that those applications, by making use of the Data Detector technologies, could still get access to the functionality." Miller Depo. at 148:6-20. Apple Data Detectors parse, recognize, and determine the type of

Exhibit E

	information in a text, e.g., email address. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was

Exhibit E

	detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Q. Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "A. LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select a region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen. And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "Q. And again, when you're referring to initiating the grammatical analysis of the selected text, what is that referring to? A. That means instructing Data Detectors to run the selected texts through its set of structure recognizers, I guess." Miller Depo. at 165:11-16. "Q. So what are the kinds of things then that LiveDoc could find and then highlight for a user? A. Well, the same things that were found by the shipping version of Data Detectors. In fact, both the shipping version and LiveDoc used the same underlying Data Detector capability." Miller Depo. at 174:14-20.

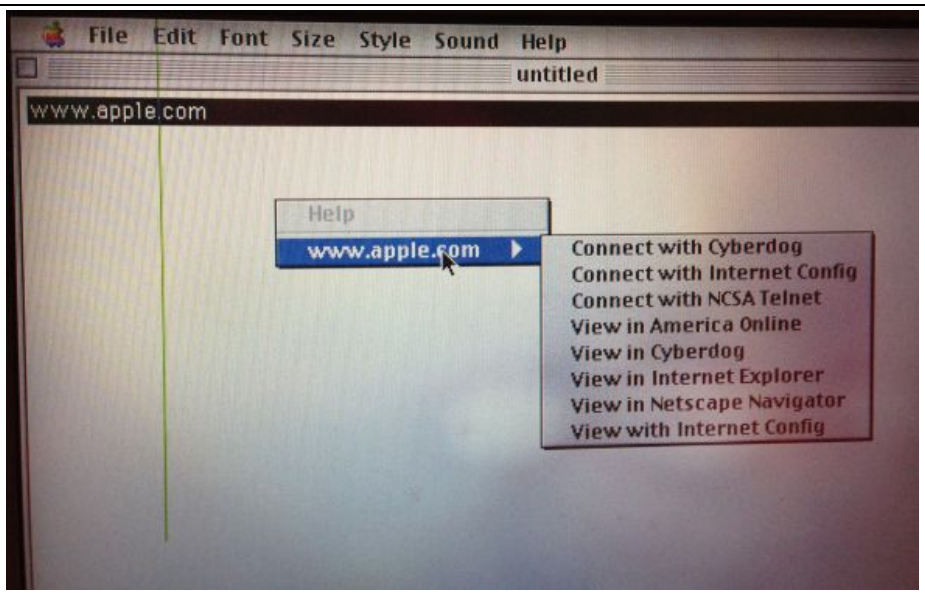
Exhibit E

	"Q. When you say 'pointing at a highlight and pressing the mouse button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "So the LiveDoc manager gets the text from the visible part of the screen and looks for the recognizable patterns in it. Now, it does that just by looking at this long string of texts making up all the content of the screen. So as noted here, it might say that there is a particular pattern between – or in this substring of the character. The LiveDoc manager is the thing that is going to draw those highlights." Miller Depo. at 179:12-22. "[W]hy did you make LiveDoc its own separate piece of code where it had to exchange information with the application rather than just making it a part of every application? A. I think for much the same reason – I mean, this is going back quite a ways. But I think it was the same sort of logic as for making the Data Detector engine a separate thing. That by keeping the LiveDoc manager as an external component, we would be more able to connect into potentially more applications. We wouldn't be building LiveDoc capability into all of those applications. All we would – all we would need would be for that application to be able to tell the LiveDoc manager where on the screen are these characters." Miller Depo. at 180:11-181:3. "A. There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains

Exhibit E

	highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "A. If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	The Apple Data Detectors System discloses this element. IAD could operate on text entered by a user. *See, e.g.* Miller Dep. at 74:10-14.

Exhibit E



See also Nardi at fig. 1.

See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts.

Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information."

Exhibit E

	See also Nardi at 98: "User interface. To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1)." See also Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." See also Nardi at 101: "Apple Data Detectors assumes a world of heterogenous data (in the user's machine) that comes from different applications in different file formats. For example, it makes sense to put an address in an address book, whether the address is from a message sent in any of several mail programs, appears in a downloaded document, or is in another address book maintained by the user. Apple Data Detectors is a pervasive technology, giving users access to actions appropriate for data in an entire set of documents." See also Nardi at 103-04: "Apple Data Detectors is a first step toward extracting semantics from everyday documents without asking users to create documents in new ways. Such an intelligent agent redefines 'document' from a stream of characters to a data structure containing specific, known kinds of structures that can play specific, known roles in user interactions. Such an approach can provide a foundation for more powerful analyses beyond our current recognition and parsing technology. Future work could explore the use of more sophisticated kinds of recognition and parsing, including those that rely on finite state technology and linguistically informed context analysis [5], as well as integration with statistical techniques of data analysis, such as relevance-based techniques. One important future step for research will be to build knowledge into

Exhibit E

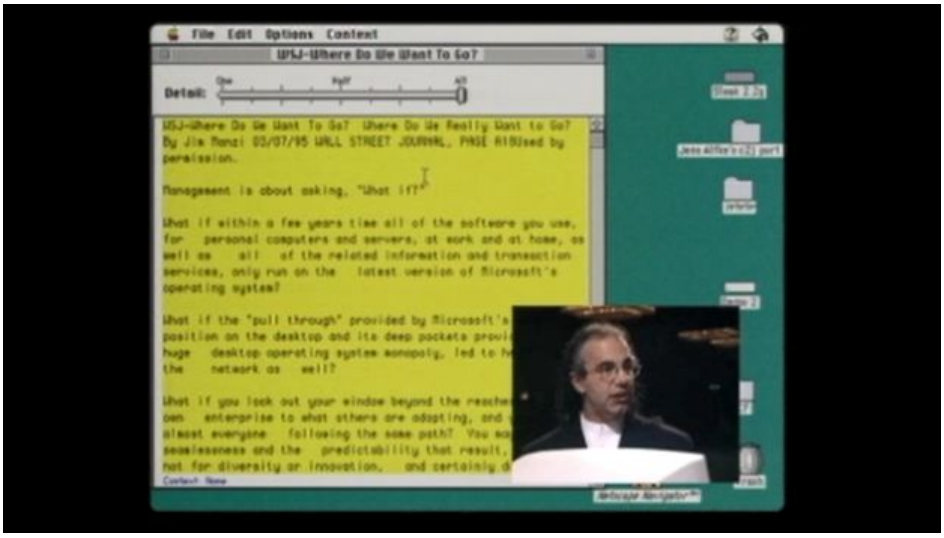
	the system about the structures being recognizing and how these structures are related to user goals and tasks. Doing so will provide the basis for more flexible and powerful task support. For instance, if we can attribute some reasonable set of email address semantics to the textual presentation of an email address in a document, the system can use the address as a pointer to the person with that address. We can then carry out system actions intended for people (such as 'Place a phone call to this person') on an email address and let the system figure out the person implied by the address. (This can already be done through Apple Data Detectors but requires writing a script, rather than relying on inferencing as suggested here.) Such interaction might require a different user interface from the one used today. One can certainly imagine many different kinds of user interfaces to the basic structure-detection technology underlying the current system." Mac World Data Detectors System included a document. *See* Video at 0:00-1:17. *See also* MacWorld 1997 Expo at 1:11:00:  For example, Miller teaches: Documents are displayed using a first computer program, such as a word processor (application 167 in Fig. 1). *See, e.g.,* Figs. 1 and 5; 5:19-22 ("FIG. 5 shows a window 510 presenting an exemplary document 210 ... "); 3:36-38 ("Application 167 is a program, such as a word processor or e-mail program, that presents data on output device 105 to user."). Output device 105 (Fig. 1) can be a CRT display device. 3:26-38.

Exhibit E

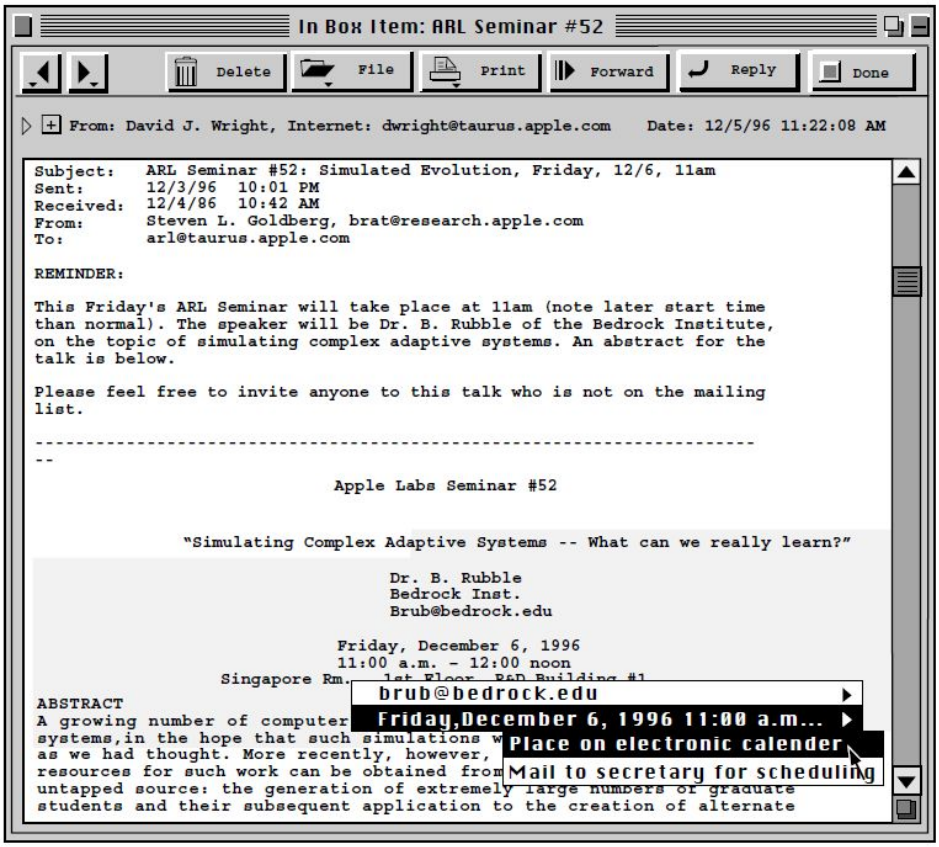


Figure 1. Sample invocation of Apple Data Detectors. The user has selected a portion of an email message describing an upcoming seminar. Two patterns are found: an email address (brub@bedrock.edu) and the announcement of the meeting (the sequence of date and time information starting Friday, Dec. 6, 1996). These patterns are presented in the pop-up menu; by pointing at the date information in the menu; a second pop-up menu offers a choice of actions: place an entry for the meeting on the user's electronic calendar or mail the selection to the user's secretary. The user can select one, thereby running a small application, or move the cursor off the menu, eliminating the pop-up menu and canceling any actions.

(Nardi, page 97).

"And so we were looking for ways to find those things in user documents and then make it easy for people to carry out actions on them, to do things with those, let's say, email addresses that were found." Miller Depo. at 39:23-40:2.

"Q. What about a word processing program, could Apple Data Detectors be used in conjunction with a word processing program at the time?
A Yes.
Q. What was an example of a word processing program available on Macintosh that Apple Data Detectors could work in conjunction with?
A. ClarisWorks is one." Miller Depo. at 74:5-14.

"Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call.
So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go

Exhibit E

	out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "And so Data Detectors in that demonstration, were they being applied to only particular text or to the entire document? A. They were being applied to whatever part of the document had been selected. The yellow color here shows the text that has been selected by the user. I don't know exactly what selection took place, but it's certainly all of the text that is visible on the screen. Q. So everything that would be selected, if it was the entire document that was selected, the entire document would be analyzed? A. Yes. Q. And did the – does the user have to particularly designate somehow a particular piece of information like a telephone number that it wants to have detected? A. Well, Data Detectors will come back with that pop-up menu showing all of the structures that it found in the selected text." Miller Depo. at 119:14-120:8. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted." Miller Depo. at 184:23-185:5. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the

Exhibit E

	option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example, in the screenshot below, which was taken during my Inspection of the Powerbook 3400C, I observed that text is displayed in the Note Pad application.  For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a	The Apple Data Detectors System discloses this element. The User Manual states at pp. 5-6:

Exhibit E

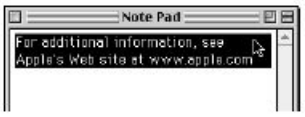
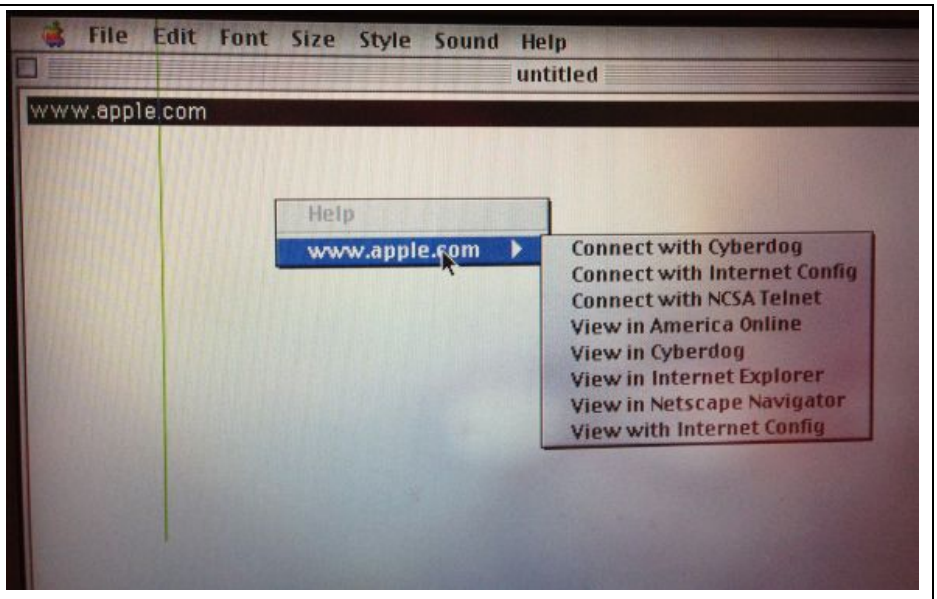
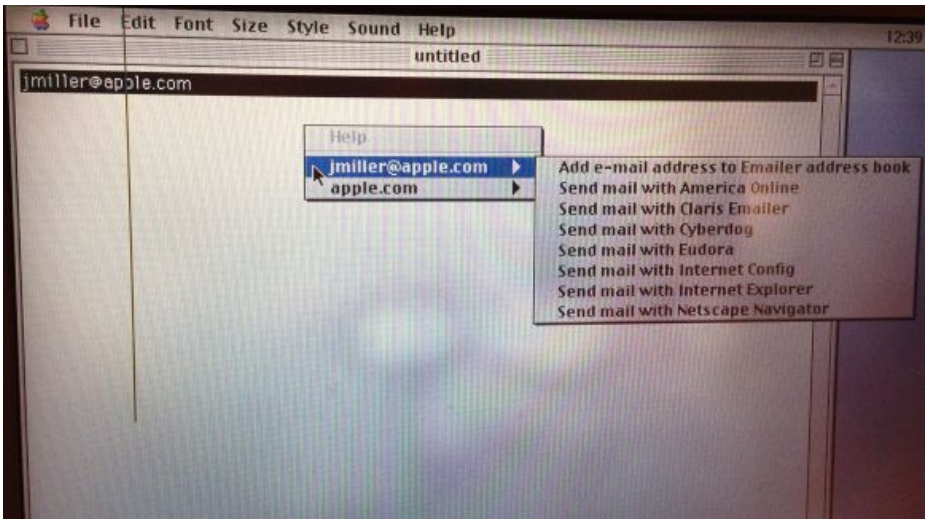
plurality of types of information that can be searched for in order to find second information related to the first information;	**Using Apple Data Detectors** To use Apple Data Detectors, follow these steps: **1** **Select some text in any application that allows you to highlight text.** Make sure the selected text contains at least one type of data that Apple Data Detectors recognizes. (In this example, the selected text contains one complete Internet address.)  **2** **Hold down the Control key, then press and hold down the mouse button.** A "contextual menu" appears. It lists all the recognized data found in the selection.  *Tip:* If the contextual menu is empty, or the message "no structures found in selection" appears, the text you selected did not contain any recognizable data. After the contextual menu appears, you can release the Control key. Be sure to keep holding down the mouse button or the menu will disappear. **3** **Choose a recognized data item from the contextual menu, then choose an action from the submenu that appears.**  For example, when a user selects text and right-clicks in Simple Text, IAD can detect URLs and provide options:

Exhibit E



(This and other screenshots contained within this chart were taken during inspection of a computer running IAD.)

When a user selects text and right-clicks in Simple Text, IAD can detect email addresses and provide options:



This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for an email address. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could

Exhibit E

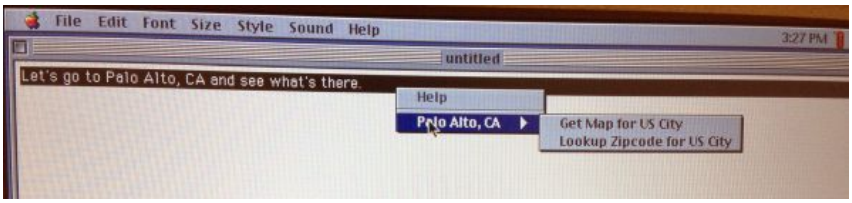
	then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could obtain a zip code for the selected US city, for which the system would search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:  This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.) See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information." See also Nardi at 99: "Apple Data Detectors is an open extensible

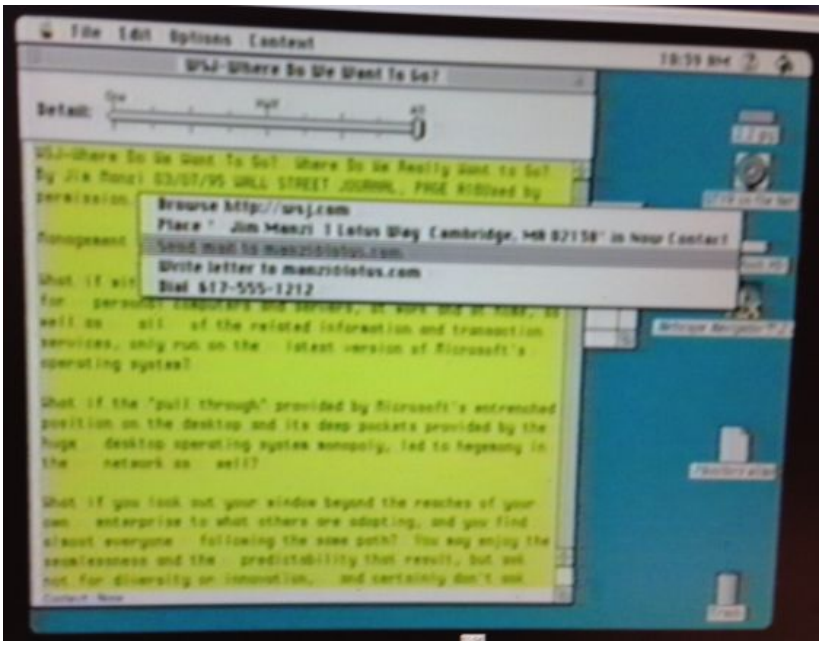
Exhibit E

	system allowing the recognition and parsing of complex structures. Its recognition technology is a hybrid system that uses Earley's algorithm and deterministic finite automata. The algorithms permit recognition of not only simple structures, such as predefined strings and atomic patterns like URLs and email addresses, but complex composite structures, such as meeting announcements composed of small more atomic structures, like date, time, and place." See also Nardi at p. 99 ("Figure 4. An action script, demonstrating the generality of Apple Data Detectors' use of a scripting language and external applications as information repositories and as end-user tools. This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a "personal information manager" (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter."). See also Nardi at 103-04: "Apple Data Detectors is a first step toward extracting semantics from everyday documents without asking users to create documents in new ways. Such an intelligent agent redefines 'document' from a stream of characters to a data structure containing specific, known kinds of structures that can play specific, known roles in user interactions. Such an approach can provide a foundation for more powerful analyses beyond our current recognition and parsing technology. Future work could explore the use of more sophisticated kinds of recognition and parsing, including those that rely on finite state technology and linguistically informed context analysis [5], as well as integration with statistical techniques of data analysis, such as relevance-based techniques. One important future step for research will be to build knowledge into the system about the structures being recognizing and how these structures are related to user goals and tasks. Doing so will provide the basis for more flexible and powerful task support. For instance, if we can attribute some reasonable set of email address semantics to the textual presentation of an email address in a document, the system can use the address as a pointer to the person with that address. We can then carry out system actions intended for people (such as 'Place a phone call to this person') on an email address and let the system figure out the person implied by the address. (This can already be done through Apple Data Detectors but requires writing a script, rather than relying on inferencing as suggested here.) Such interaction might require a

Exhibit E

different user interface from the one used today. One can certainly imagine many different kinds of user interfaces to the basic structure-detection technology underlying the current system."

Mac World Data Detectors System could detect data structures in a written document and provide options for the detected structures. For example:



See Video.

See also MacWorld 1997 Expo at 1:11:00:

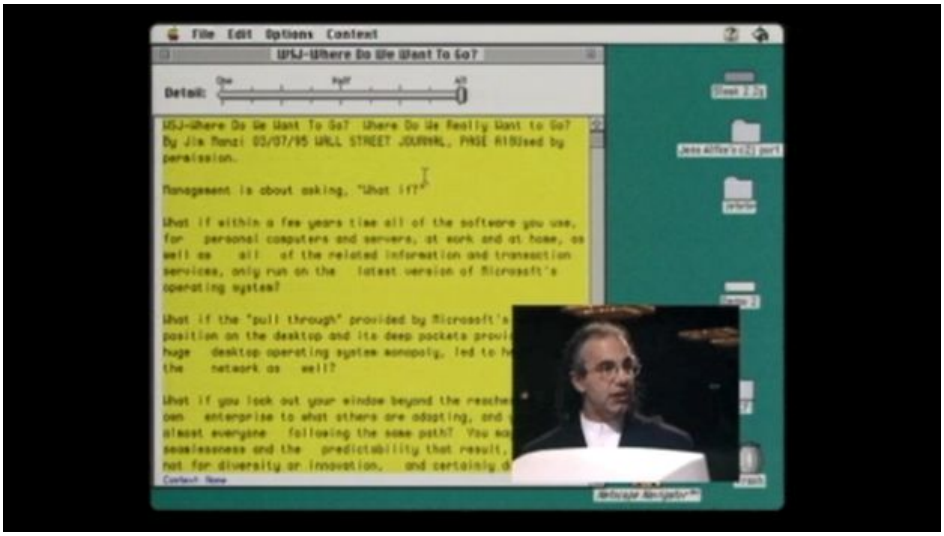
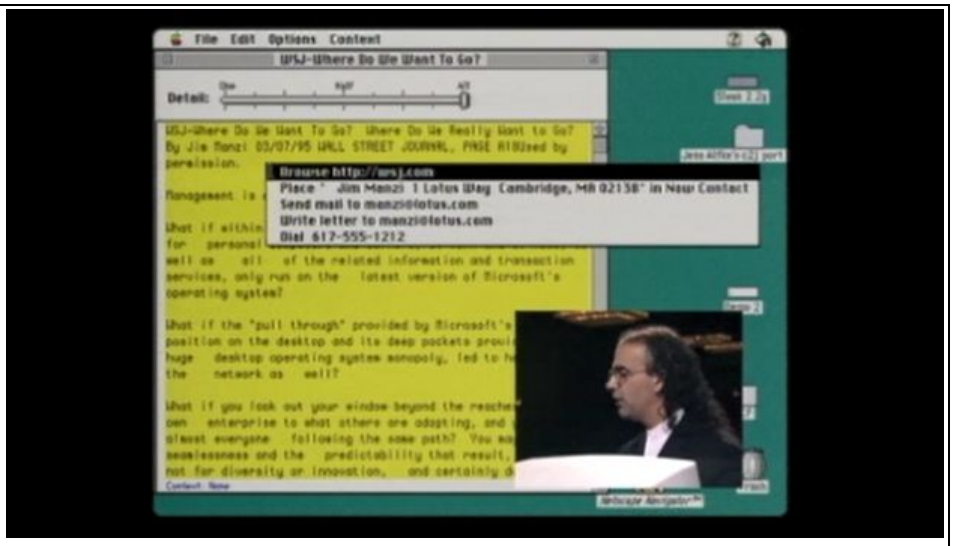


Exhibit E



"Find inherently structured data, let user take action on data"
See WWDC Presentation at 2.

See WWDC Presentation at 4-5:

"Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications."

Miller at 2:42-46.

"FIGS. 8 and 9 display a flowchart illustrating preferred method 800 for recognizing patterns in documents and performing actions. This method is carried out during the run-time of application 167."

Miller at 5:51-54.

Exhibit E

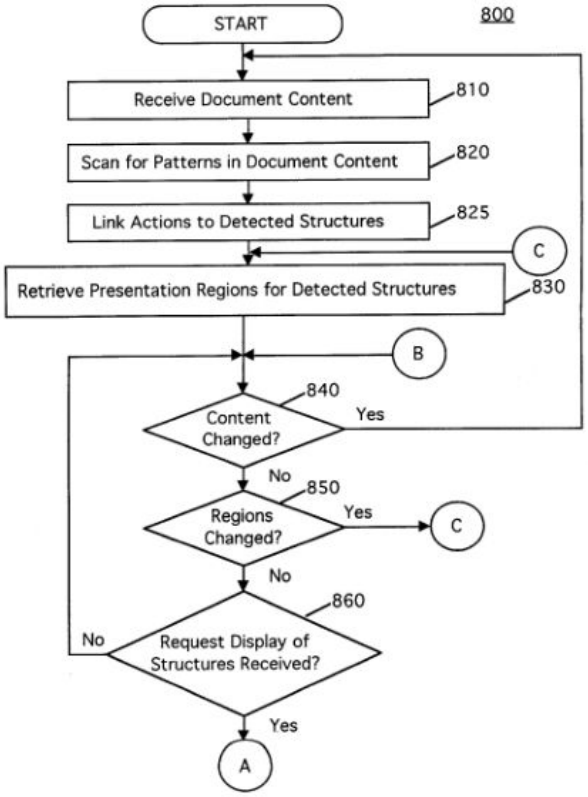


FIG. 8

See also FIGS. 5 and 6.

"If the document content being displayed on output device 105 is changed 840, for example by the user adding or modifying text, method 800 restarts."

Miller at 5:64-66.

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an

Exhibit E

address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

Miller at 5:6-5:18.

See also FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18

See also, e.g., Figs. 5 and 6; 3:61-64 ("Analyzer server 220 ... uses patterns to parse document 210 for recognizable structures."); 5: 19-36
(" ... As illustrated in FIG. 6, analyzer server 220 identifies the phone number, post-office address, e-mail address and name ... ").

FIG. 5

Miller at Fig. 5.

User interface. To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98).

"Then the Data Detectives CM plug-in then exchanges information, it

Exhibit E

	appears, with something called the Data Detectives shared library. Do you see that? A. Yes. Q. Then within that there are two folders, one called "structures folder" and one called "actions folder." Is that right? A. Yes. Q. What is meant by "structures folder"? A. These were the patterns that would recognize different kinds of information. So there would be a component that was able to recognize URLs, http:// followed by a bunch of stuff, or an email address. Q. And I notice in the structures folder, in addition to URLs, it also indicates email address; is that right? A. Yes. Q. And is that something then that you had disclosed that the Apple Data Detectives would be able to recognize as a structure? A. Yes. Q. And I also notice meetings is included in the structure folder. Was that another structure that you disclosed that Apple Data Detectives would be able to identify? A. Yes. Q. Then there's the separate folder called "actions folder." What is that intended to indicate? A. This contains these small bits of code that are able to talk to the applications to carry out the actions. And that after a user had -- or after we had identified a piece of information like an email address, and the user had then indicated I would like to send an email message to that -- to that person, that little application icon would send email to, that would run accept the email address and tell its targeted application to create a new email message." Miller Depo. at 46:23-48:15. "Am I correct that as we saw on the demo, you could, in fact, select the entire document to have Data Detectors review it? A. Yes. Q. And am I correct that Data Detectors would work even if a user did not specifically designate, for example, the particular phone number that they were interested in acting on? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. Q. Why is it that you wanted to allow users to be able to analyze as much as all of the text in a document? A. Well, it's much easier to just very quickly select a region of text on the screen and let the computer figure out what things can be acted on in it as opposed to having to go in, in this case with a mouse, and very carefully only select the characters that you're interested in. It just requires less precise action on the part of the user.

Exhibit E

	Q. What happens if there were multiple different data objects within the text of the letter, let's say, an email address and a phone number and a map address? Could the Data Detectors detect all of those things? A. Yes, that's what is shown in the figure at the bottom of, I guess, Page 2 of Miller 7. There is a highlighted region of text in the window and the hierarchical menus showing up on the page are showing a phone number and an email address and a URL." Miller Depo. at 71:13-72:20. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4

Exhibit E

	"And so Data Detectors in that demonstration, were they being applied to only particular text or to the entire document? A. They were being applied to whatever part of the document had been selected. The yellow color here shows the text that has been selected by the user. I don't know exactly what selection took place, but it's certainly all of the text that is visible on the screen. Q. So everything that would be selected, if it was the entire document that was selected, the entire document would be analyzed? A. Yes. Q. And did the – does the user have to particularly designate somehow a particular piece of information like a telephone number that it wants to have detected? A. Well, Data Detectors will come back with that pop-up menu showing all of the structures that it found in the selected text." Miller Depo. at 119:14-120:8. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually

Exhibit E

	get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "And again, when you're referring to initiating the grammatical analysis of the selected text, what is that referring to? A. That means instructing Data Detectors to run the selected texts through its set of structure recognizers, I guess." Miller Depo. at 165:11-16. "Q. So what are the kinds of things then that LiveDoc could find and then highlight for a user? A. Well, the same things that were found by the shipping version of Data Detectors. In fact, both the shipping version and LiveDoc used the same underlying Data Detector capability." Miller Depo. at 174:14-20. "So the LiveDoc manager gets the text from the visible part of the screen and looks for the recognizable patterns in it. Now, it does that just by looking at this long string of texts making up all the content of the screen. So as noted here, it might say that there is a particular pattern between – or in this substring of the character. The LiveDoc manager is the thing that is going to draw those highlights." Miller Depo. at 179:12-22. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example, in the screenshot below, which was taken during my Inspection of the Powerbook 3400C, while the text document was being displayed, the text "test1@apple.com" was highlighted in Note Pad with the mouse cursor and right-clicked. A menu then appeared that included the highlighted "test1@apple.com" text along with a sub-menu of actions associated with the highlighted "test1@apple.com" text. Data Detectors detected that "test1@apple.com" was an email address, and presented actions in the sub-menu associated with an email address.

Exhibit E

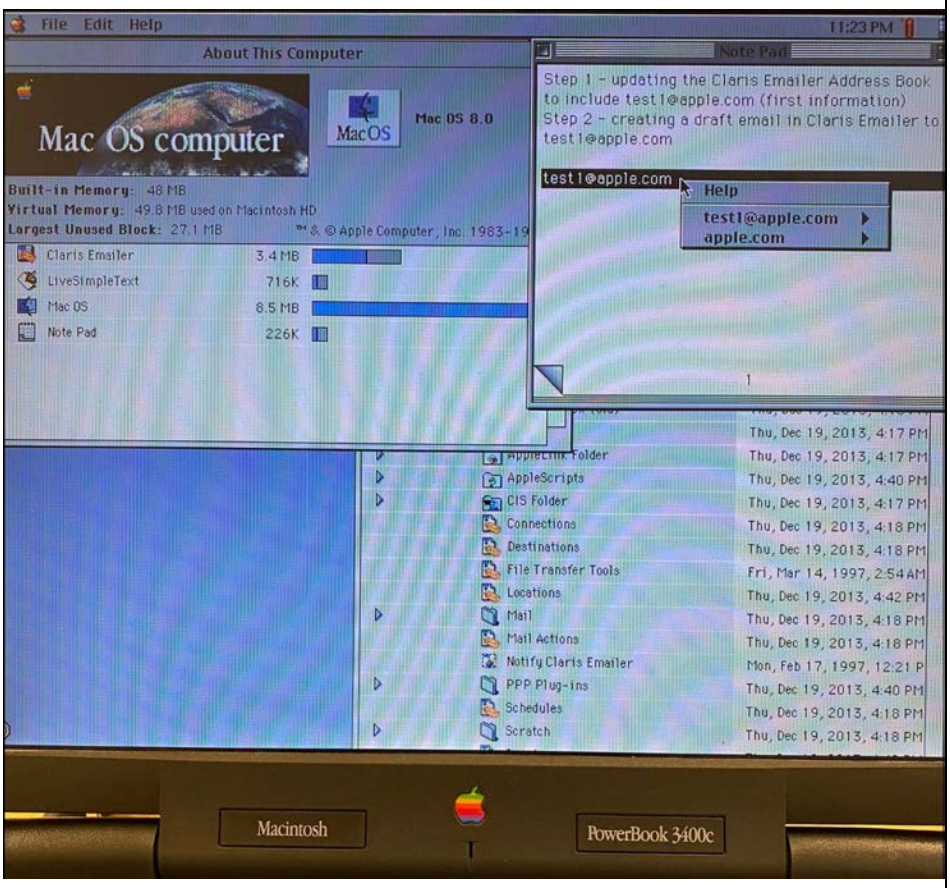
	Data Detectors also detected that "apple.com" was an Internet address, and presented actions in the sub-menu associated with an Internet address. 

Exhibit E

I further observed that Data Detectors was configured to present the aforementioned sub-menu of actions when an email address was detected. The following images show that a user could select which detectors to use to detect an email address, and further select which actions to present to future users when an email address was detected. The enabled actions from the configuration screen were indeed the actions presented in the sub-menu when "test1@apple.com" was detected as an email address.

Exhibit E

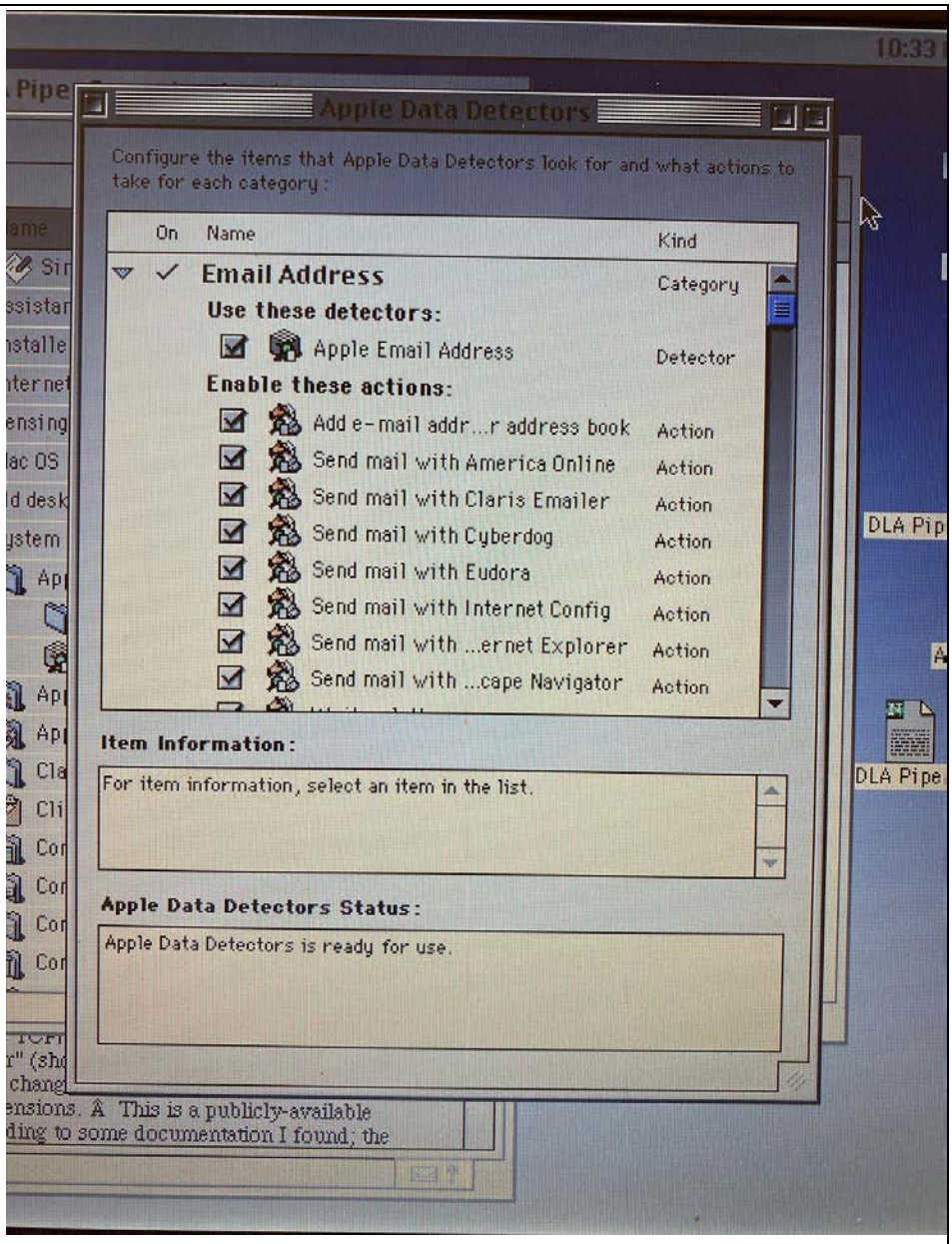
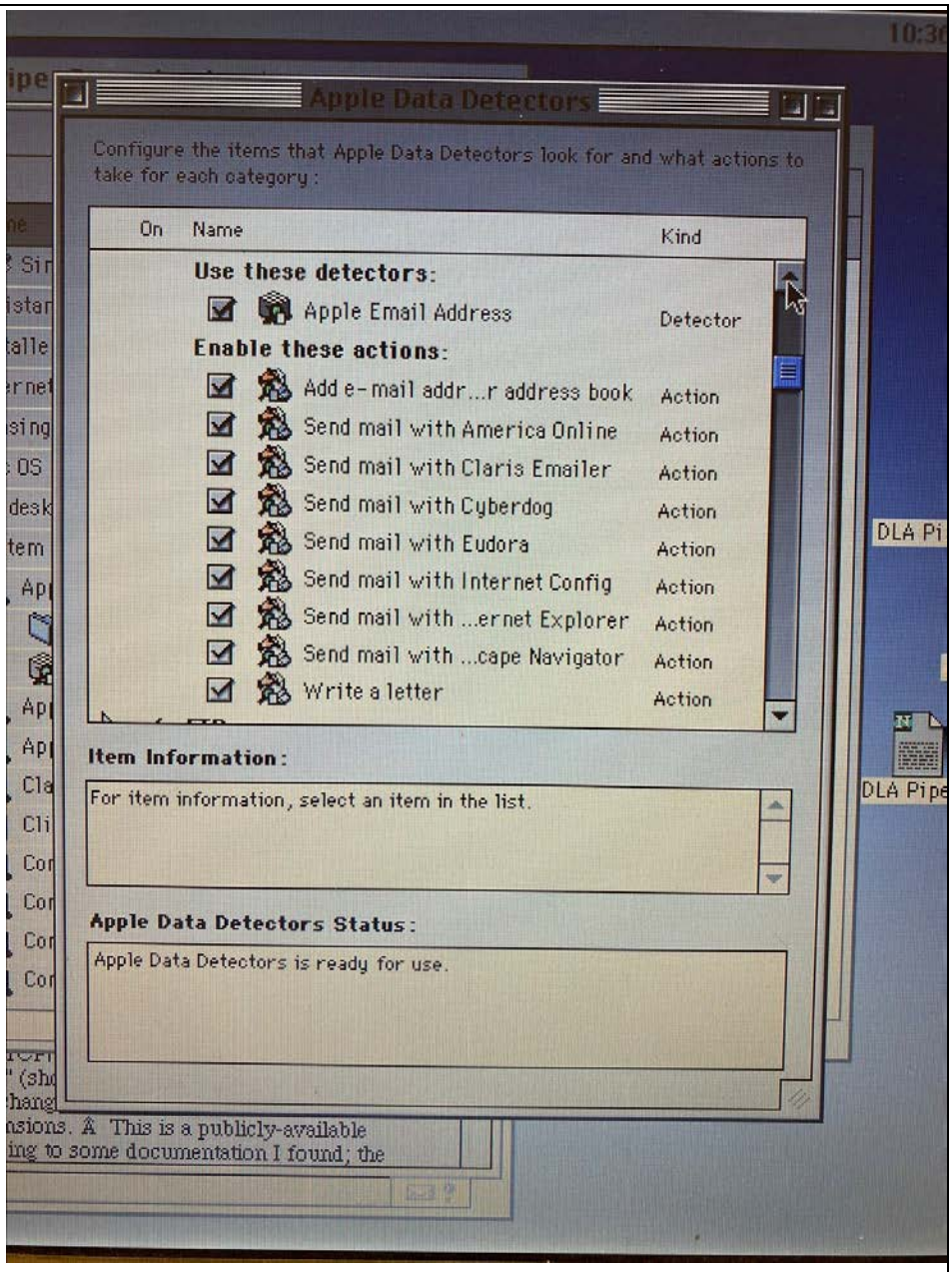

Apple Data Detectors
Configure the items that Apple Data Detectors look for and what actions to take for each category:
On
Name
Kind
Email Address
Category
Use these detectors:
Apple Email Address
Detector
Enable these actions:
Add e-mail addr...r address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with ...ernet Explorer
Send mail with ...cape Navigator
Action
Item Information:
For item information, select an item in the list.
Apple Data Detectors Status:
Apple Data Detectors is ready for use.
DLA Pip
DLA Pipe

Exhibit E



I further observed that there were a number of other actions available to Data Detectors on the Powerbook 3400C. As seen in the screenshots below, the "Apple Data Detectors" folder contained a subfolder called "Actions," which contained code files for a number of actions for detected data types.

Exhibit E

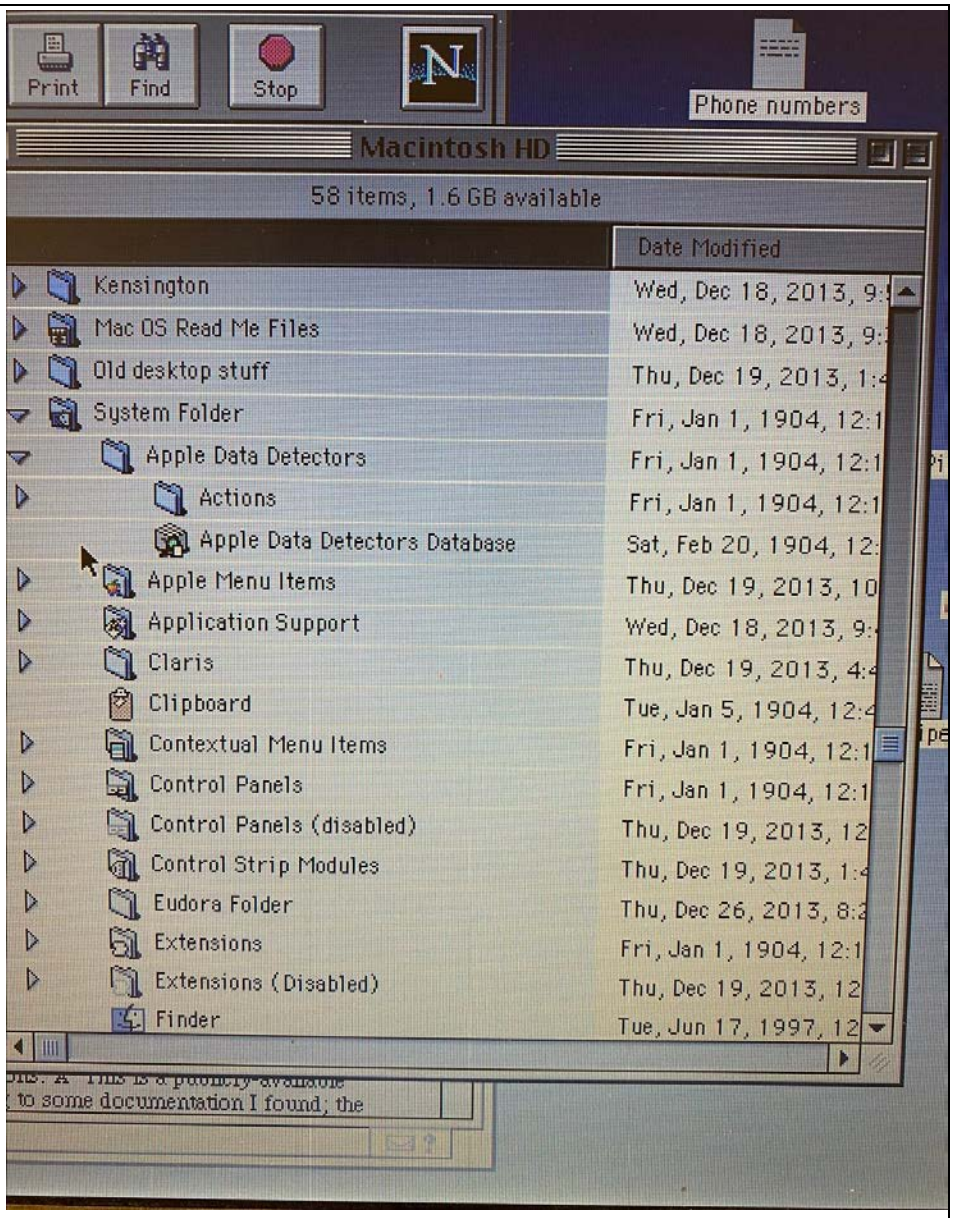

Print
Find
Stop
Phone numbers
Macintosh HD
58 items, 1.6 GB available
Date Modified
Kensington
Mac OS Read Me Files
Old desktop stuff
System Folder
Apple Data Detectors
Actions
Apple Data Detectors Database
Apple Menu Items
Application Support
Claris
Clipboard
Contextual Menu Items
Control Panels
Control Panels (disabled)
Control Strip Modules
Eudora Folder
Extensions
Extensions (Disabled)
Finder
Wed, Dec 18, 2013,
Wed, Dec 18, 2013,
Thu, Dec 19, 2013,
Fri, Jan 1, 1904,
Fri, Jan 1, 1904,
Fri, Jan 1, 1904,
Sat, Feb 20, 1904,
Thu, Dec 19, 2013,
Wed, Dec 18, 2013,
Thu, Dec 19, 2013,
Tue, Jan 5, 1904,
Fri, Jan 1, 1904,
Fri, Jan 1, 1904,
Thu, Dec 19, 2013,
Thu, Dec 19, 2013,
Thu, Dec 26, 2013,
Fri, Jan 1, 1904,
Thu, Dec 19, 2013,
Tue, Jun 17, 1997,
to some documentation I found; the

Exhibit E

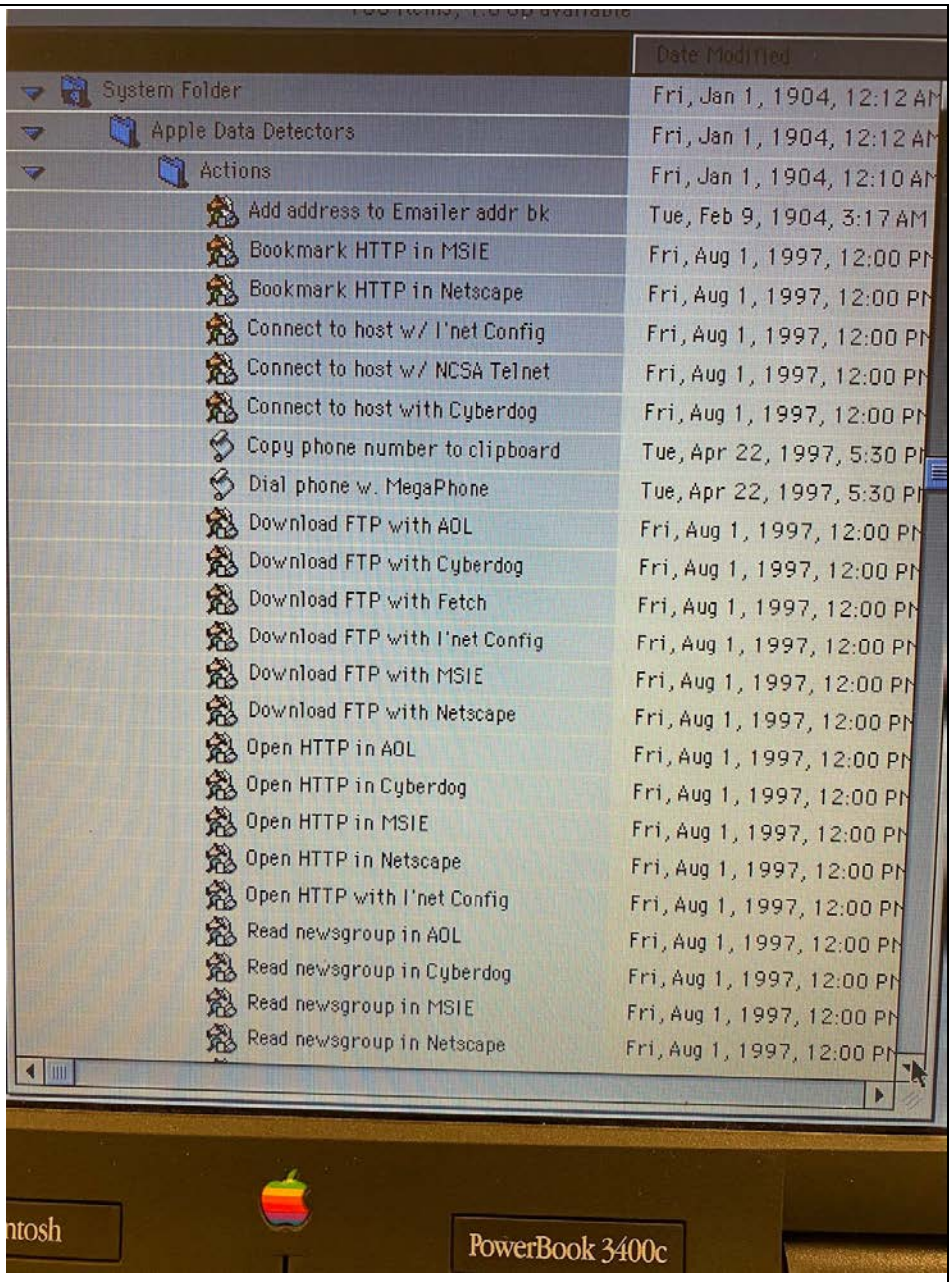
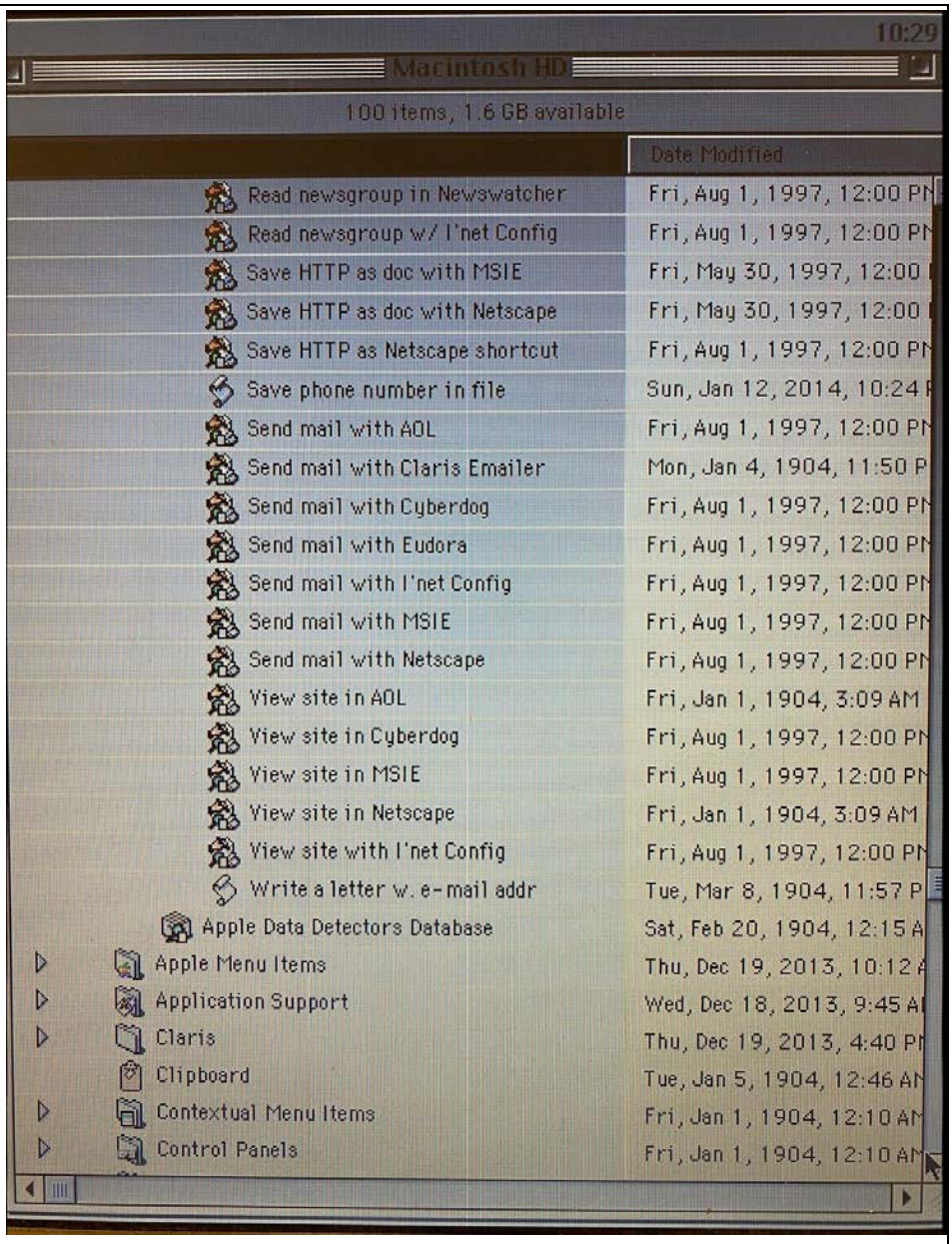

Date Modified
System Folder
Fri, Jan 1, 1904, 12:12 AM
Apple Data Detectors
Fri, Jan 1, 1904, 12:12 AM
Actions
Fri, Jan 1, 1904, 12:10 AM
Add address to Emailer addr bk
Tue, Feb 9, 1904, 3:17 AM
Bookmark HTTP in MSIE
Fri, Aug 1, 1997, 12:00 PM
Bookmark HTTP in Netscape
Fri, Aug 1, 1997, 12:00 PM
Connect to host w/ I'net Config
Fri, Aug 1, 1997, 12:00 PM
Connect to host w/ NCSA Telnet
Fri, Aug 1, 1997, 12:00 PM
Connect to host with Cyberdog
Fri, Aug 1, 1997, 12:00 PM
Copy phone number to clipboard
Tue, Apr 22, 1997, 5:30 PM
Dial phone w. MegaPhone
Tue, Apr 22, 1997, 5:30 PM
Download FTP with AOL
Fri, Aug 1, 1997, 12:00 PM
Download FTP with Cyberdog
Fri, Aug 1, 1997, 12:00 PM
Download FTP with Fetch
Fri, Aug 1, 1997, 12:00 PM
Download FTP with I'net Config
Fri, Aug 1, 1997, 12:00 PM
Download FTP with MSIE
Fri, Aug 1, 1997, 12:00 PM
Download FTP with Netscape
Fri, Aug 1, 1997, 12:00 PM
Open HTTP in AOL
Fri, Aug 1, 1997, 12:00 PM
Open HTTP in Cyberdog
Fri, Aug 1, 1997, 12:00 PM
Open HTTP in MSIE
Fri, Aug 1, 1997, 12:00 PM
Open HTTP in Netscape
Fri, Aug 1, 1997, 12:00 PM
Open HTTP with I'net Config
Fri, Aug 1, 1997, 12:00 PM
Read newsgroup in AOL
Fri, Aug 1, 1997, 12:00 PM
Read newsgroup in Cyberdog
Fri, Aug 1, 1997, 12:00 PM
Read newsgroup in MSIE
Fri, Aug 1, 1997, 12:00 PM
Read newsgroup in Netscape
Fri, Aug 1, 1997, 12:00 PM
PowerBook 3400c

Exhibit E



These actions could be associated with, for example, a detected email address (as shown above), phone number, Newsgroup, HTTP address, or Host.

Exhibit E

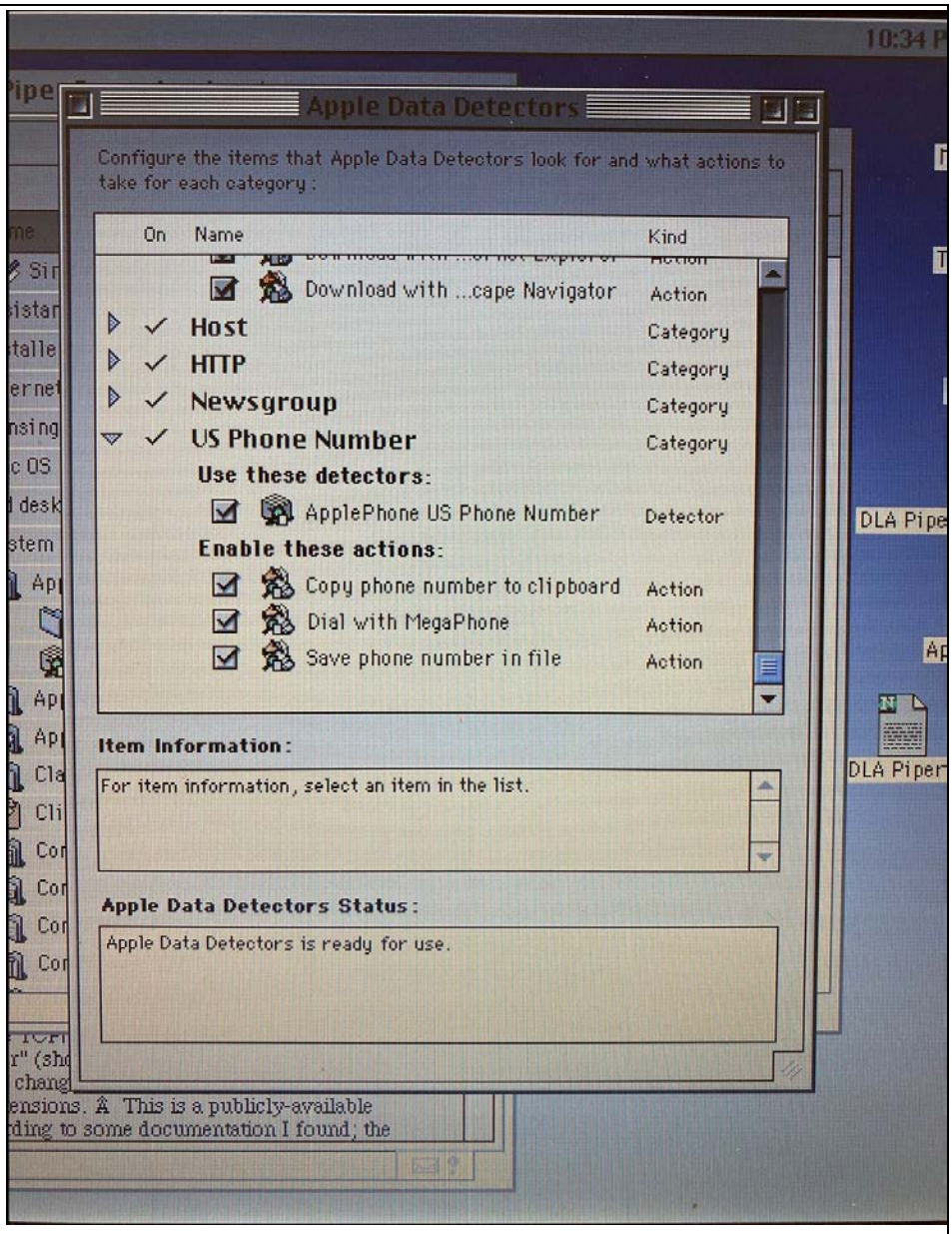
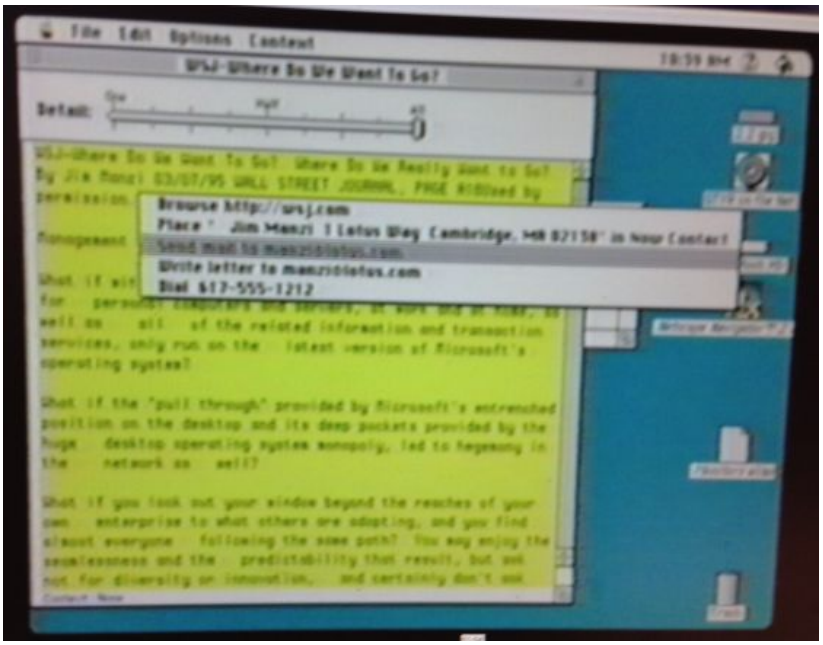
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	The Apple Data Detectors System discloses this element. *See* previous element. *See also* Nardi at 99-101 (discussion of "Architecture and Implementation"). Mac World Data Detectors System could detect data structures in a written document and provide options for the detected structures. For

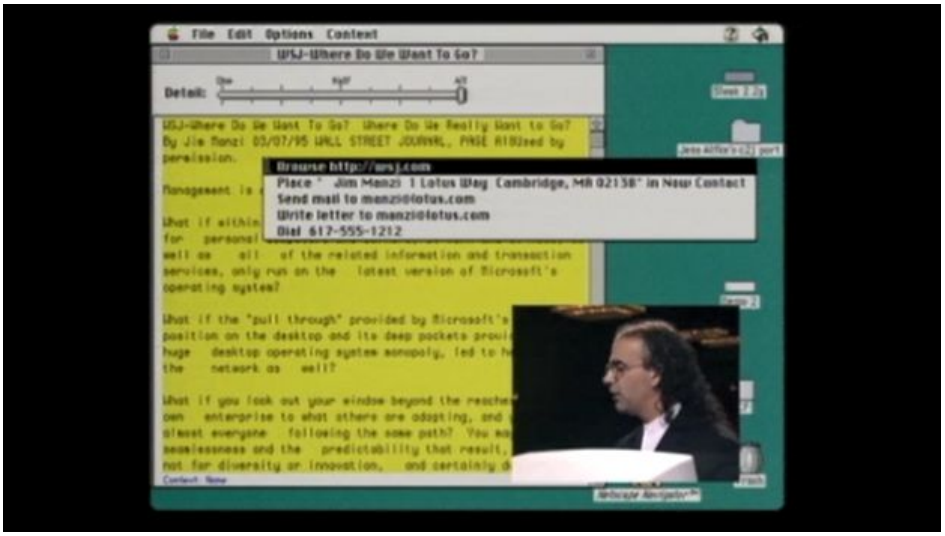
Exhibit E

example:



See Video

See also MacWorld 1997 Expo:



"The analyzer server receives data from a document."

Miller at 2:28.

See also, e.g., 3:8-10 ("FIG. 6 illustrates a window with the identified structures in the example document of FIG. 5 highlighted based on the analyzer server of FIG. 4.").

Exhibit E

	A user can select a whole document or part of a document without having to make a careful selection; the grammars find any embedded structures they know about within the selection and offer an appropriate set of actions from which to choose. (Nardi, page 98). "**on** handle detected data inStructure **try** activate **set** myEmailAddress **to** detected text **of** inStructure addressLetterTo(myEmailAddress)" *See* Write a Letter AppleScript Code. "Am I correct that as we saw on the demo, you could, in fact, select the entire document to have Data Detectors review it? A. Yes. Q. And am I correct that Data Detectors would work even if a user did not specifically designate, for example, the particular phone number that they were interested in acting on? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. Q. Why is it that you wanted to allow users to be able to analyze as much as all of the text in a document? A. Well, it's much easier to just very quickly select a region of text on the screen and let the computer figure out what things can be acted on in it as opposed to having to go in, in this case with a mouse, and very carefully only select the characters that you're interested in. It just requires less precise action on the part of the user. Q. What happens if there were multiple different data objects within the text of the letter, let's say, an email address and a phone number and a map address? Could the Data Detectors detect all of those things? A. Yes, that's what is shown in the figure at the bottom of, I guess, Page 2 of Miller 7. There is a highlighted region of text in the window and the hierarchical menus showing up on the page are showing a phone number and an email address and a URL." Miller Depo. at 71:13-72:20. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you

Exhibit E

	can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4 "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field

Exhibit E

	of the new email." Miller Depo. at 98:24-99:21. "And so Data Detectors in that demonstration, were they being applied to only particular text or to the entire document? A. They were being applied to whatever part of the document had been selected. The yellow color here shows the text that has been selected by the user. I don't know exactly what selection took place, but it's certainly all of the text that is visible on the screen. Q. So everything that would be selected, if it was the entire document that was selected, the entire document would be analyzed? A. Yes. Q. And did the – does the user have to particularly designate somehow a particular piece of information like a telephone number that it wants to have detected? A. Well, Data Detectors will come back with that pop-up menu showing all of the structures that it found in the selected text." Miller Depo. at 119:14-120:8. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing

Exhibit E

	a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "And again, when you're referring to initiating the grammatical analysis of the selected text, what is that referring to? A. That means instructing Data Detectors to run the selected texts through its set of structure recognizers, I guess." Miller Depo. at 165:11-16. "Q. When you say 'pointing at a highlight and pressing the mouse button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button.

Exhibit E

	Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had

Exhibit E

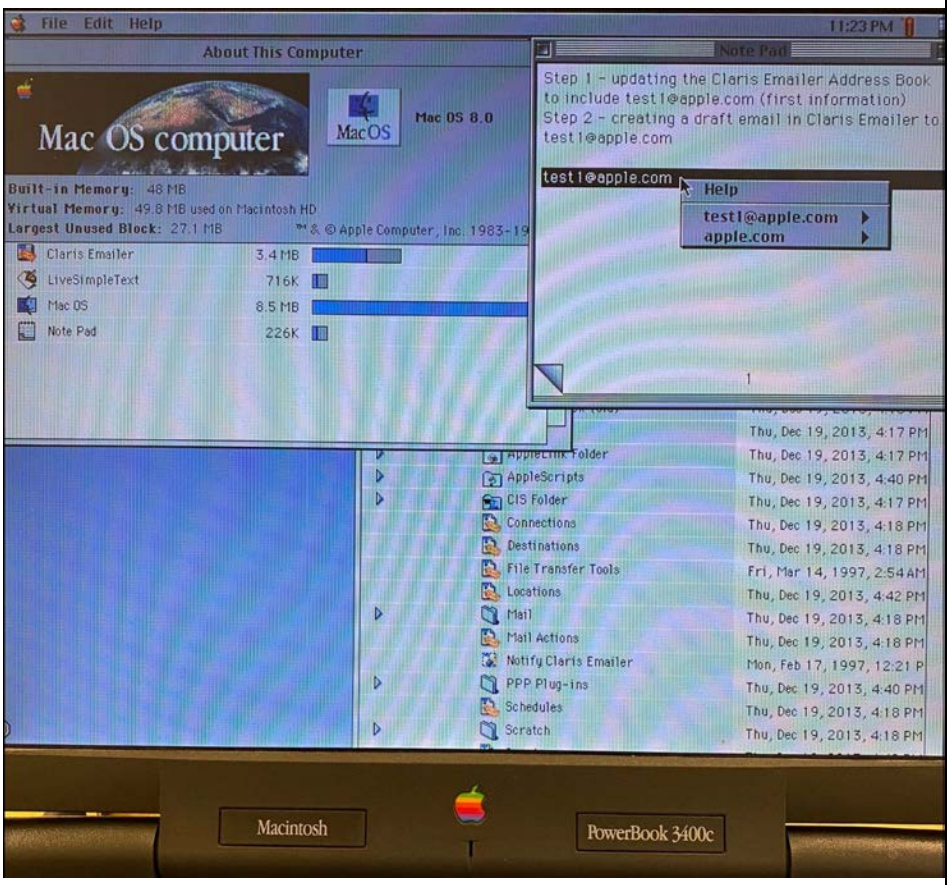
	run with Data Detectors." Miller Depo. at 186:4-23. For example, in the screenshot below, which was taken during my Inspection of the Powerbook 3400C, while the text document was being displayed, the text "test1@apple.com" was highlighted in Note Pad with the mouse cursor and right-clicked. A menu then appeared that included the highlighted "test1@apple.com" text along with a sub-menu of actions associated with the highlighted "test1@apple.com" text. Data Detectors detected that "test1@apple.com" was an email address, and presented actions in the sub-menu associated with an email address. Data Detectors also detected that "apple.com" was an Internet address, and presented actions in the sub-menu associated with an Internet address. 

Exhibit E

	
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	The Apple Data Detectors System discloses this element. When a user selects text and right-clicks in Simple Text, IAD can detect URLs and provide options: 

Exhibit E

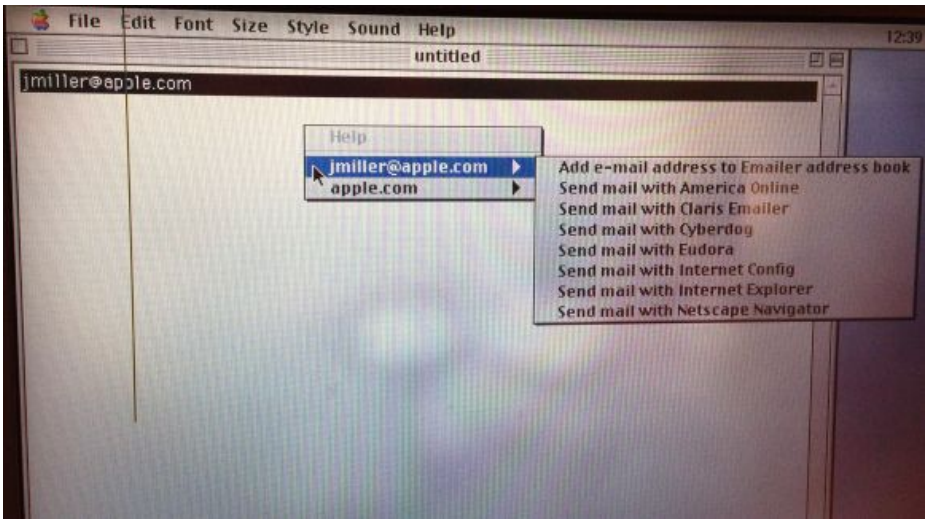
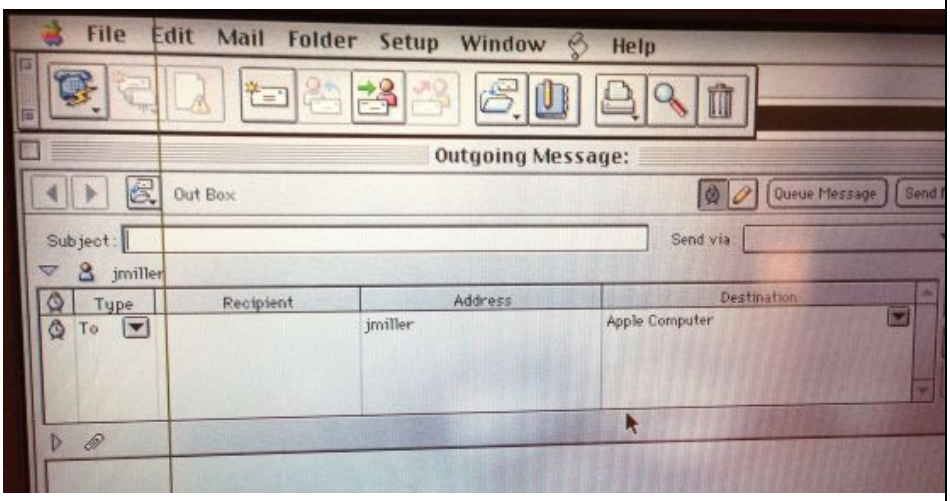
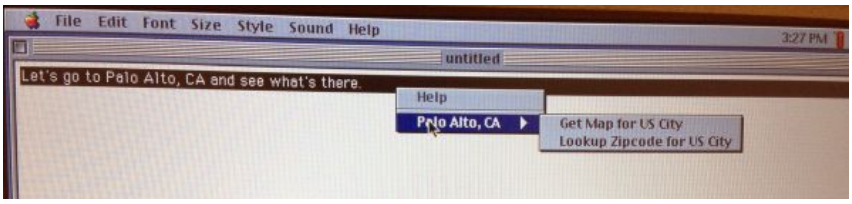
	When a user selects text and right-clicks in Simple Text, IAD can detect email addresses and provide options:  When IAD detects an email address and a user selects "Send mail with Claris Emailer," Claris Emailer will search its information source for a company name associated with the domain of the email address and will insert this into the Outgoing Message panel for a new email. For example:  When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could obtain a zip code for the selected US city, for which the system would

Exhibit E

search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:



This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

See also MacWorld Expo 1997 at 1:11:00:

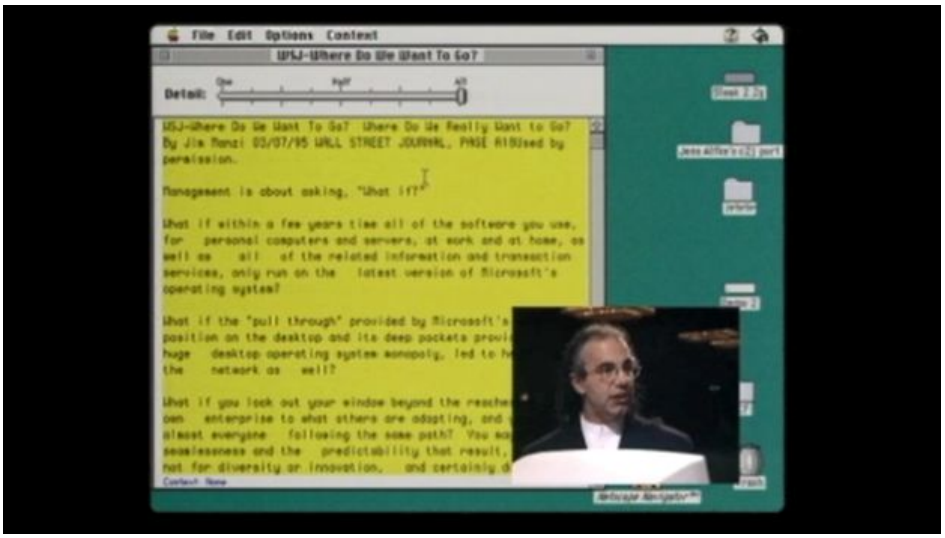
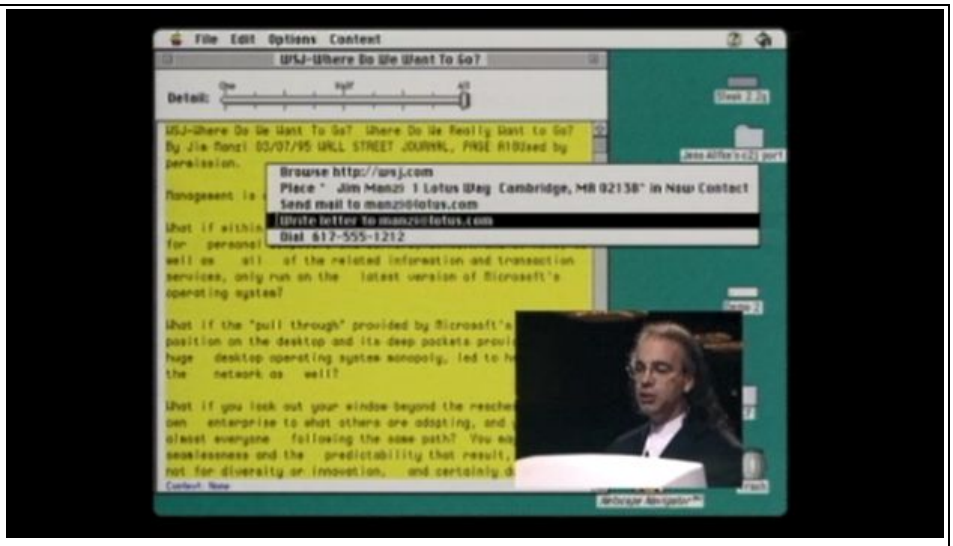
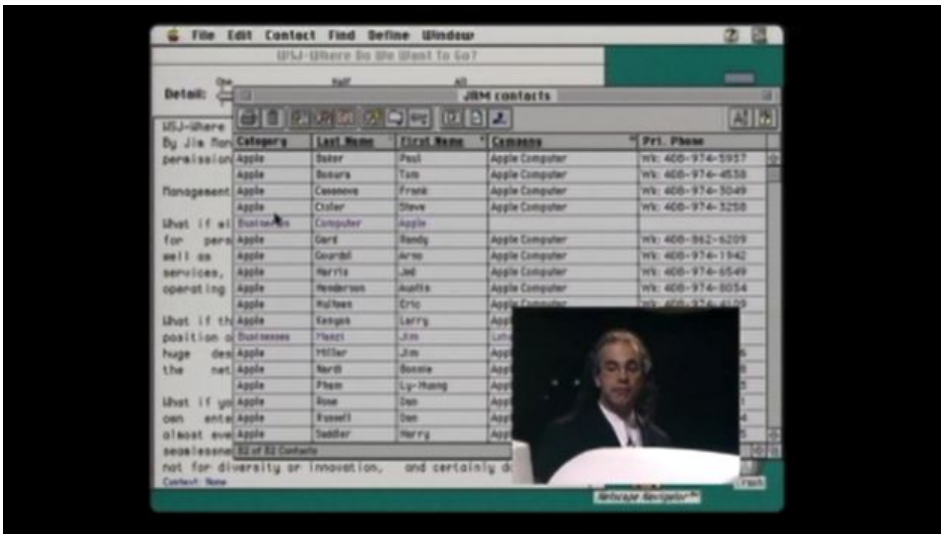


Exhibit E


File Edit Options Content
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212


File Edit Contact Find Define Window
WSJ-Where Do We Want To Go?
Detail:
JBM contacts
Category
Last Name
First Name
Company
Pri. Phone

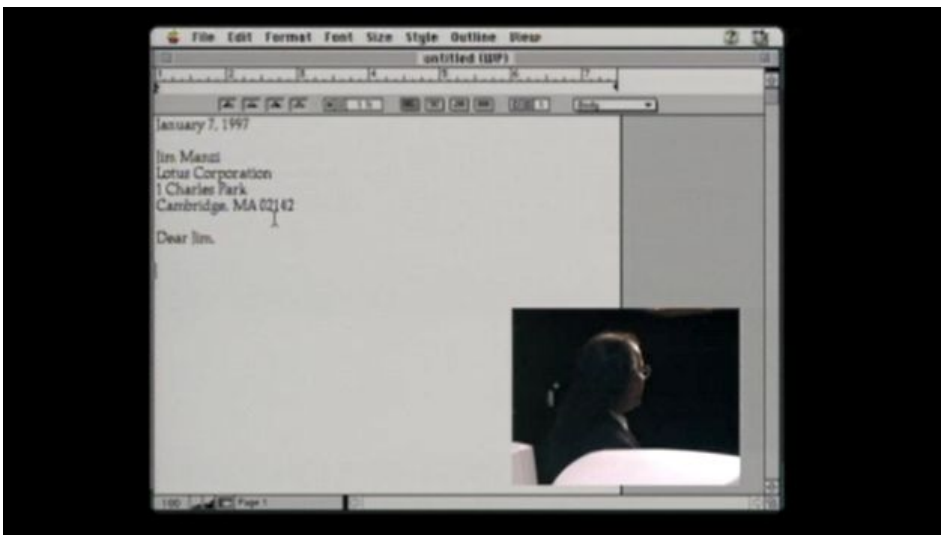

File Edit Format Font Size Style Outline View
untitled (WP)
January 7, 1997
Jim Manzi
Lotus Corporation
1 Charles Park
Cambridge, MA 02142
Dear Jim,

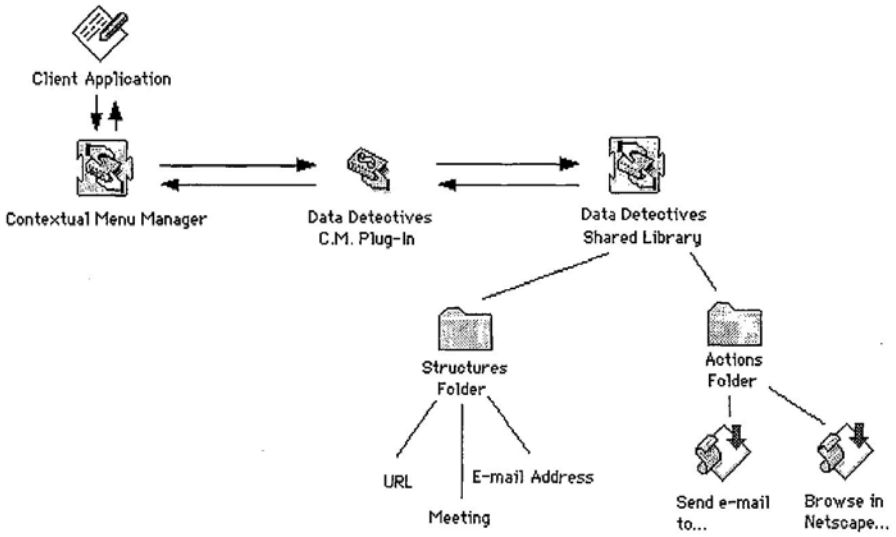
Exhibit E

	See also Nardi at fig. 1: "Two patterns are found: an email address . . . and the announcement of the meeting . . . . These patterns are presented in the pop-up menu; by pointing at the date information in the menu; a second pop-up menu offers a choice of actions . . . . The user can select one, thereby running a small application, or move the cursor off the menu, eliminating the pop-up menu and canceling any actions." See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information." See also Nardi at 98: "User interface. To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1)." See also Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on

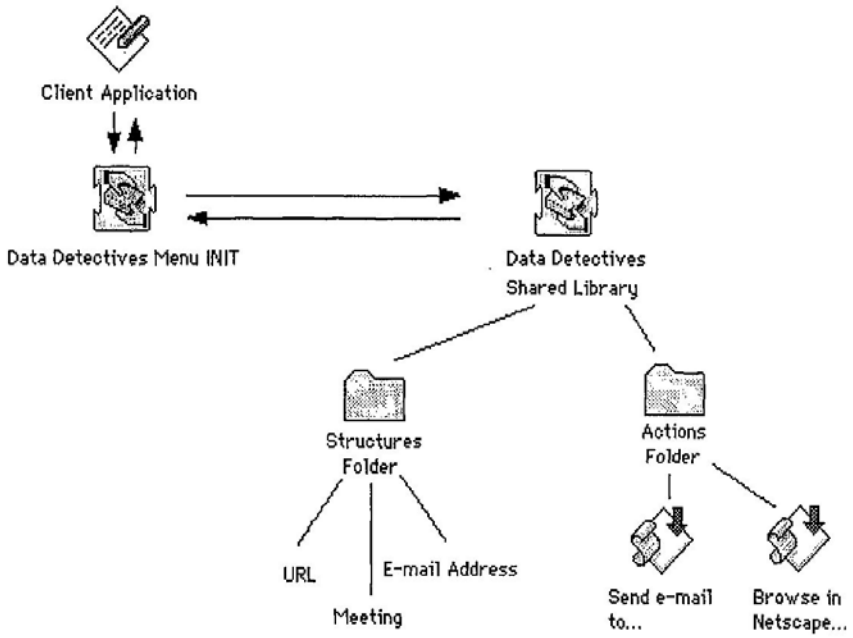
Exhibit E

	a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." See also Nardi at 103: "Our early contact with the product group again served us well. During the system's development, we experimented with various user interfaces to its basic technology, an effort that paid off in a number of ways. The different interfaces followed different assumptions about various aspects of the system, such as the interface techniques themselves, the demand the techniques would place on the underlying operating system, and the demand imposed on developers who wanted to adapt Apple Data Detectors to their own uses. It was important for us to understand such trade-offs, so we could advise the product group on the technology's possibilities." *See also* Nardi at p. 99 ("Figure 4. An action script, demonstrating the generality of Apple Data Detectors' use of a scripting language and external applications as information repositories and as end-user tools. This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a "personal information manager" (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter."). The system could detect an email address, and in response to the "Write letter" command, obtain the mailing address for that person by searching an address book using Now Contact, and address a letter to that person and mailing address in Claris Works. *See* screenshot above, Video at 1:22-1:40. "Find inherently structured data, let user take action on data" *See* WWDC Presentation at 2. *See* WWDC Presentation at 4-5: [Technical Overview - with Contextual Menus in Copland:]

Exhibit E



[Technical Overview - Current]



"**on** addressLetterTo(emailAddress)
set emailAddress **to** removeTextStyling(emailAddress)
tell application "Now Contact 3.5"
launch
activate
run
--find the person in the Now Contact database
try
set thePerson **to the first** person **whose** (customer 2 is emailAddress)

Exhibit E

	on error beep display dialog "Sorry, I don't have any information about " & emailAddress & "." **return** **end try** *--get the address information for the person* **try** **set** firstAndLastName **to** (**the** first name **of** thePerson) & " " & (**the** last name **of** thePerson) **set** theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (the work address **of** thePerson) & return & (**the** work city **of** thePerson) & ", " & (**the** work state **of** thePerson) & " " & (**the** work zip of thePerson) & return **on error** beep display dialog "Sorry, I couldn't get the address of " & emailAddress 7 "." **return** **end try** *--write the address information into ClarisWorks* **tell**application "ClarisWorks" **try** launch activate *--run --don't do a "run here, or you'll get the "New document" dialog…* open (path to system folder as string) & "SD letter template" as "WP" **on error** beep activate display dialog "Sorry, I couldn't open the letter template: 'SD letter template' in the System Folder." *See* Write a Letter AppleScript Code. "An input device 110, such as a keyboard and mouse, and an output device 105, such as a CRT or voice module, are coupled to CPU 120." Miller at 3:26-28.

Exhibit E

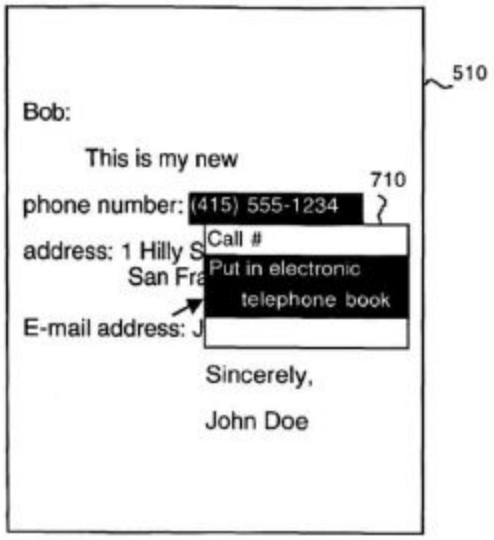
	"As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." Miller at 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

Exhibit E

	Miller at 5:6-5:18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100). "Q. Can you give me a general description of what it is that Apple Data

Exhibit E

	Detectors did? A. The idea at the time was to identify bits of information in user documents. For instance, phone numbers or email addresses or, you know, other sorts of things that could be easily identified and make it very easy for people to carry out actions on them." Miller Depo. at 27:2-9. "And so we were looking for ways to find those things in user documents and then make it easy for people to carry out actions on them, to do things with those, let's say, email addresses that were found." Miller Depo. at 39:23-40:2. "The phrase 'your application' refers to the applications that the attendees of WWDC were building. And the point that we wanted to make was that what we were building could provide a way for users to be able to very easily interact with those applications without having to go through many of the details of pulling down menus off of the menu bar and clicking on things and repositioning the cursor and typing stuff into text fields and clicking submit buttons and all of that. They could just interact directly with the content and then let the software go through all of the tedious parts of actually carrying out actions on that information. Q. And when you say 'input data more easily,' do you recall what that was referring to in your presentation? A. Well, that if we could identify, say, a very long email address and provide a way for someone to act directly on it, then the user would not have to open up a text box and remember and type in that email address, possibly making mistakes along the way. Q. So what sorts of things could be done with the email address like that using the Apple Data Detectives that you're describing in this presentation? A. Well, in particular you could create a new outgoing email message to that email address. You could also put it into an address book so that you could easily use it later. Q. How would that work? Using the Apple Data Detectors or Detectives as they are called in this presentation, how would you actually take an email address in a document and put it into an address book? A. There were application programmatic interfaces, APIs, that would receive requests from outside parts of the computer system. And so the way that this would ordinarily work is that our part of the system would identify this and then send a message through this API channel over to the application and say, 'Make a new email message addressed to this person.' Q. How about to save it in their address book, how would that work?

Exhibit E

	A. There would be another kind of message that could be sent to the application. This one would be 'save this email address inside of your address book.' Q And did you actually build that kind of functionality into the Apple Data Detectors? A. Yes. We would work with applications that could accept those messages and then we would write the bits of code that would take the discovered pieces of information and send them to the application." Miller Depo. at 42:23-45:7. "Then the Data Detectives CM plug-in then exchanges information, it appears, with something called the Data Detectives shared library. Do you see that? A. Yes. Q. Then within that there are two folders, one called "structures folder" and one called "actions folder." Is that right? A. Yes. Q. What is meant by "structures folder"? A. These were the patterns that would recognize different kinds of information. So there would be a component that was able to recognize URLs, http:// followed by a bunch of stuff, or an email address. Q. And I notice in the structures folder, in addition to URLs, it also indicates email address; is that right? A. Yes. Q. And is that something then that you had disclosed that the Apple Data Detectives would be able to recognize as a structure? A. Yes. Q. And I also notice meetings is included in the structure folder. Was that another structure that you disclosed that Apple Data Detectives would be able to identify? A. Yes. Q. Then there's the separate folder called "actions folder." What is that intended to indicate? A. This contains these small bits of code that are able to talk to the applications to carry out the actions. And that after a user had -- or after we had identified a piece of information like an email address, and the user had then indicated I would like to send an email message to that -- to that person, that little application icon would send email to, that would run accept the email address and tell its targeted application to create a new email message." Miller Depo. at 46:23-48:15. "Am I correct that as we saw on the demo, you could, in fact, select the entire document to have Data Detectors review it? A. Yes. Q. And am I correct that Data Detectors would work even if a user did

Exhibit E

	not specifically designate, for example, the particular phone number that they were interested in acting on? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. Q. Why is it that you wanted to allow users to be able to analyze as much as all of the text in a document? A. Well, it's much easier to just very quickly select a region of text on the screen and let the computer figure out what things can be acted on in it as opposed to having to go in, in this case with a mouse, and very carefully only select the characters that you're interested in. It just requires less precise action on the part of the user. Q. What happens if there were multiple different data objects within the text of the letter, let's say, an email address and a phone number and a map address? Could the Data Detectors detect all of those things? A. Yes, that's what is shown in the figure at the bottom of, I guess, Page 2 of Miller 7. There is a highlighted region of text in the window and the hierarchical menus showing up on the page are showing a phone number and an email address and a URL." Miller Depo. at 71:13-72:20. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form.

Exhibit E

	THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4 "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "It was possible to add an address to some contact managers' address book. I believe it was Now Contact." Miller Depo. at 84:18-20. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field

Exhibit E

	of the new email." Miller Depo. at 98:24-99:21. "So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15. "And so Data Detectors in that demonstration, were they being applied to only particular text or to the entire document? A. They were being applied to whatever part of the document had been selected. The yellow color here shows the text that has been selected by the user. I don't know exactly what selection took place, but it's certainly all of the text that is visible on the screen. Q. So everything that would be selected, if it was the entire document that was selected, the entire document would be analyzed? A. Yes. Q. And did the – does the user have to particularly designate somehow a particular piece of information like a telephone number that it wants to have detected? A. Well, Data Detectors will come back with that pop-up menu showing all of the structures that it found in the selected text." Miller Depo. at 119:14-120:8. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next?

Exhibit E

	A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And so looking back at the screenshot that is Number 2 in this Miller Exhibit 14, am I correct that when Lotus – I'm sorry, when Apple Data Detectors are invoked on this document, the user is given, for this particular document, five different options of things that he or show can do? A. Yes. * * * Q. And the user can click any of these with a single click of the mouse? A. Yes." Miller Depo. at 129:17-130:18. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16.

Exhibit E

	"Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12.

Exhibit E

	"LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "Q. When you say 'pointing at a highlight and pressing the mouse button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing.

Exhibit E

	If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example, as observed during my Inspection of the Powerbook 3400C, while the text document was being displayed, the text "test1@apple.com" was highlighted in Note Pad with the mouse cursor and right-clicked. A menu then appeared that included the highlighted "test1@apple.com" text along with a sub-menu of actions associated with the highlighted "test1@apple.com" text. Data Detectors detected that "test1@apple.com" was an email address, and presented actions in the sub-menu associated with an email address. Data Detectors also detected that "apple.com" was an Internet address, and presented actions in the sub-menu associated with an Internet address.

Exhibit E

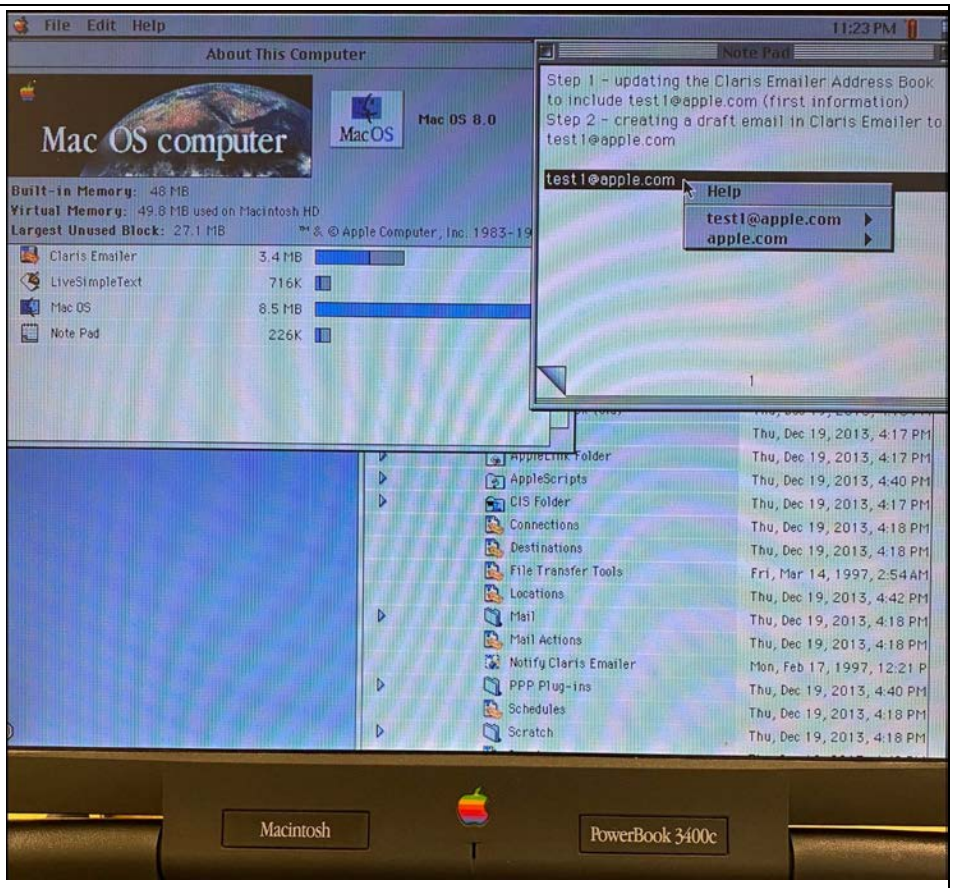

File Edit Help
11:23 PM
About This Computer
Mac OS computer
MacOS
Mac OS 8.0
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 27.1 MB
Claris Emailer 3.4 MB
LiveSimpleText 716K
Mac OS 8.5 MB
Note Pad 226K
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information)
Step 2 - creating a draft email in Claris Emailer to test1@apple.com
test1@apple.com
Help
test1@apple.com
apple.com
AppleScripts
CIS Folder
Connections
Destinations
File Transfer Tools
Locations
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Fri, Mar 14, 1997, 2:54 AM
Thu, Dec 19, 2013, 4:42 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Mon, Feb 17, 1997, 12:21 P
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

Exhibit E

	 For example, when the “Send mail with Claris Emailer” is selected in the sub-menu, the Claris Emailer application is launched, and an “Outgoing Message” window is presented in the Claris Emailer application. In this example, “Apple Computer” is associated with “test1@apple.com,” and is displayed in the “Outgoing Message” window.

Exhibit E

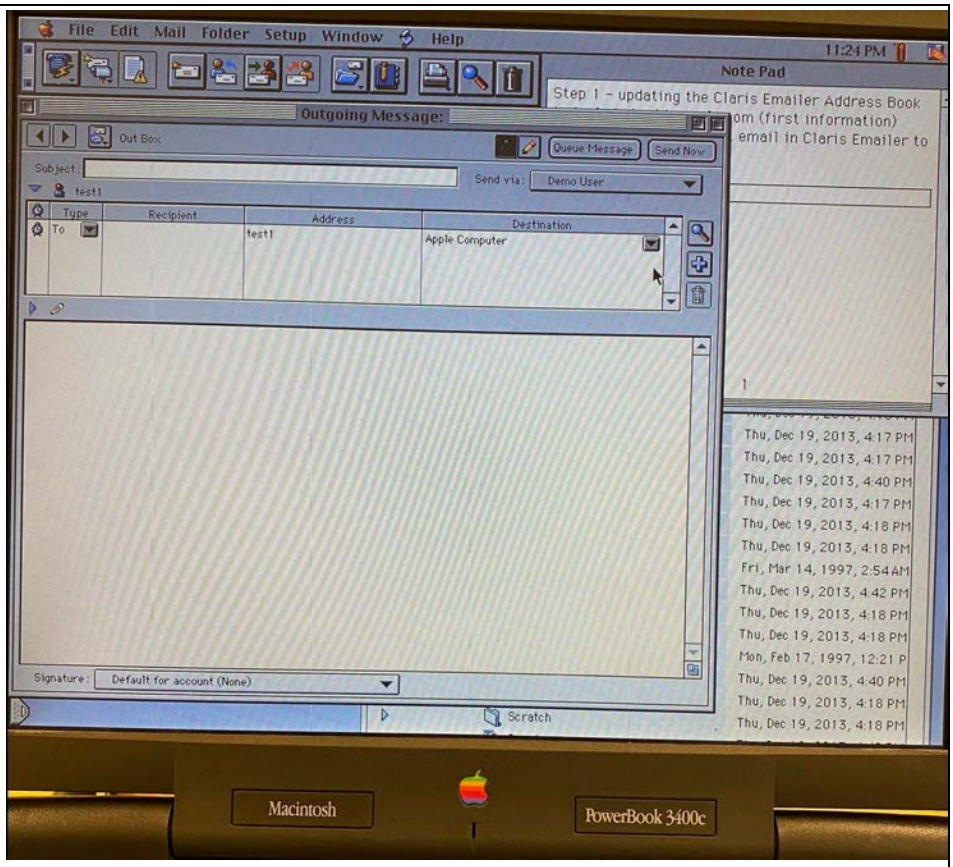
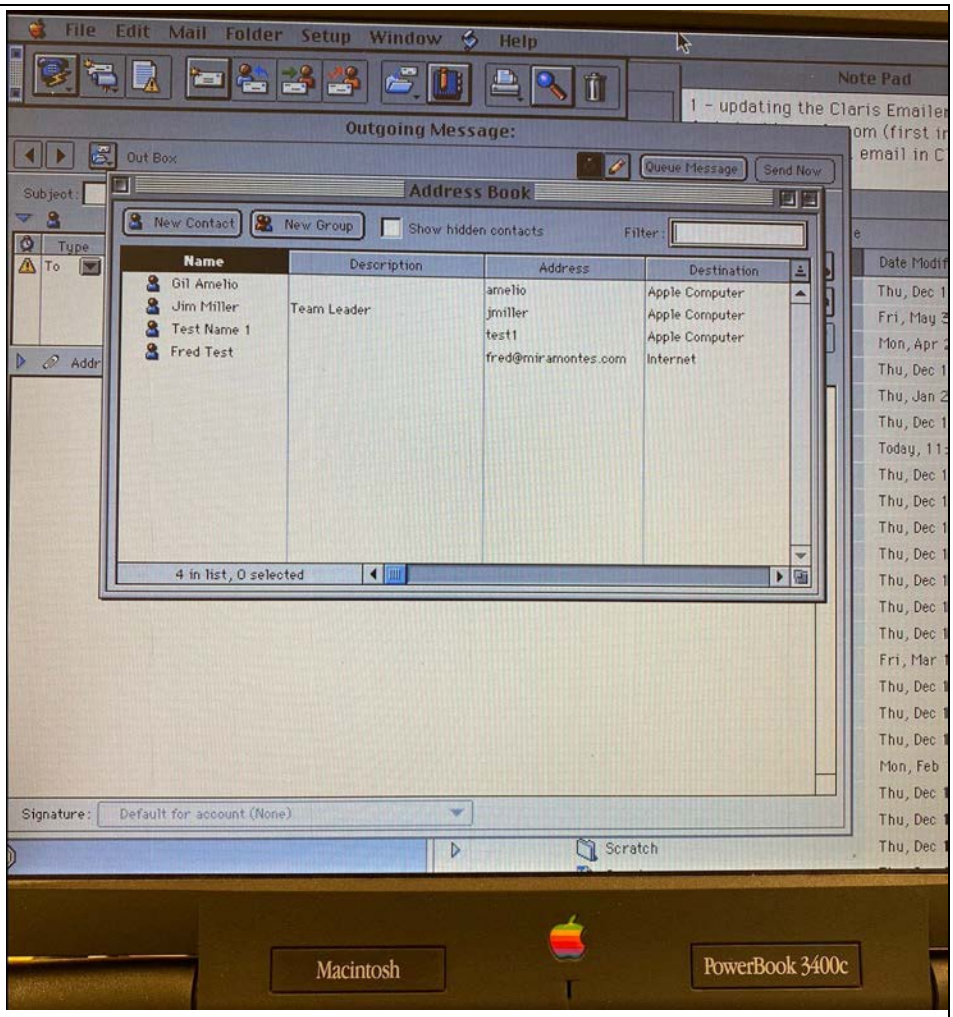
	 I then observed that the Claris Emailer "Address Book," contained an association of "Apple Computer" (in the "Destination" column for the "Test Name 1" entry) with "test1@apple.com."

Exhibit E



Indeed, prior to the aforementioned sequence of events observed during my Inspection, an entry was added in the Claris Emailer Address Book to include the email address "test1@apple.com" to the Claris Emailer Address Book, and was associated with the user name "Test Name 1."

Exhibit E

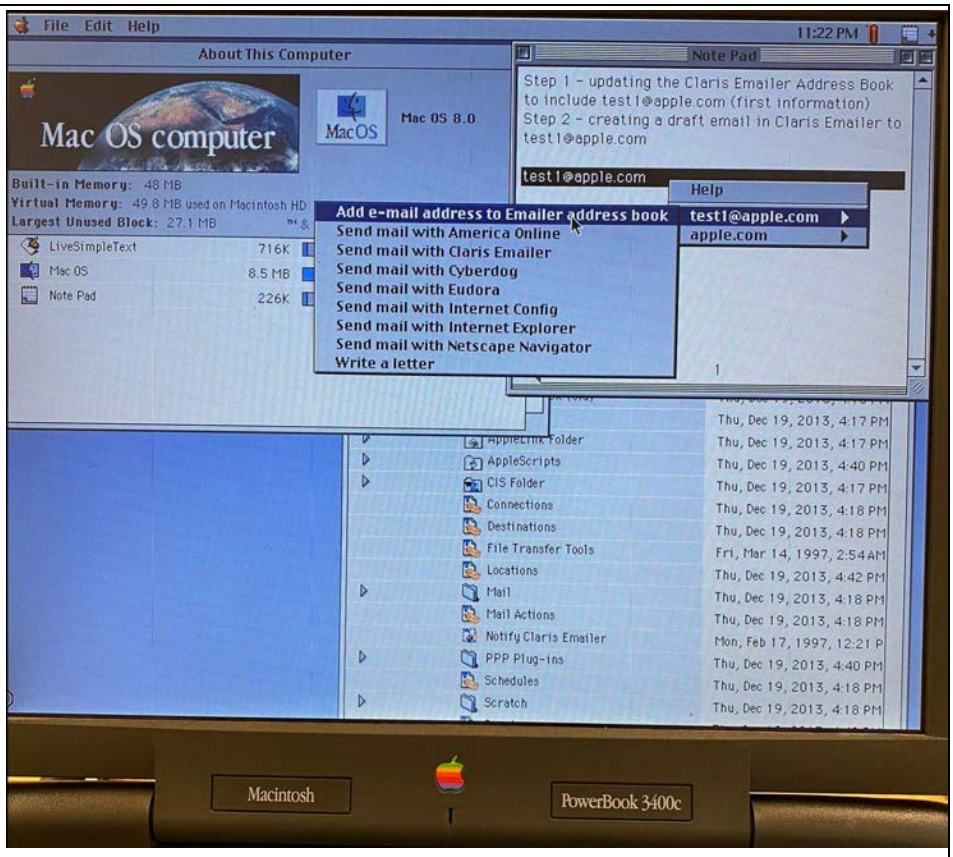
	 File Edit Help 11:22 PM About This Computer Mac OS computer MacOS Mac OS 8.0 Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 27.1 MB LiveSimpleText 716K Mac OS 8.5 MB Note Pad 226K Note Pad Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information) Step 2 - creating a draft email in Claris Emailer to test1@apple.com test1@apple.com Help Add e-mail address to Emailer address book Send mail with America Online Send mail with Claris Emailer Send mail with Cyberdog Send mail with Eudora Send mail with Internet Config Send mail with Internet Explorer Send mail with Netscape Navigator Write a letter test1@apple.com apple.com AppleScripts CIS Folder Connections Destinations File Transfer Tools Locations Mail Mail Actions Notify Claris Emailer PPP Plug-ins Schedules Scratch Thu, Dec 19, 2013, 4:17 PM Thu, Dec 19, 2013, 4:17 PM Thu, Dec 19, 2013, 4:40 PM Thu, Dec 19, 2013, 4:17 PM Thu, Dec 19, 2013, 4:18 PM Thu, Dec 19, 2013, 4:18 PM Fri, Mar 14, 1997, 2:54 AM Thu, Dec 19, 2013, 4:42 PM Thu, Dec 19, 2013, 4:18 PM Thu, Dec 19, 2013, 4:18 PM Mon, Feb 17, 1997, 12:21 P Thu, Dec 19, 2013, 4:40 PM Thu, Dec 19, 2013, 4:18 PM Thu, Dec 19, 2013, 4:18 PM Macintosh PowerBook 3400c

Exhibit E

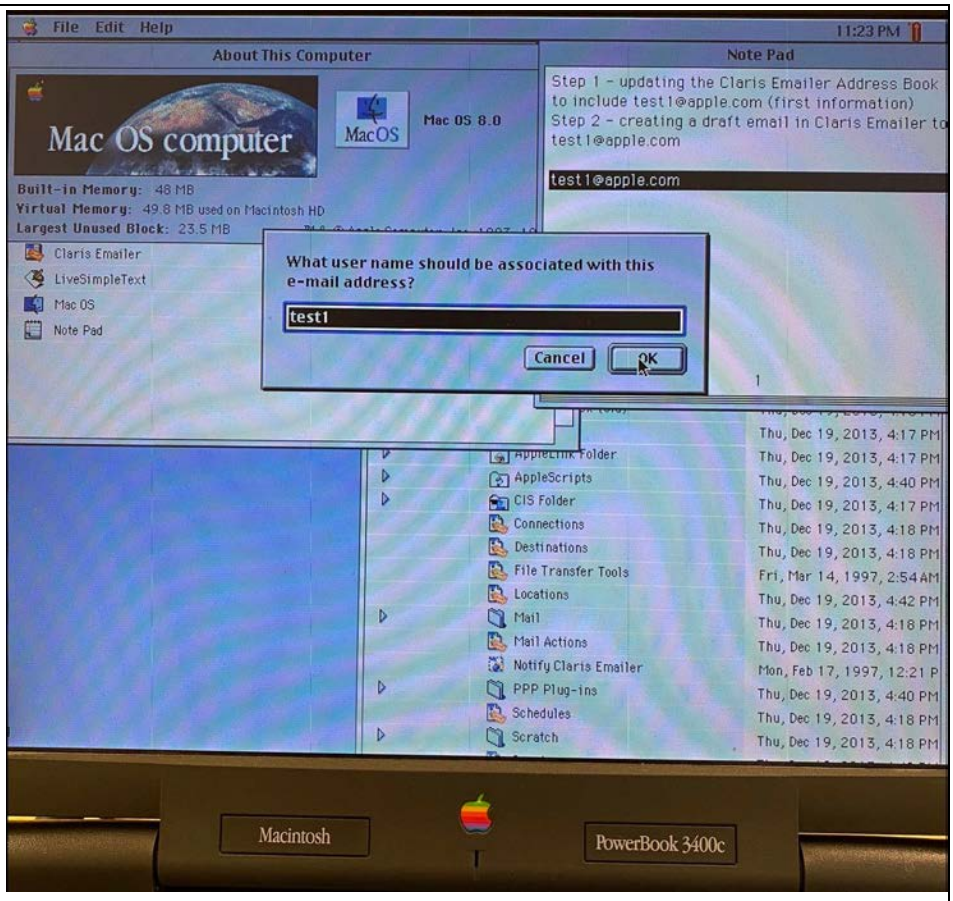

File Edit Help
11:23 PM
About This Computer
Mac OS computer
MacOS
Mac OS 8.0
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.5 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
What user name should be associated with this e-mail address?
test1
Cancel
OK
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information)
Step 2 - creating a draft email in Claris Emailer to test1@apple.com
test1@apple.com
AppleScripts
CIS Folder
Connections
Destinations
File Transfer Tools
Locations
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Fri, Mar 14, 1997, 2:54 AM
Thu, Dec 19, 2013, 4:42 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Mon, Feb 17, 1997, 12:21 P
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

Exhibit E

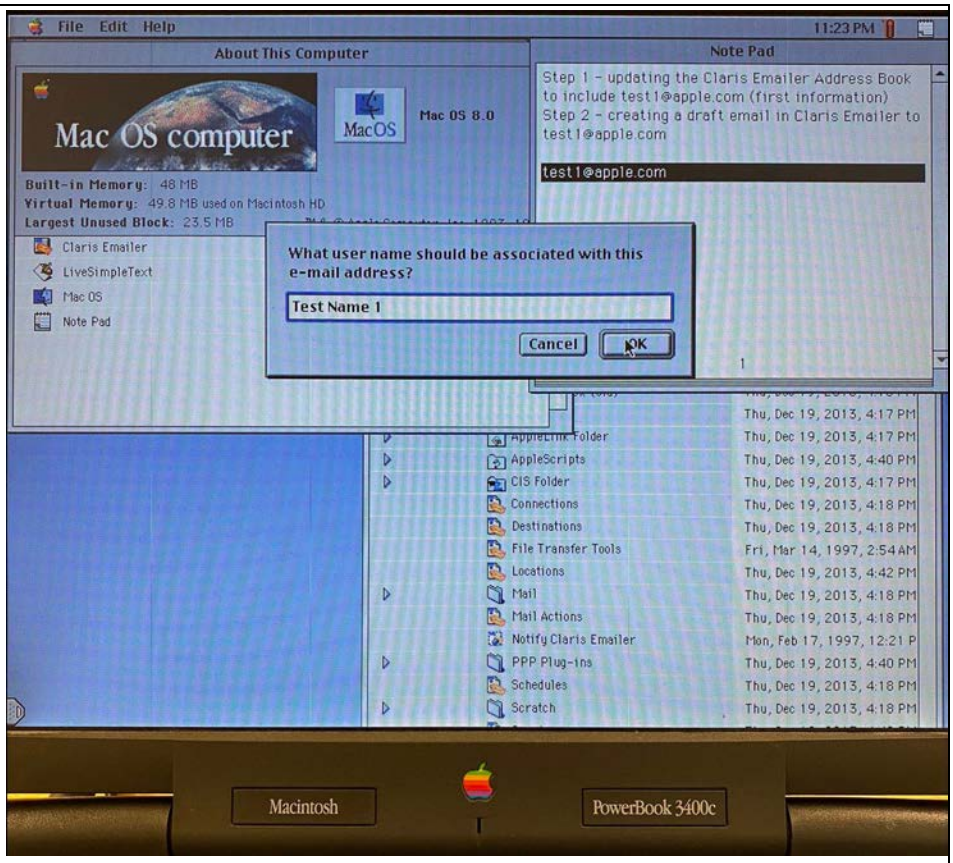
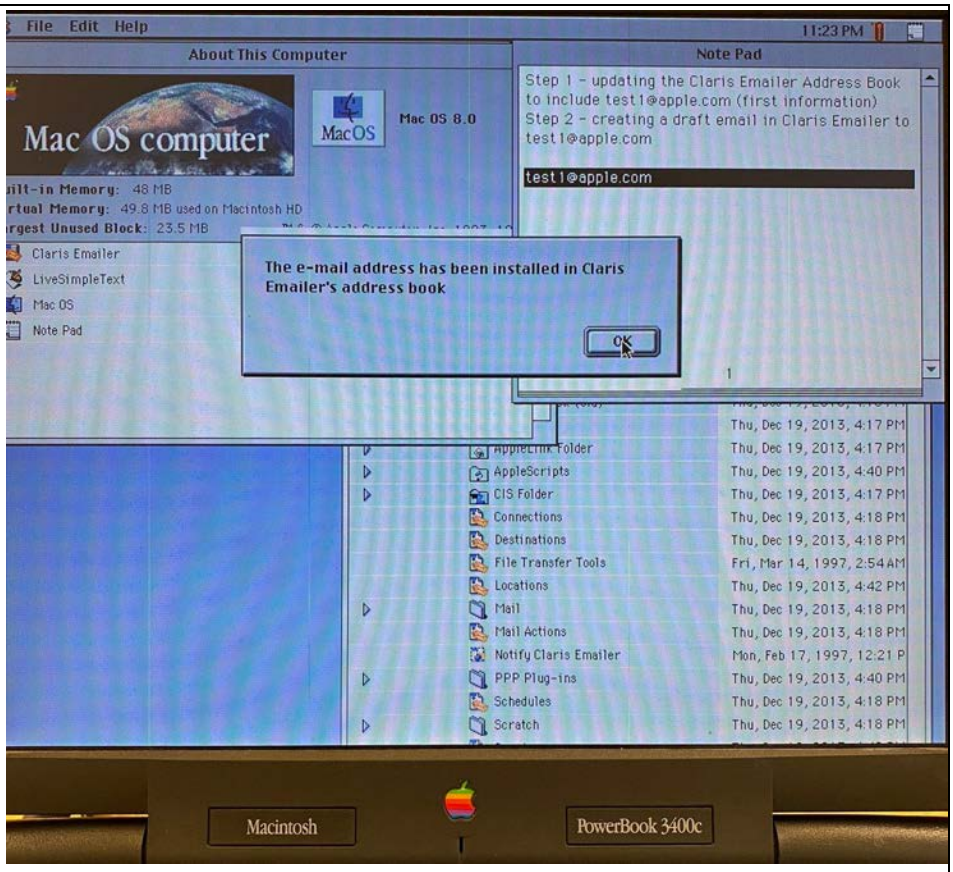
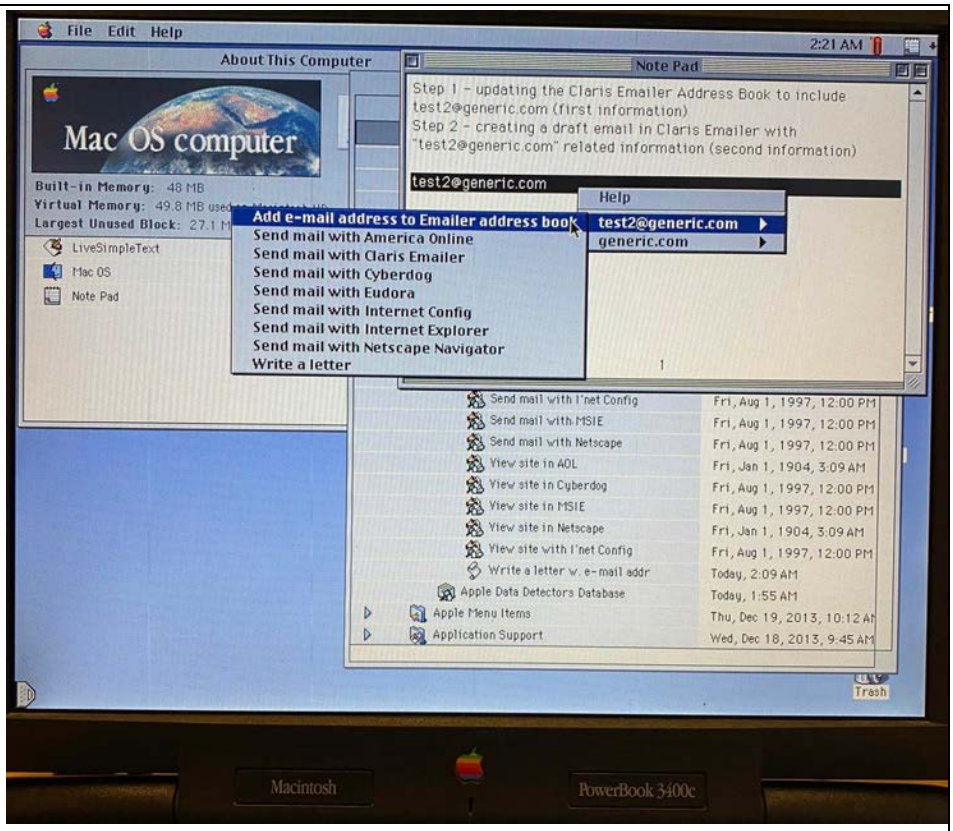
	 File Edit Help 11:23 PM About This Computer Mac OS computer MacOS Mac OS 8.0 Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 23.5 MB Claris Emailer LiveSimpleText Mac OS Note Pad Note Pad Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information) Step 2 - creating a draft email in Claris Emailer to test1@apple.com test1@apple.com What user name should be associated with this e-mail address? Test Name 1 Cancel OK AppleScripts Thu, Dec 19, 2013, 4:40 PM CIS Folder Thu, Dec 19, 2013, 4:17 PM Connections Thu, Dec 19, 2013, 4:18 PM Destinations Thu, Dec 19, 2013, 4:18 PM File Transfer Tools Fri, Mar 14, 1997, 2:54 AM Locations Thu, Dec 19, 2013, 4:42 PM Mail Thu, Dec 19, 2013, 4:18 PM Mail Actions Thu, Dec 19, 2013, 4:18 PM Notify Claris Emailer Mon, Feb 17, 1997, 12:21 P PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM Schedules Thu, Dec 19, 2013, 4:18 PM Scratch Thu, Dec 19, 2013, 4:18 PM Macintosh PowerBook 3400c

Exhibit E



As another example observed during my Inspection, the email address "test2@generic.com" was added to the Claris Emailer Address Book, and associated with the user name "Test Name 2." The text "test2@generic.com" was then highlighted and right-clicked in the Note Pad application, which presented a menu including "test2@generic.com" and "generic.com." Hovering with a mouse over the "test2@generic.com" item yielded a sub-menu including a number of actions associated with an email address. Selection of "Send mail with Claris Emailer" launched the Claris Emailer application, and an "Outgoing Message" window was presented in the Claris Emailer application. In this example, "Internet" is associated with "test2@generic.com," and is displayed in the "Outgoing Message" window.

Exhibit E


File Edit Help
2:21 AM
About This Computer
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB
Largest Unused Block: 27.1 M
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test2@generic.com (first information)
Step 2 - creating a draft email in Claris Emailer with "test2@generic.com" related information (second information)
test2@generic.com
Help
test2@generic.com
generic.com
Add e-mail address to Emailer address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with Internet Explorer
Send mail with Netscape Navigator
Write a letter
Send mail with I'net Config Fri, Aug 1, 1997, 12:00 PM
Send mail with MSIE Fri, Aug 1, 1997, 12:00 PM
Send mail with Netscape Fri, Aug 1, 1997, 12:00 PM
View site in AOL Fri, Jan 1, 1904, 3:09 AM
View site in Cyberdog Fri, Aug 1, 1997, 12:00 PM
View site in MSIE Fri, Aug 1, 1997, 12:00 PM
View site in Netscape Fri, Jan 1, 1904, 3:09 AM
View site with I'net Config Fri, Aug 1, 1997, 12:00 PM
Write a letter w. e-mail addr Today, 2:09 AM
Apple Data Detectors Database Today, 1:55 AM
Apple Menu Items Thu, Dec 19, 2013, 10:12 AM
Application Support Wed, Dec 18, 2013, 9:45 AM
Trash
Macintosh
PowerBook 3400c

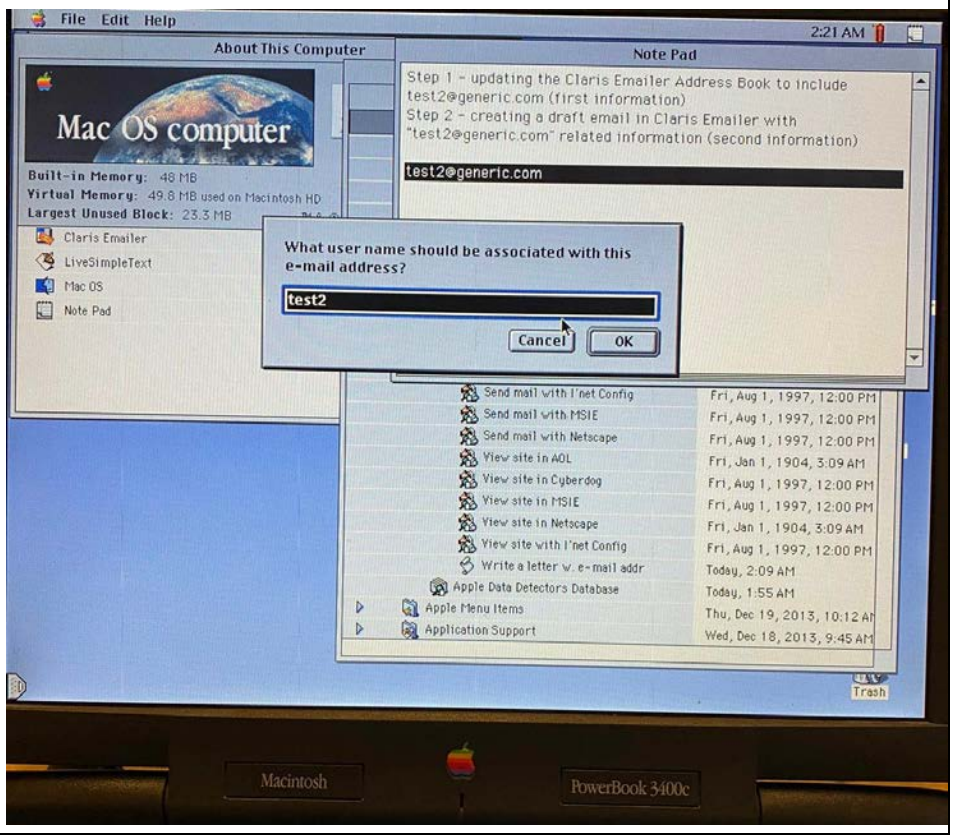
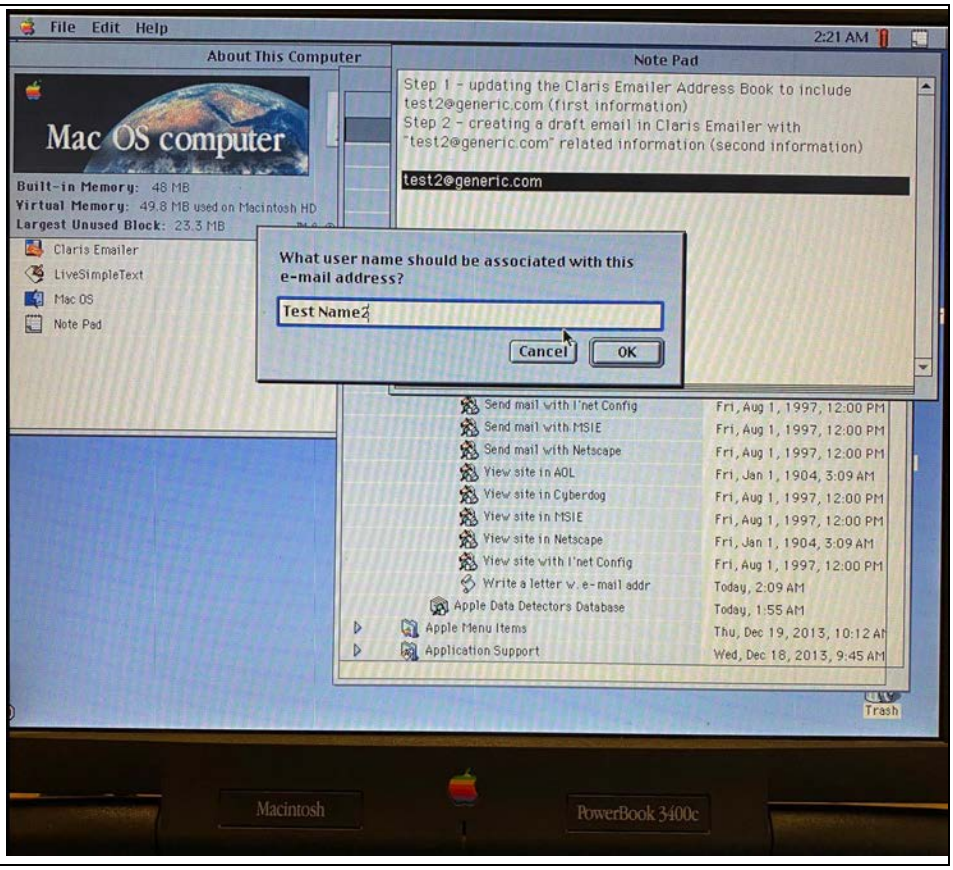
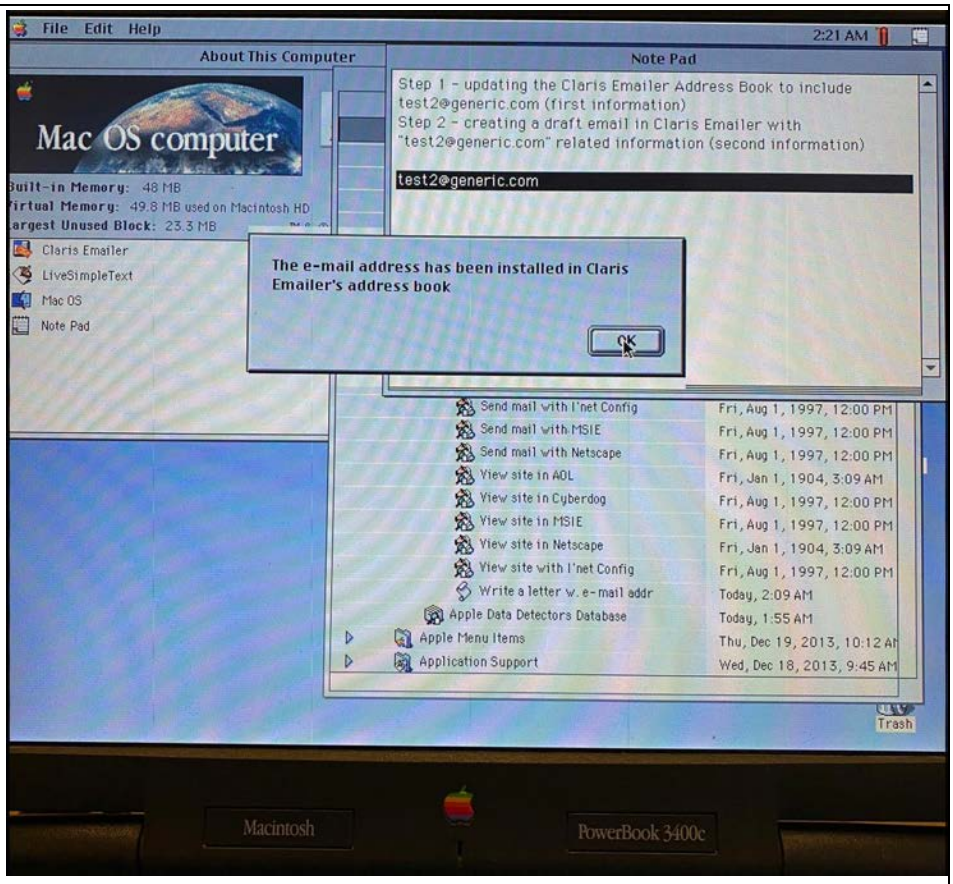

File Edit Help
2:21 AM
About This Computer
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.3 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test2@generic.com (first information)
Step 2 - creating a draft email in Claris Emailer with "test2@generic.com" related information (second information)
test2@generic.com
What user name should be associated with this e-mail address?
test2
Cancel
OK
Send mail with I'net Config Fri, Aug 1, 1997, 12:00 PM
Send mail with MSIE Fri, Aug 1, 1997, 12:00 PM
Send mail with Netscape Fri, Aug 1, 1997, 12:00 PM
View site in AOL Fri, Jan 1, 1904, 3:09 AM
View site in Cyberdog Fri, Aug 1, 1997, 12:00 PM
View site in MSIE Fri, Aug 1, 1997, 12:00 PM
View site in Netscape Fri, Jan 1, 1904, 3:09 AM
View site with I'net Config Fri, Aug 1, 1997, 12:00 PM
Write a letter w. e-mail addr Today, 2:09 AM
Apple Data Detectors Database Today, 1:55 AM
Apple Menu Items Thu, Dec 19, 2013, 10:12 AM
Application Support Wed, Dec 18, 2013, 9:45 AM
Trash
Macintosh
PowerBook 3400c

Exhibit E

File Edit Help
2:21 AM
About This Computer
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.3 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test2@generic.com (first information)
Step 2 - creating a draft email in Claris Emailer with "test2@generic.com" related information (second information)
test2@generic.com
What user name should be associated with this e-mail address?
Test Name2
Cancel
OK
Send mail with I'net Config Fri, Aug 1, 1997, 12:00 PM
Send mail with MSIE Fri, Aug 1, 1997, 12:00 PM
Send mail with Netscape Fri, Aug 1, 1997, 12:00 PM
View site in AOL Fri, Jan 1, 1904, 3:09 AM
View site in Cyberdog Fri, Aug 1, 1997, 12:00 PM
View site in MSIE Fri, Aug 1, 1997, 12:00 PM
View site in Netscape Fri, Jan 1, 1904, 3:09 AM
View site with I'net Config Fri, Aug 1, 1997, 12:00 PM
Write a letter w. e-mail addr Today, 2:09 AM
Apple Data Detectors Database Today, 1:55 AM
Apple Menu Items Thu, Dec 19, 2013, 10:12 AM
Application Support Wed, Dec 18, 2013, 9:45 AM
Trash
Macintosh
PowerBook 3400c

Exhibit E


File Edit Help
2:21 AM
About This Computer
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.3 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test2@generic.com (first information)
Step 2 - creating a draft email in Claris Emailer with "test2@generic.com" related information (second information)
test2@generic.com
The e-mail address has been installed in Claris Emailer's address book
OK
Send mail with I'net Config Fri, Aug 1, 1997, 12:00 PM
Send mail with MSIE Fri, Aug 1, 1997, 12:00 PM
Send mail with Netscape Fri, Aug 1, 1997, 12:00 PM
View site in AOL Fri, Jan 1, 1904, 3:09 AM
View site in Cyberdog Fri, Aug 1, 1997, 12:00 PM
View site in MSIE Fri, Aug 1, 1997, 12:00 PM
View site in Netscape Fri, Jan 1, 1904, 3:09 AM
View site with I'net Config Fri, Aug 1, 1997, 12:00 PM
Write a letter w. e-mail addr Today, 2:09 AM
Apple Data Detectors Database Today, 1:55 AM
Apple Menu Items Thu, Dec 19, 2013, 10:12 AM
Application Support Wed, Dec 18, 2013, 9:45 AM
Trash
Macintosh
PowerBook 3400c

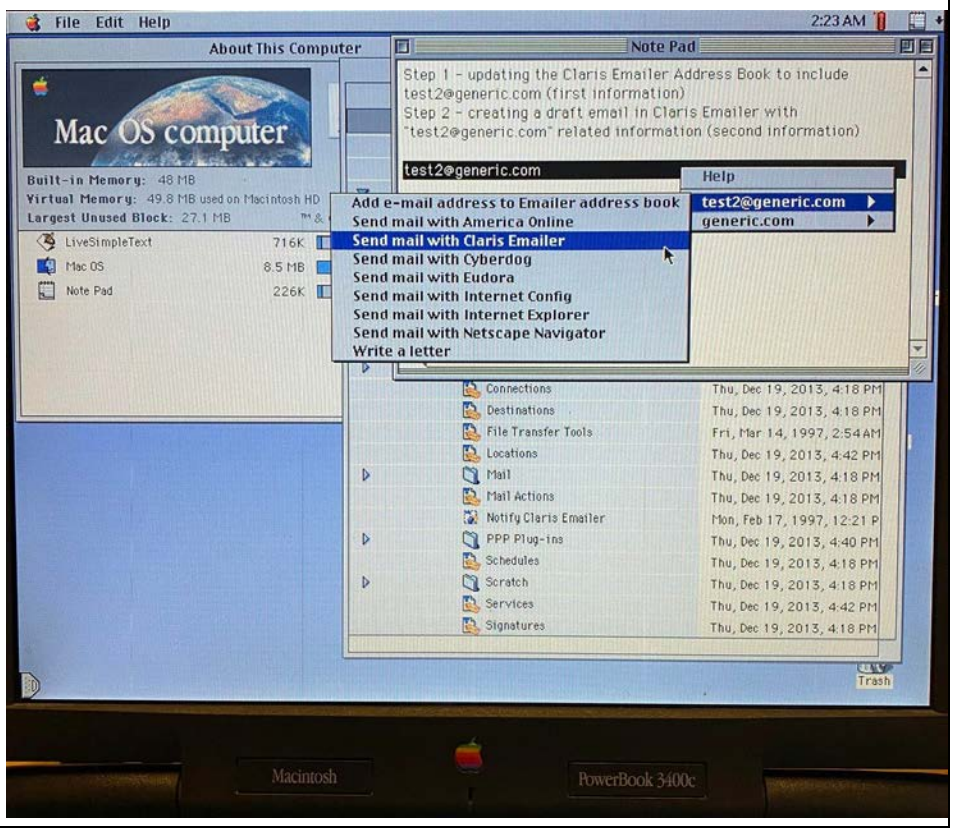
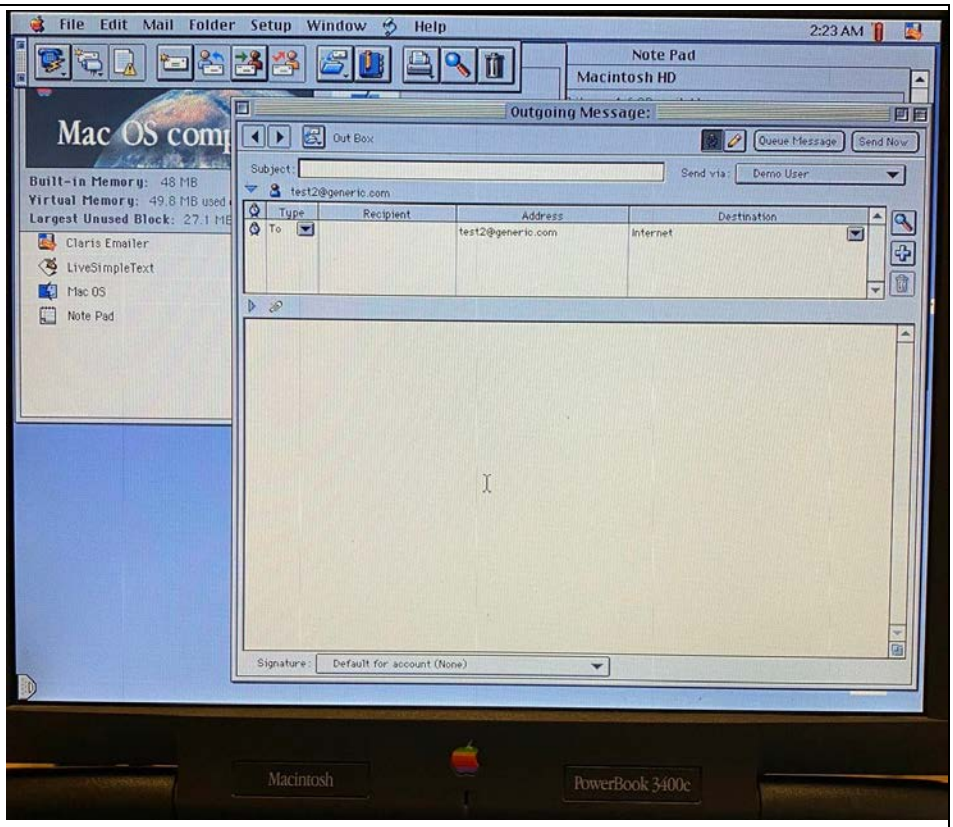

File Edit Help
2:23 AM
About This Computer
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 27.1 MB
LiveSimpleText 716K
Mac OS 8.5 MB
Note Pad 226K
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test2@generic.com (first information)
Step 2 - creating a draft email in Claris Emailer with "test2@generic.com" related information (second information)
test2@generic.com
Help
Add e-mail address to Emailer address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with Internet Explorer
Send mail with Netscape Navigator
Write a letter
test2@generic.com
generic.com
Connections Thu, Dec 19, 2013, 4:18 PM
Destinations Thu, Dec 19, 2013, 4:18 PM
File Transfer Tools Fri, Mar 14, 1997, 2:54 AM
Locations Thu, Dec 19, 2013, 4:42 PM
Mail Thu, Dec 19, 2013, 4:18 PM
Mail Actions Thu, Dec 19, 2013, 4:18 PM
Notify Claris Emailer Mon, Feb 17, 1997, 12:21 PM
PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM
Schedules Thu, Dec 19, 2013, 4:18 PM
Scratch Thu, Dec 19, 2013, 4:18 PM
Services Thu, Dec 19, 2013, 4:42 PM
Signatures Thu, Dec 19, 2013, 4:18 PM
Trash
Macintosh
PowerBook 3400c

Exhibit E


File Edit Mail Folder Setup Window Help
2:23 AM
Note Pad
Macintosh HD
Outgoing Message:
Out Box
Queue Message
Send Now
Subject:
Send via:
Demo User
test2@generic.com
Type
Recipient
Address
Destination
To
test2@generic.com
Internet
Signature:
Default for account (None)
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used
Largest Unused Block: 27.1 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Macintosh
PowerBook 3400c

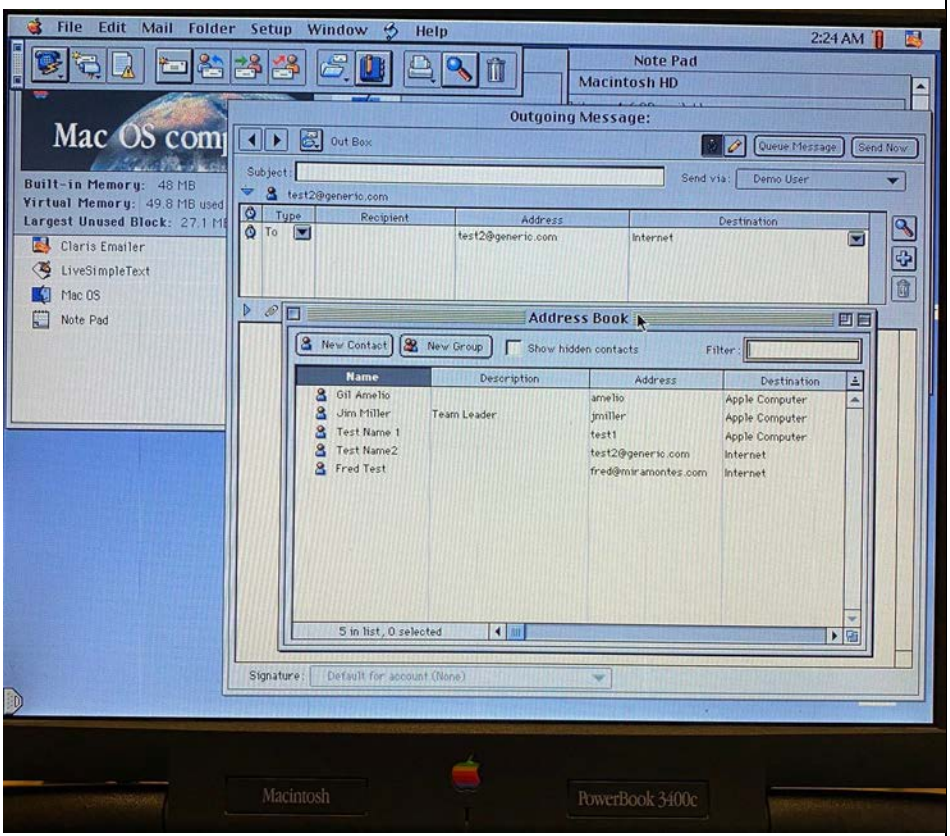
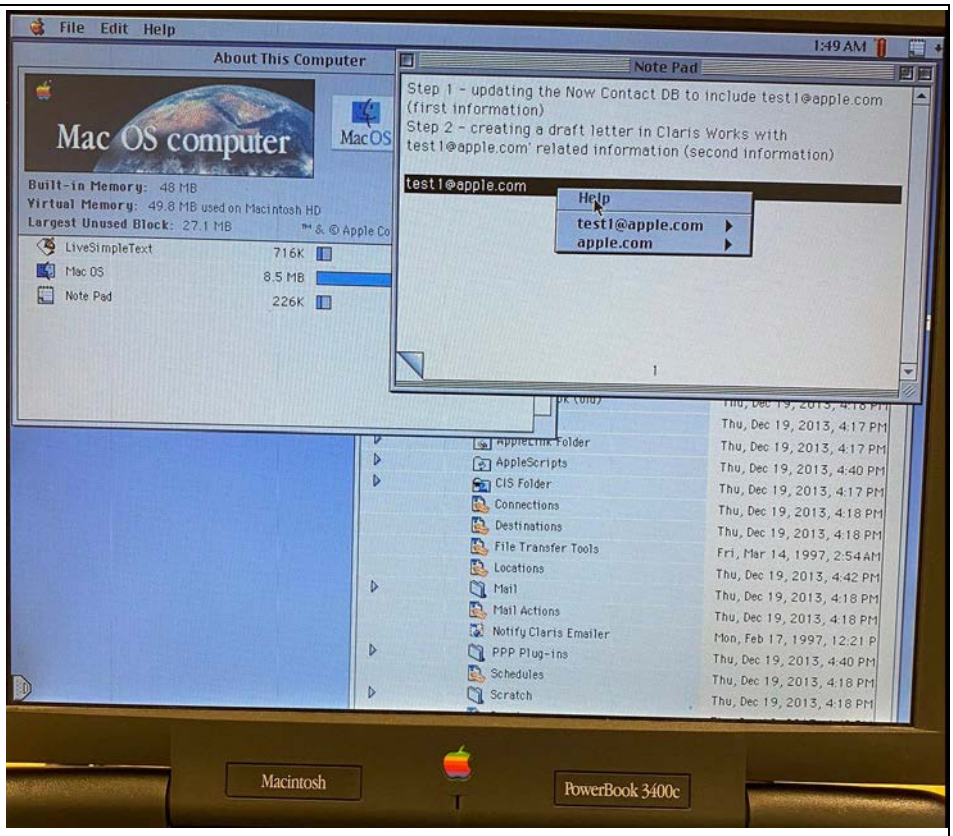

File Edit Mail Folder Setup Window Help
2:24 AM
Note Pad
Macintosh HD
Outgoing Message:
Out Box
Queue Message
Send Now
Subject:
Send via:
Demo User
test2@generic.com
Type
Recipient
Address
Destination
To
test2@generic.com
Internet
Address Book
New Contact
New Group
Show hidden contacts
Filter:
Name
Description
Address
Destination
Gil Amelio
amelio
Apple Computer
Jim Miller
Team Leader
jmiller
Apple Computer
Test Name 1
test1
Apple Computer
Test Name2
test2@generic.com
Internet
Fred Test
fred@miramontes.com
Internet
5 in list, 0 selected
Signature:
Default for account (None)
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used
Largest Unused Block: 27.1 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Macintosh
PowerBook 3400c

Exhibit E

	As another example observed during my Inspection, the email address "test1@apple.com" was added to the Now Contact "Demo Contact File," and associated with the first name "Test" and last name "One." "test1@apple.com" is associated with the "InterNet" field for that contact. The text "test1@apple.com" was then highlighted and right-clicked in the Note Pad application, which presented a menu including "test1@apple.com" and "apple.com." Hovering a mouse over the "test1@apple.com" item yielded a sub-menu including a number of actions associated with an email address. Selection of "Write a letter" launched the Claris Works application, and an "untitled (WP)" window was presented in the Claris Works application. In this example, the first name, last name, and physical address information are associated with "test1@apple.com," and is displayed in the "untitled (WP)" window. 

Exhibit E


File Edit Help
1:49 AM
About This Computer
Mac OS computer
MacOS
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 27.1 MB
LiveSimpleText 716K
Mac OS 8.5 MB
Note Pad 226K
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Help
test1@apple.com
apple.com
AppleScripts
CIS Folder
Connections
Destinations
File Transfer Tools
Locations
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Macintosh
PowerBook 3400c

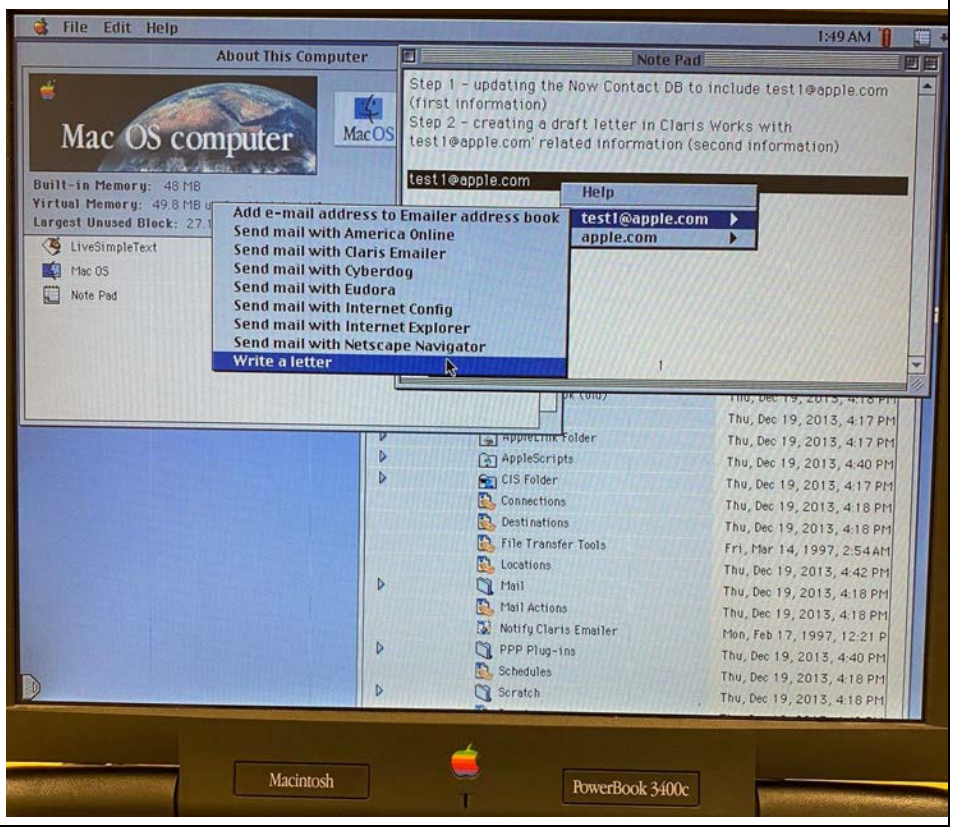
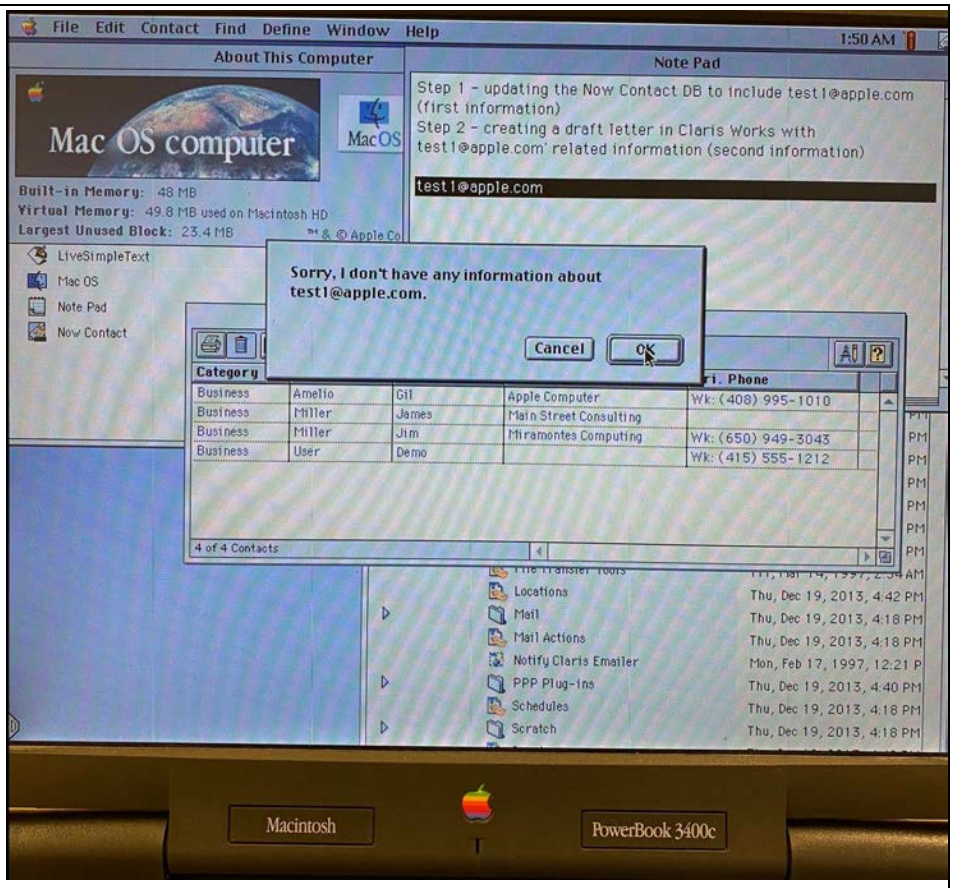
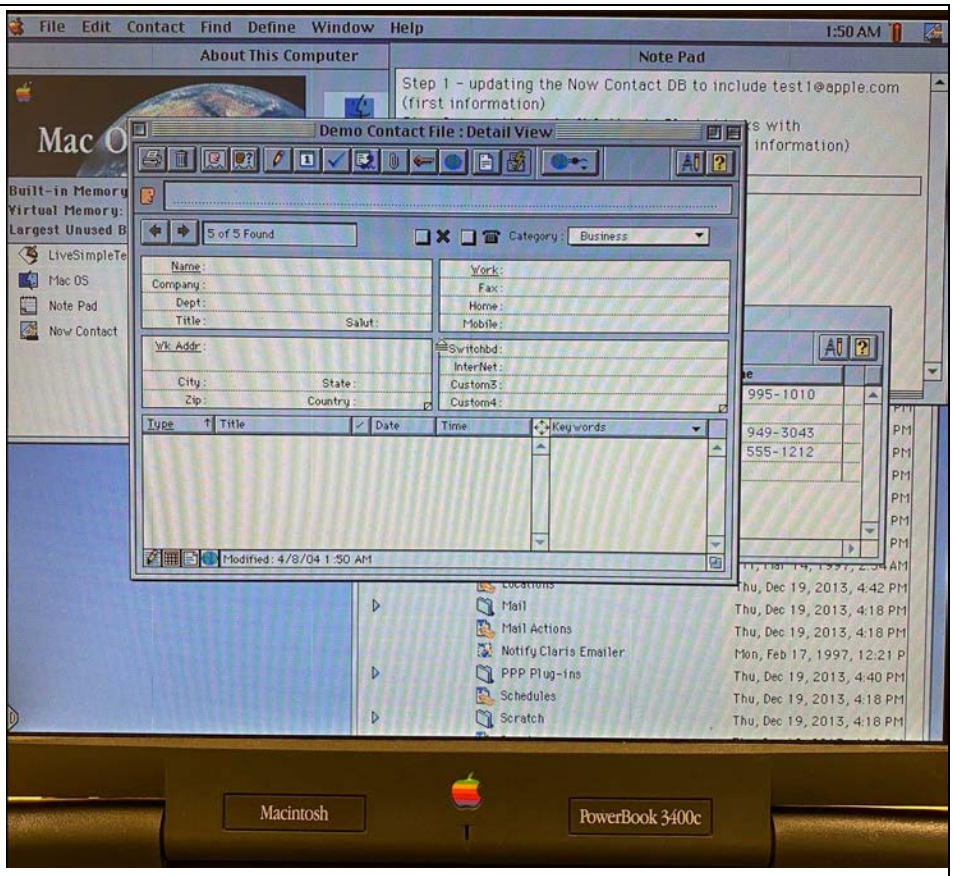

File Edit Help
1:49 AM
About This Computer
Mac OS computer
MacOS
Built-in Memory: 48 MB
Largest Unused Block:
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Help
test1@apple.com
apple.com
Add e-mail address to Emailer address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with Internet Explorer
Send mail with Netscape Navigator
Write a letter
AppleScripts
CIS Folder
Connections
Destinations
File Transfer Tools
Locations
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Macintosh
PowerBook 3400c

Exhibit E


File Edit Contact Find Define Window Help
1:50 AM
About This Computer
Mac OS computer
MacOS
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.4 MB
LiveSimpleText
Mac OS
Note Pad
New Contact
Note Pad
Step 1 - updating the New Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Sorry, I don't have any information about test1@apple.com.
Cancel
OK
Category
Business Amelio Gil Apple Computer Wk: (408) 995-1010
Business Miller James Main Street Consulting
Business Miller Jim Miramontes Computing Wk: (650) 949-3043
Business User Demo Wk: (415) 555-1212
4 of 4 Contacts
Locations Thu, Dec 19, 2013, 4:42 PM
Mail Thu, Dec 19, 2013, 4:18 PM
Mail Actions Thu, Dec 19, 2013, 4:18 PM
Notify Claris Emailer Mon, Feb 17, 1997, 12:21 P
PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM
Schedules Thu, Dec 19, 2013, 4:18 PM
Scratch Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c


File Edit Contact Find Define Window Help
1:50 AM
About This Computer
Mac OS computer
MacOS
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 27.1 MB
LiveSimpleText 716K
Mac OS 8.5 MB
Note Pad
New Contact
Note Pad
Step 1 - updating the New Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Demo Contact File
Category Last Name First Name Company Pri. Phone
Business Amelio Gil Apple Computer Wk: (408) 995-1010
Business Miller James Main Street Consulting
Business Miller Jim Miramontes Computing Wk: (650) 949-3043
Business User Demo Wk: (415) 555-1212
4 of 4 Contacts
Locations Thu, Dec 19, 2013, 4:42 PM
Mail Thu, Dec 19, 2013, 4:18 PM
Mail Actions Thu, Dec 19, 2013, 4:18 PM
Notify Claris Emailer Mon, Feb 17, 1997, 12:21 P
PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM
Schedules Thu, Dec 19, 2013, 4:18 PM
Scratch Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

Exhibit E


File Edit Contact Find Define Window Help
1:50 AM
About This Computer
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Demo Contact File : Detail View
Built-in Memory
Virtual Memory:
Largest Unused B
LiveSimpleTe
Mac OS
Note Pad
Now Contact
5 of 5 Found
Category : Business
Name :
Company :
Dept :
Title :
Salut :
Work :
Fax :
Home :
Mobile :
Wk Addr :
City :
State :
Zip :
Country :
Switchbd :
InterNet :
Custom3 :
Custom4 :
Type
Title
Date
Time
Keywords
Modified: 4/8/04 1:50 AM
995-1010
949-3043
555-1212
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Thu, Dec 19, 2013, 4:42 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Mon, Feb 17, 1997, 12:21 P
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

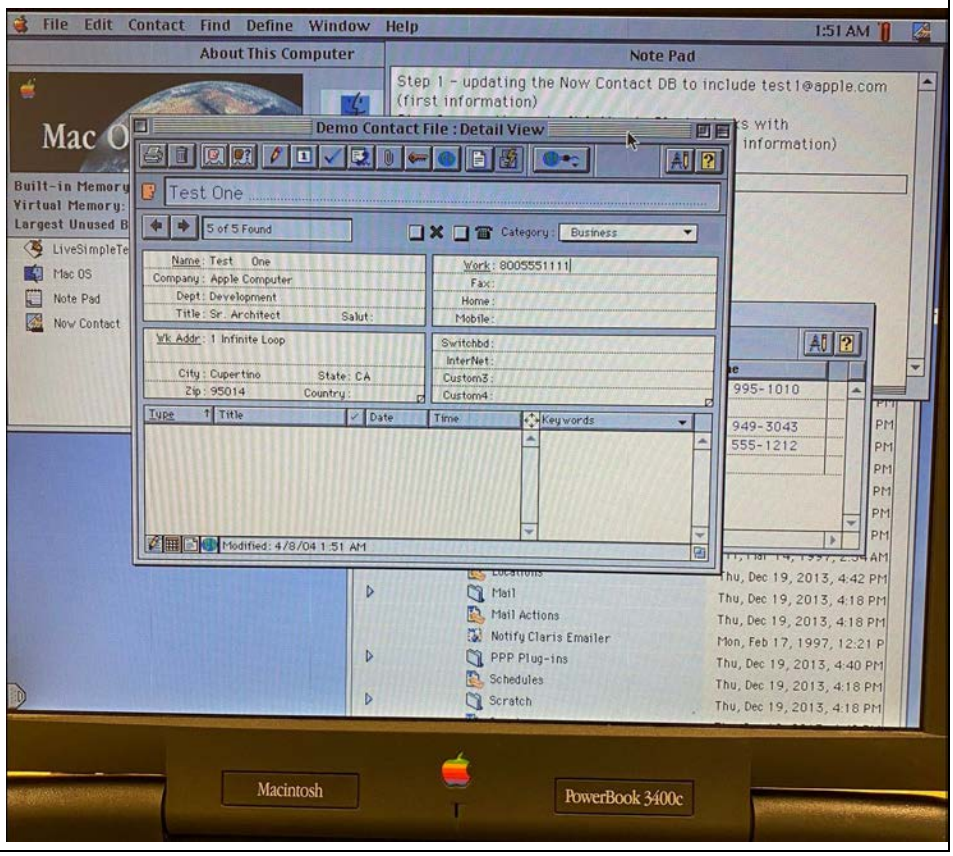
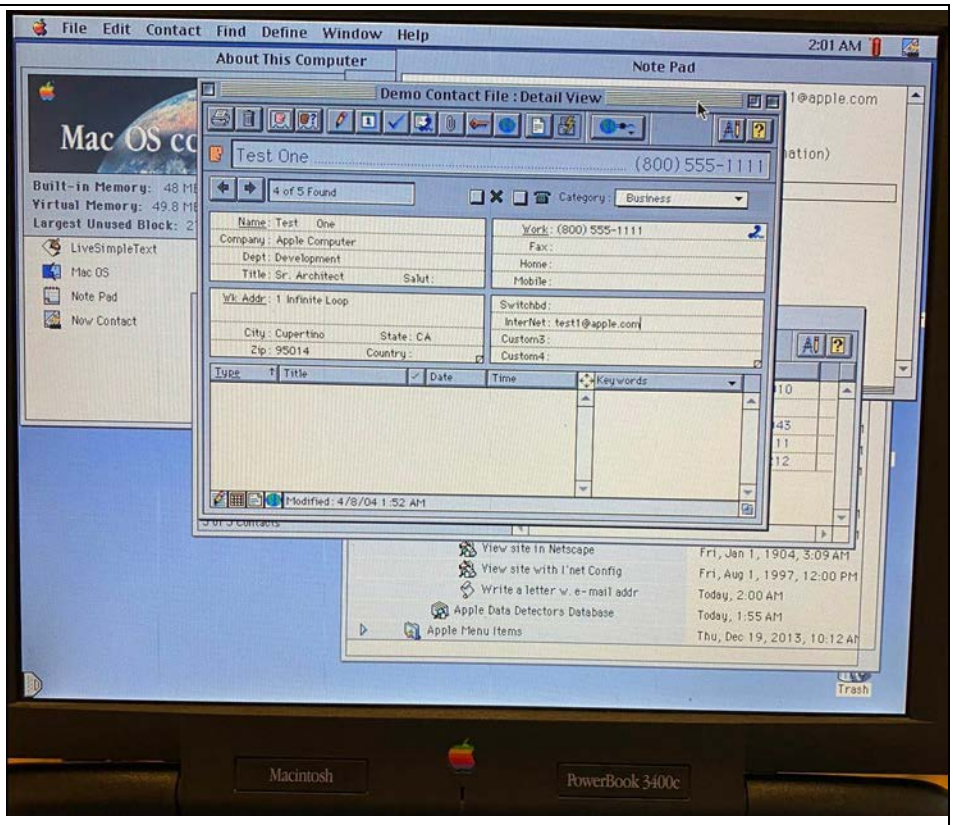

File Edit Contact Find Define Window Help
1:51 AM
About This Computer
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Demo Contact File : Detail View
Test One
Built-in Memory
Virtual Memory:
Largest Unused B
LiveSimpleTe
Mac OS
Note Pad
Now Contact
5 of 5 Found
Category : Business
Name : Test One
Company : Apple Computer
Dept : Development
Title : Sr. Architect
Salut :
Work : 8005551111
Fax :
Home :
Mobile :
Wk Addr : 1 Infinite Loop
City : Cupertino
State : CA
Zip : 95014
Country :
Switchbd :
InterNet :
Custom3 :
Custom4 :
Type
Title
Date
Time
Keywords
Modified: 4/8/04 1:51 AM
995-1010
949-3043
555-1212
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Thu, Dec 19, 2013, 4:42 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Mon, Feb 17, 1997, 12:21 P
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

Exhibit E


File Edit Contact Find Define Window Help
2:01 AM
About This Computer
Note Pad
Demo Contact File : Detail View
Test One
(800) 555-1111
Mac OS
Built-in Memory:
Virtual Memory:
Largest Unused Block:
LiveSimpleText
Mac OS
Note Pad
Now Contact
Name: Test One
Company: Apple Computer
Dept: Development
Title: Sr. Architect
Work: (800) 555-1111
Fax:
Home:
Mobile:
Wk Addr: 1 Infinite Loop
City: Cupertino
State: CA
Zip: 95014
Country:
Switchbd:
InterNet: test1@apple.com
Custom3:
Custom4:
Modified: 4/8/04 1:52 AM
View site in Netscape
View site with I'net Config
Write a letter w. e-mail addr
Apple Data Detectors Database
Apple Menu Items
Trash
Macintosh
PowerBook 3400c

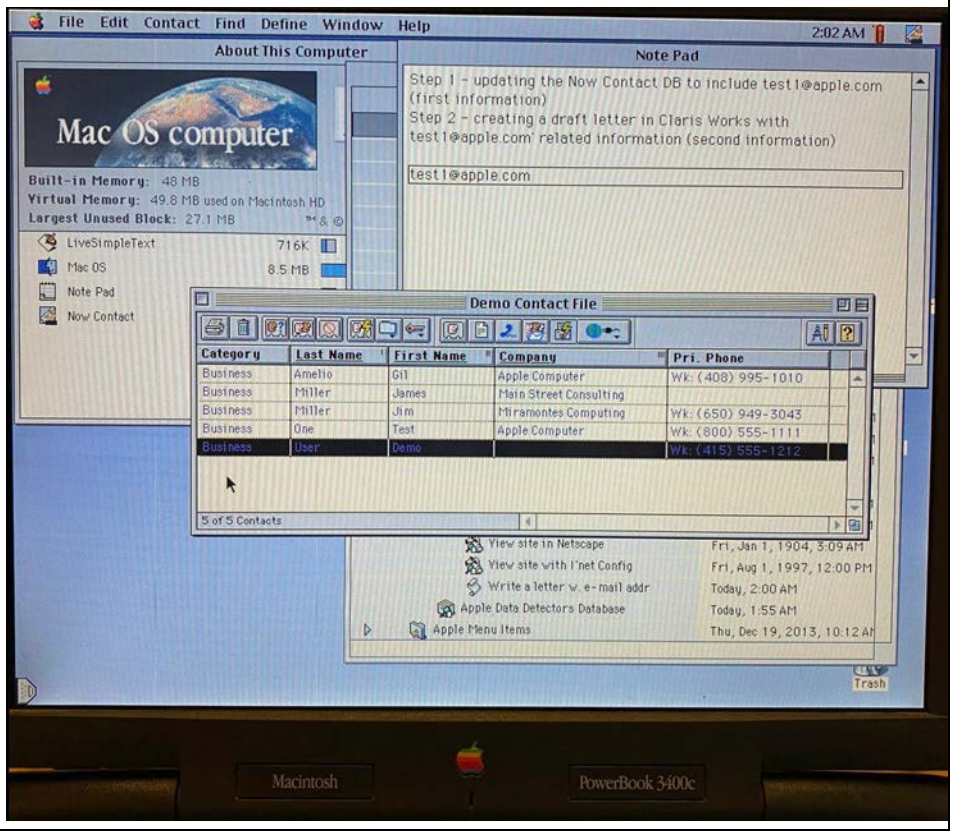
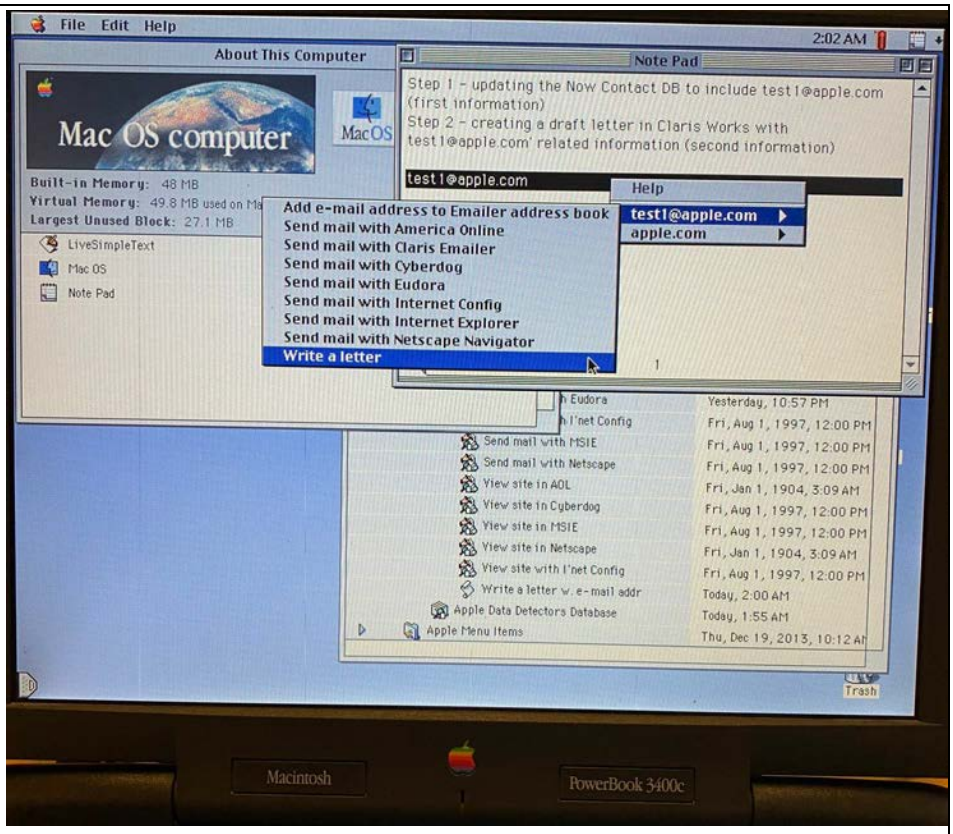

File Edit Contact Find Define Window Help
2:02 AM
About This Computer
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Mac OS computer
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 27.1 MB
LiveSimpleText 716K
Mac OS 8.5 MB
Note Pad
Now Contact
Demo Contact File
Category
Last Name
First Name
Company
Pri. Phone
Business Amelio Gil Apple Computer Wk: (408) 995-1010
Business Miller James Main Street Consulting
Business Miller Jim Miramontes Computing Wk: (650) 949-3043
Business One Test Apple Computer Wk: (800) 555-1111
Business User Demo Wk: (415) 555-1212
5 of 5 Contacts
View site in Netscape
View site with I'net Config
Write a letter w. e-mail addr
Apple Data Detectors Database
Apple Menu Items
Trash
Macintosh
PowerBook 3400c

Exhibit E


File Edit Help
2:02 AM
About This Computer
Mac OS computer
MacOS
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Ma
Largest Unused Block: 27.1 MB
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Help
Add e-mail address to Emailer address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with Internet Explorer
Send mail with Netscape Navigator
Write a letter
test1@apple.com
apple.com
Yesterday, 10:57 PM
Fri, Aug 1, 1997, 12:00 PM
Fri, Aug 1, 1997, 12:00 PM
Fri, Aug 1, 1997, 12:00 PM
Fri, Jan 1, 1904, 3:09 AM
Fri, Aug 1, 1997, 12:00 PM
Fri, Aug 1, 1997, 12:00 PM
Fri, Jan 1, 1904, 3:09 AM
Fri, Aug 1, 1997, 12:00 PM
Today, 2:00 AM
Today, 1:55 AM
Thu, Dec 19, 2013, 10:12 AM
Send mail with MSIE
Send mail with Netscape
View site in AOL
View site in Cyberdog
View site in MSIE
View site in Netscape
View site with I'net Config
Write a letter w. e-mail addr
Apple Data Detectors Database
Apple Menu Items
Trash
Macintosh
PowerBook 3400c

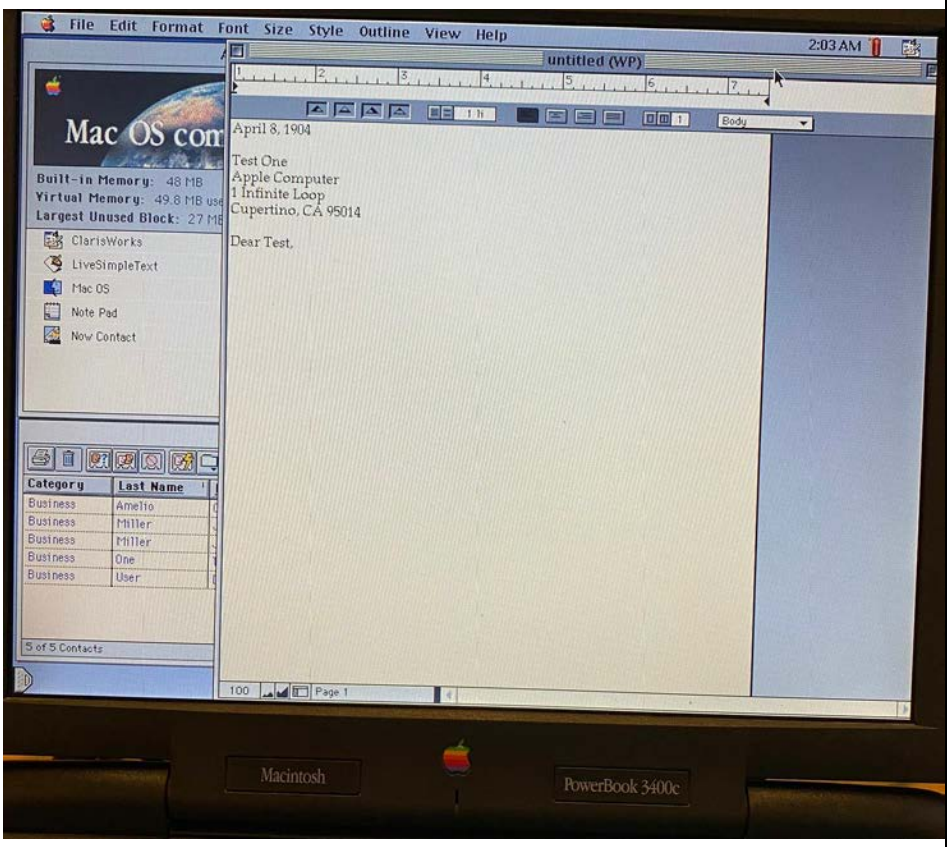

File Edit Format Font Size Style Outline View Help
2:03 AM
untitled (WP)
Mac OS com
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB use
Largest Unused Block: 27 MB
ClarisWorks
LiveSimpleText
Mac OS
Note Pad
Now Contact
April 8, 1904
Test One
Apple Computer
1 Infinite Loop
Cupertino, CA 95014
Dear Test,
Category
Last Name
Business Amelio
Business Miller
Business Miller
Business One
Business User
5 of 5 Contacts
Page 1
Macintosh
PowerBook 3400c

Exhibit E

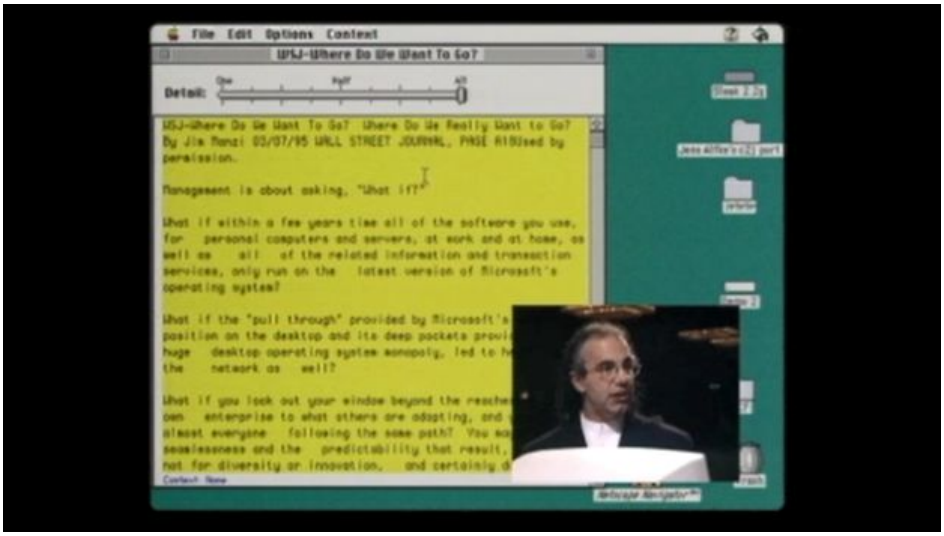
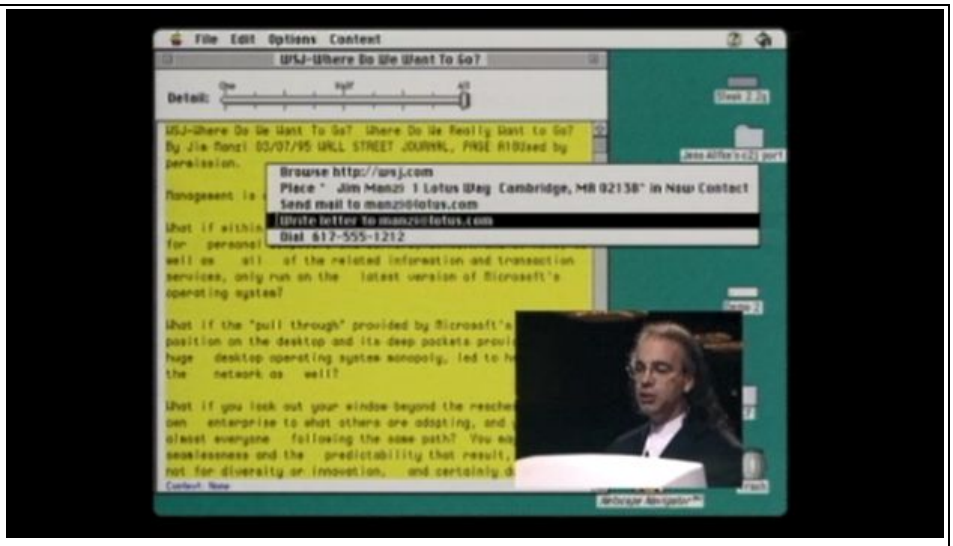
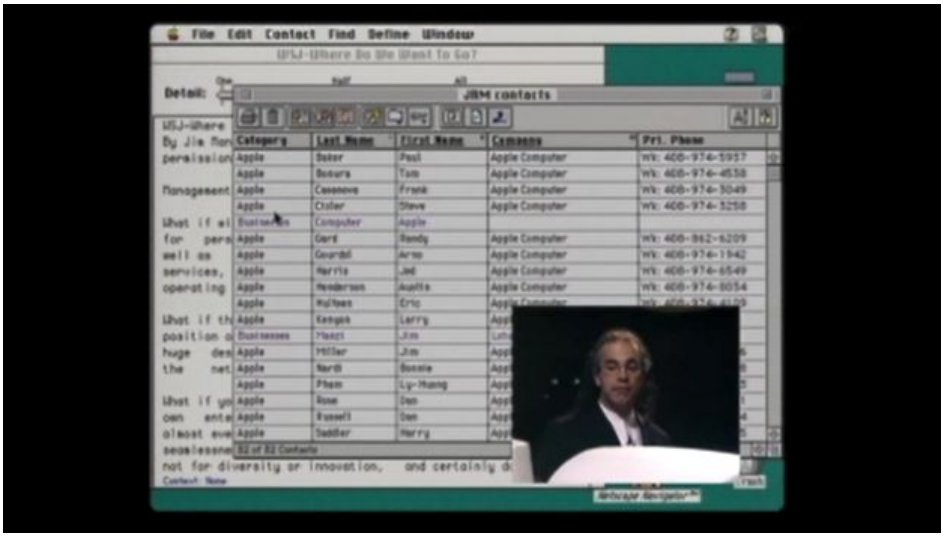
	See also next element. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	The Apple Data Detectors System discloses this element. *See* previous element. *See also* Nardi at Fig. 4: "**on** addressLetterTo(phoneNumber) --Open Now Contact and find a person with the indicated phone number "**tell** application "Now Contact 3.5" --find the person with the supplied e-mail address **set** thePerson **to the first** person **whose** (work phone **is** phoneNumber) --get the address information for this person **set** firstAndLastName **to** (**the** first name of thePerson) & " " & (**the** last name of thePerson) **set** theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (**the** work address of thePerson) & return & (**the** work city **of** thePerson) & "," & (**the** work state of **the** Person) & " " & (the work zip of thePerson) & return & return **end tell"** *See also* MacWorld 1997 Expo at 1:11:00: 

Exhibit E


File Edit Options Content
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212


File Edit Contact Find Define Window
WSJ-Where Do We Want To Go?
Detail:
JBM contacts
Category
Last Name
First Name
Company
Ph1 Phone

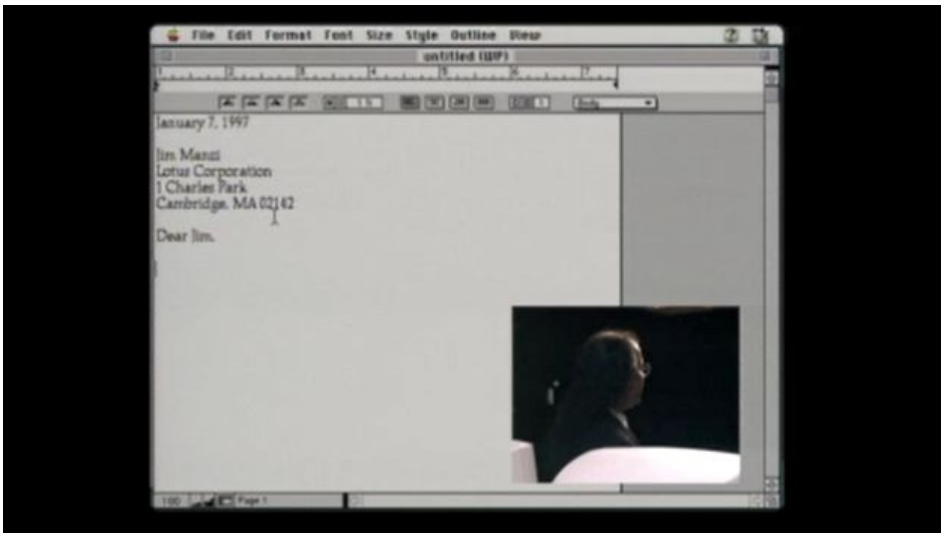

File Edit Format Font Size Style Outline View
untitled (WP)
January 7, 1997
Jim Manzi
Lotus Corporation
1 Charles Park
Cambridge, MA 02142
Dear Jim,

Exhibit E

	See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information." *See also* Nardi at fig. 4: "This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address and salutation into it, leaving the user ready to write the letter." *See also* Nardi at 99: "While Apple and third-party developers will provide many detectors and actions, it is clear that enabling end users to write their own detectors and actions will make Apple Data Detectors much more powerful and useful by providing domain-specific programming capabilities appropriate for the specific needs of specific users [8]." *See also* Nardi at 101: "Unlike systems based on predefined

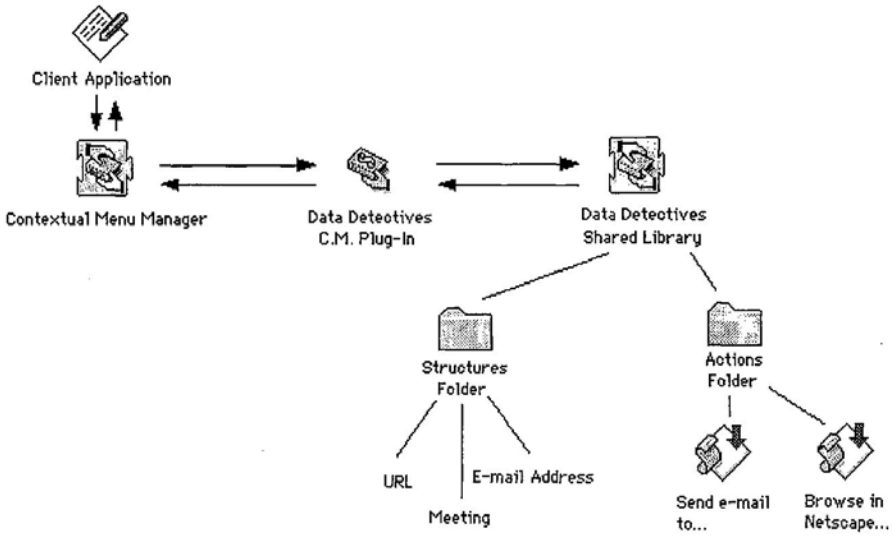
Exhibit E

	recognizer/action pairs, such as the Selection Recognition Agent [10], the Apple Data Detectors' scripting ability allows any set of arbitrary actions to be executed when the user selects a particular action (see Figure 3). Scripting is not limited to the parameters of a command line interface to the application; it can do anything that can be expressed in the scripting language, including manipulation of data structures inside the application, if the application's scripting model makes that possible." *See also* Nardi at 101: "The ability to work within existing documents provides immediate user value and leverages the data that the user is already using." *See also* Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." See also Nardi at 102: "We also see evidence that end users with varying degrees of programming skill are extending actions much as we had expected. One user—a skilled programmer—used the basic detectors and actions that ship with the product as the basis for a new detector/action pair for a personally relevant task. He wanted to look up software bug reports in a database based on the ID numbers of the reports commonly found in other bug reports, email messages, and other program management documents. He distributed this detector/action throughout his work group, and a colleague—a marketing manager with much less programming experience than the original programmer—was able to adapt the action so the detector could run on his laptop computer (where his email and other documents reside) and display the bug reports on his desktop machine (where is program management tools reside). This extension required changes to only a few lines of code in the action, something well within his technical ability." The system could detect an email address, and in response to the "Write letter" command, obtain the mailing address for that person by searching an address book using Now Contact, and address a letter to that person and mailing address in Claris Works. *See* screenshot above, Video at 1:22-1:40. "Find inherently structured data, let user take action on data" *See* WWDC Presentation at 2.

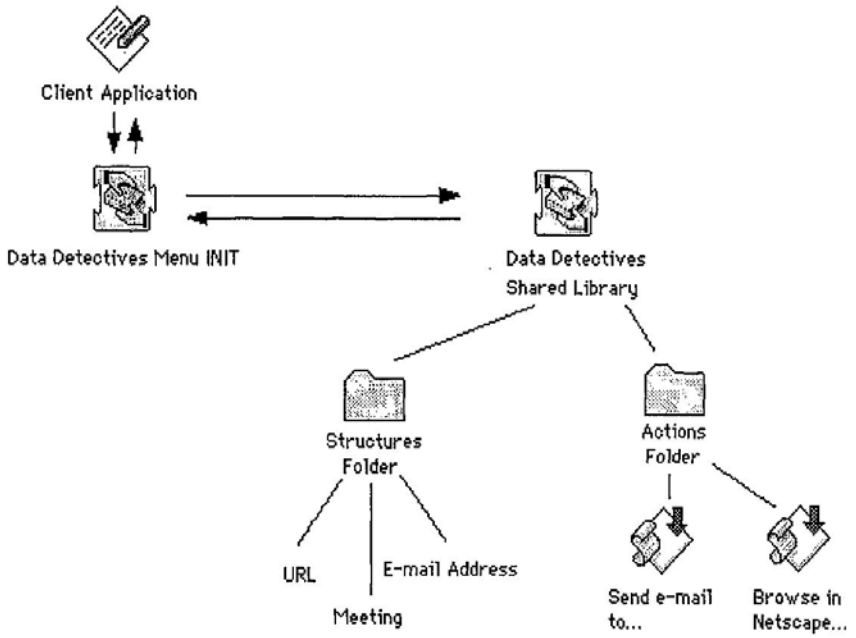
Exhibit E

See WWDC Presentation at 4-5:

[Technical Overview - with Contextual Menus in Copland:]



[Technical Overview - Current]



"**on** addressLetterTo(emailAddress)
set emailAddress **to** removeTextStyling(emailAddress)
tell application "Now Contact 3.5"
launch
activate
run

Exhibit E

	--find the person in the Now Contact database **try** **set** thePerson **to the first** person **whose** (customer 2 is emailAddress) **on error** beep display dialog “Sorry, I don’t have any information about “ & emailAddress & “.” **return** **end try** *--get the address information for the person* **try** **set** firstAndLastName **to** (**the** first name **of** thePerson) & “ ” & (**the** last name **of** thePerson) **set** theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (the work address **of** thePerson) & return & (**the** work city **of** thePerson) & “, ” & (**the** work state **of** thePerson) & “ ” & (**the** work zip of thePerson) & return **on error** beep display dialog “Sorry, I couldn’t get the address of “ & emailAddress 7 “.” **return** **end try**” *See* Write a Letter AppleScript Code. "Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." Miller at 2:42-46. "Window 510 includes a button 520 for initiating program 165 . . . . Upon initiation of program 165, system 100 transmits the contents of document 210 to analyzer server 220." Miller at 5:22-26. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 *See also, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "); 5:6-37; 3:52-4:10 ("After identifying structures and

Exhibit E

	linking actions, application program interface 230 [of program 165] communicates with application 167 to obtain information on the identified structure so that user interface 240 can successfully present and enable selection of the actions"). Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98). "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send

Exhibit E

	that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4 "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned?

Exhibit E

	A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted.

Exhibit E

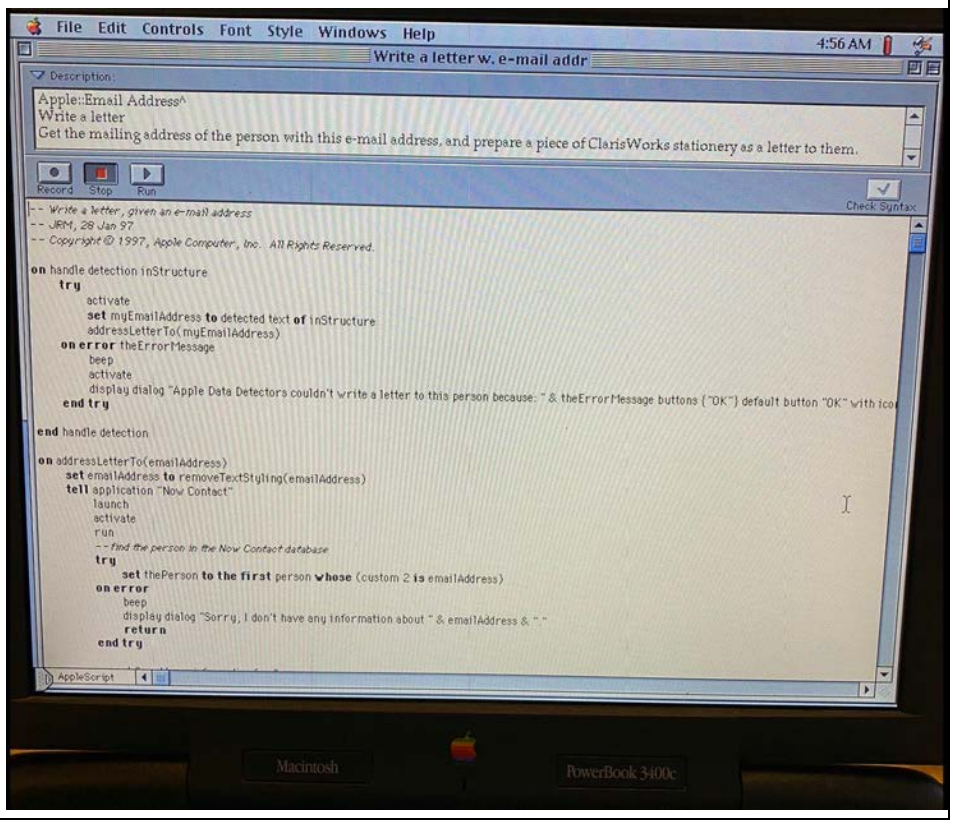
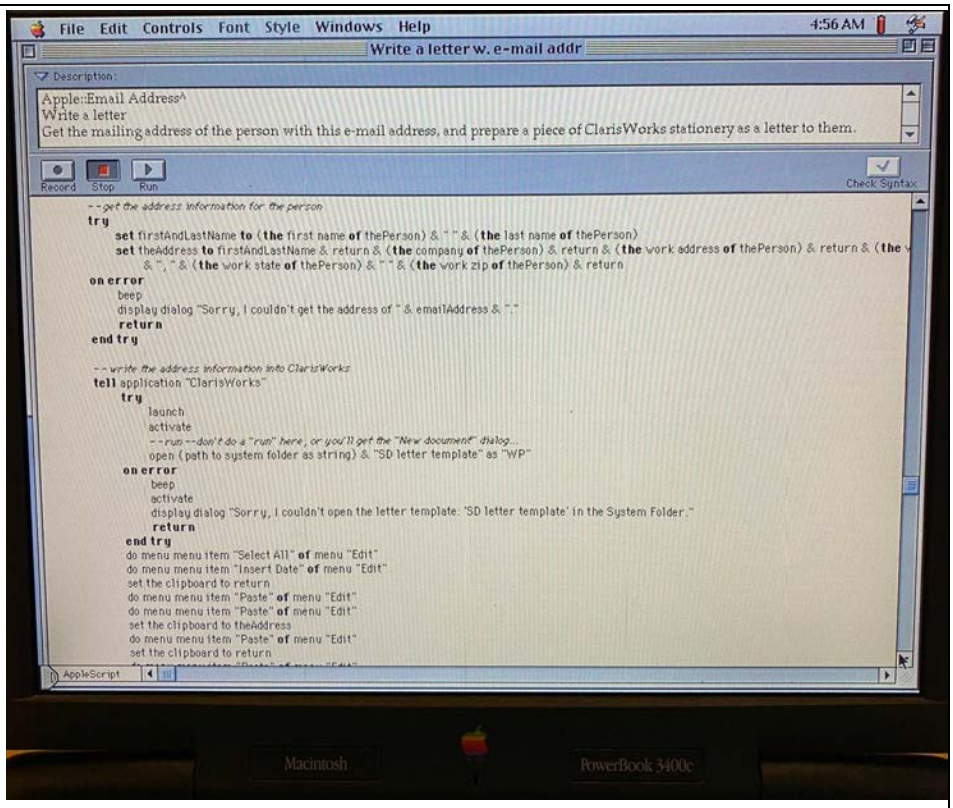
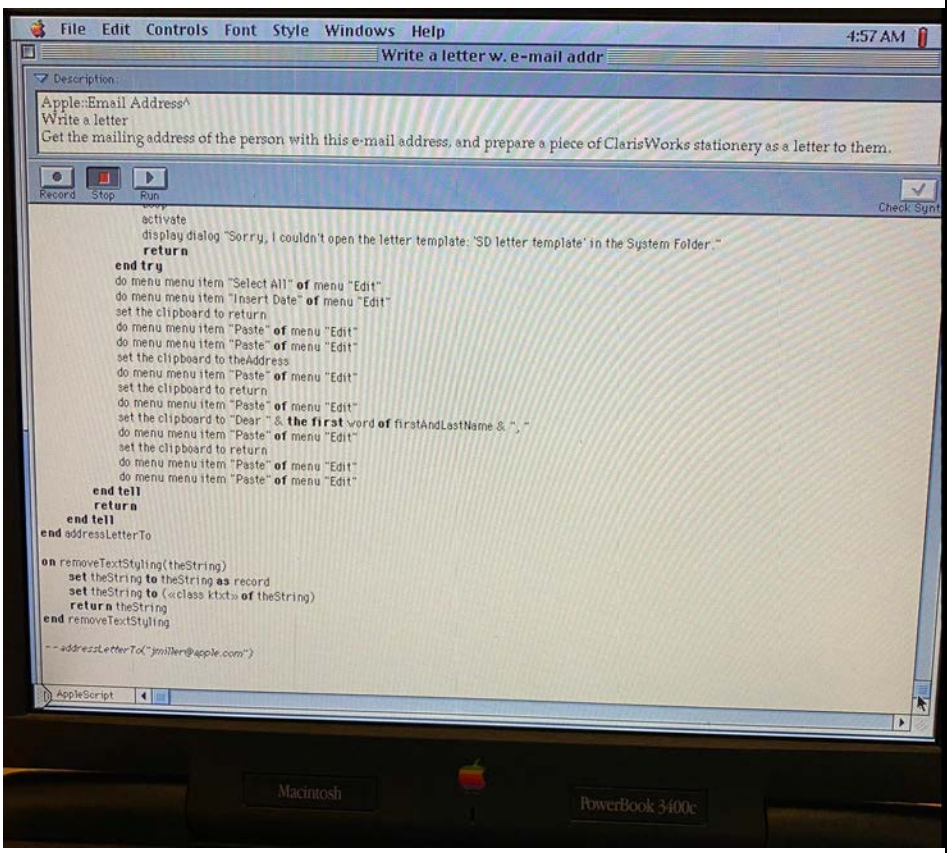
	You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. For example, during my Inspection of the Powerbook 3400C, I located and observed the written computer program code associated with the "Write a letter" action that was described in the previous element. 

Exhibit E





For example (and without limitation to the Obviousness Statement that is

Exhibit E

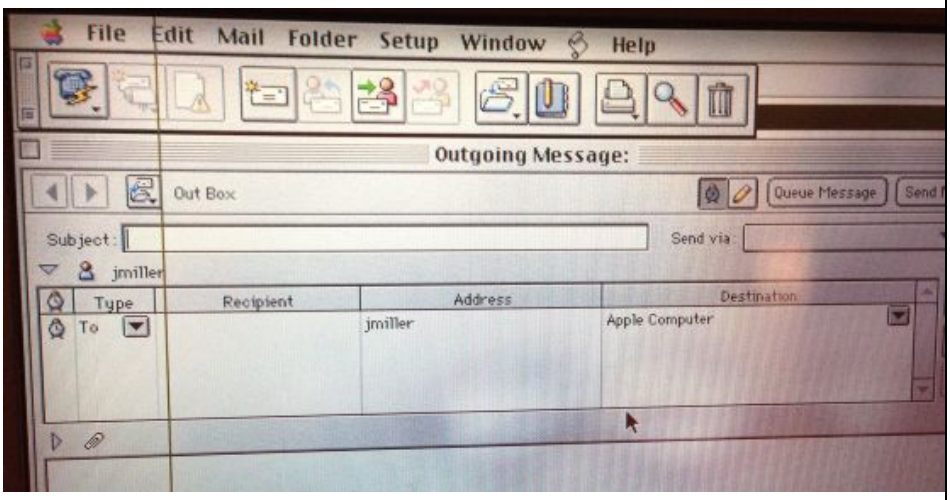
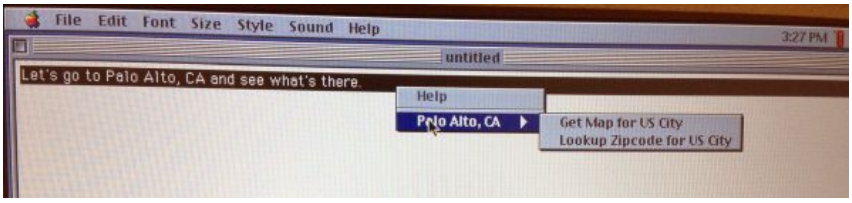
	incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	The Apple Data Detectors System discloses this element. When IAD detects an email address and a user selects "Send mail with Claris Emailer," Claris Emailer will search its information source for a company name associated with the domain of the email address and will insert this into the Outgoing Message panel for a new email. For example:  When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could obtain a zip code for the selected US city, for which the system would search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:  This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

Exhibit E

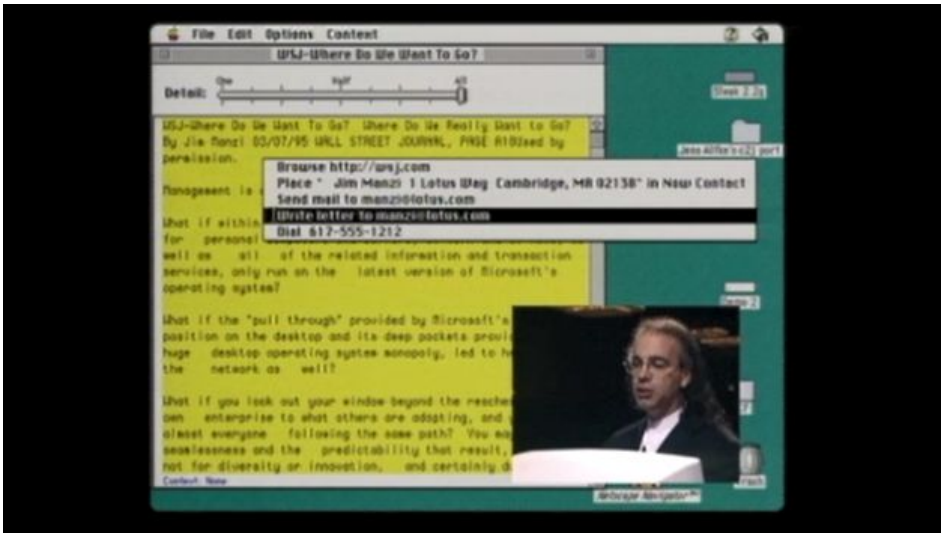
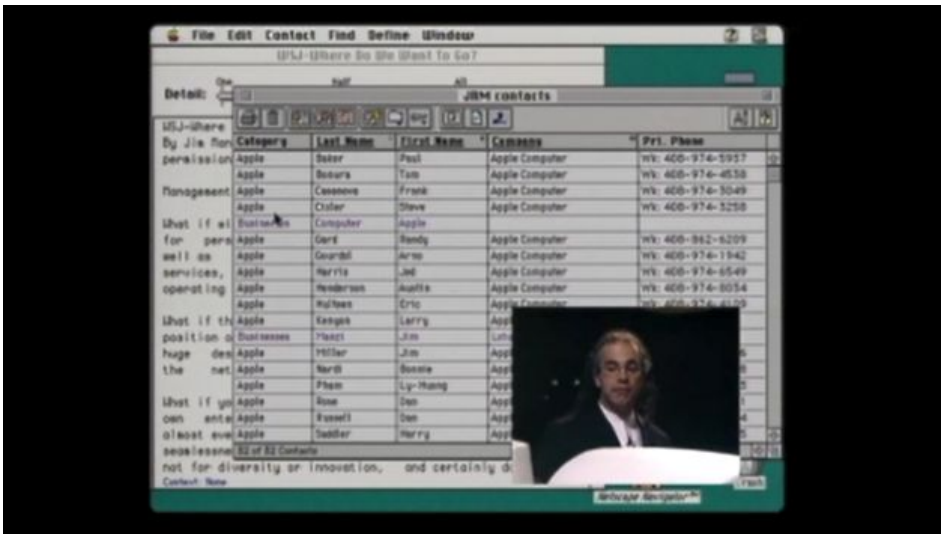
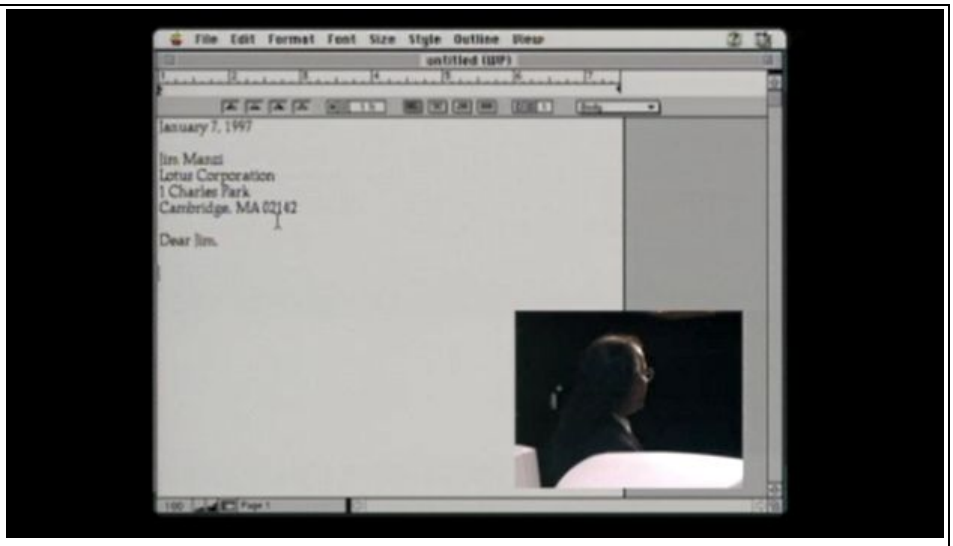
	See also MacWorld 1997 Expo at 1:11:00:  

Exhibit E



See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts.

Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information."

See also Nardi at fig. 4: "This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date

Exhibit E

	and salutation information. This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address and salutation into it, leaving the user ready to write the letter." See also Nardi at 99: "While Apple and third-party developers will provide many detectors and actions, it is clear that enabling end users to write their own detectors and actions will make Apple Data Detectors much more powerful and useful by providing domain-specific programming capabilities appropriate for the specific needs of specific users [8]." See also Nardi at 101: "Unlike systems based on predefined recognizer/action pairs, such as the Selection Recognition Agent [10], the Apple Data Detectors' scripting ability allows any set of arbitrary actions to be executed when the user selects a particular action (see Figure 3). Scripting is not limited to the parameters of a command line interface to the application; it can do anything that can be expressed in the scripting language, including manipulation of data structures inside the application, if the application's scripting model makes that possible." See also Nardi at 101: "The ability to work within existing documents provides immediate user value and leverages the data that the user is already using." See also Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." See also Nardi at 102: "We also see evidence that end users with varying degrees of programming skill are extending actions much as we had expected. One user—a skilled programmer—used the basic detectors and actions that ship with the product as the basis for a new detector/action pair for a personally relevant task. He wanted to look up software bug reports in a database based on the ID numbers of the reports commonly found in other bug reports, email messages, and other program management documents. He distributed this detector/action throughout his work group, and a colleague—a marketing manager with much less programming experience than the original programmer—was

Exhibit E

	able to adapt the action so the detector could run on his laptop computer (where his email and other documents reside) and display the bug reports on his desktop machine (where is program management tools reside). This extension required changes to only a few lines of code in the action, something well within his technical ability." See also Nardi at fig. 1: "Two patterns are found: an email address . . . and the announcement of the meeting . . . . These patterns are presented in the pop-up menu; by pointing at the date information in the menu; a second pop-up menu offers a choice of actions . . . . The user can select one, thereby running a small application, or move the cursor off the menu, eliminating the pop-up menu and canceling any actions." See also Nardi at 98: "User interface. To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1)." See also Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." See also Nardi at 103: "Our early contact with the product group again served us well. During the system's development, we experimented with various user interfaces to its basic technology, an effort that paid off in a number of ways. The different interfaces followed different assumptions about various aspects of the system, such as the interface techniques themselves, the demand the techniques would place on the underlying operating system, and the demand imposed on developers who wanted to adapt Apple Data Detectors to their own uses. It was important for us to understand such trade-offs, so we could advise the product group on the technology's possibilities."

Exhibit E

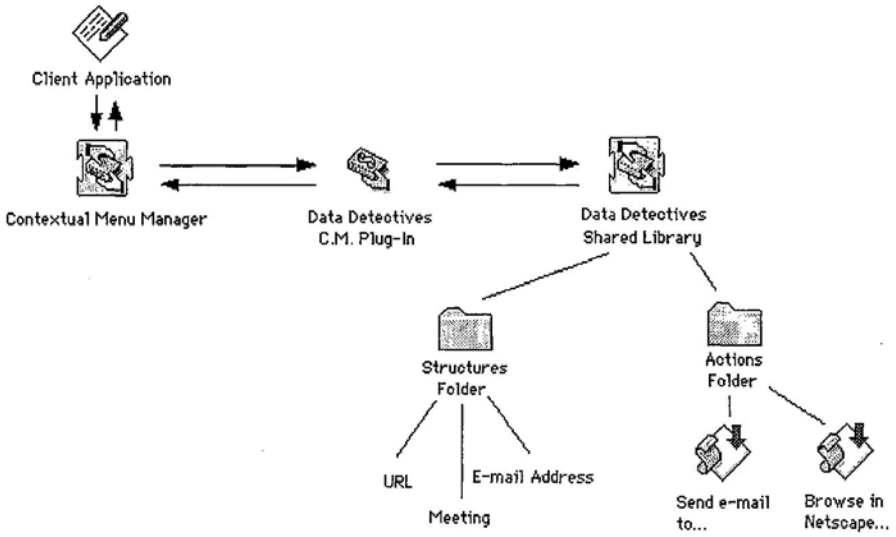
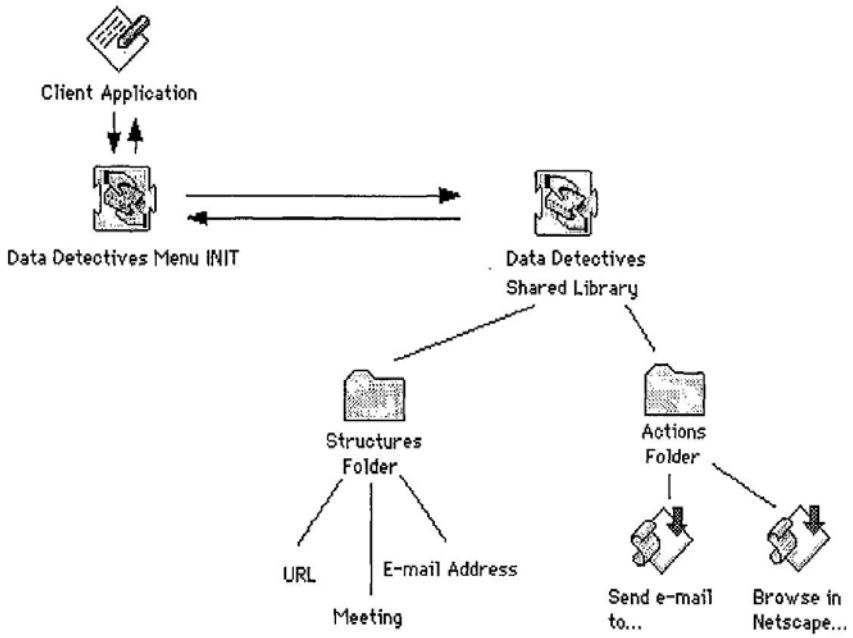
	See also Nardi at p. 99 ("Figure 4. An action script, demonstrating the generality of Apple Data Detectors' use of a scripting language and external applications as information repositories and as end-user tools. This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a "personal information manager" (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter."). The system could detect an email address, and in response to the "Write letter" command, obtain the mailing address for that person from Now Contact, and address a letter to that person and mailing address in Claris Works. See screenshot above, Video at 1:22-1:40. "Find inherently structured data, let user take action on data" *See* WWDC Presentation at 2. *See* WWDC Presentation at 4-5: [Technical Overview - with Contextual Menus in Copland:]  [Technical Overview - Current]

Exhibit E



“ *--write the address information into ClarisWorks*
tellapplication “ClarisWorks”
try
launch
activate
--run --don’t do a “run here, or you’ll get the “New document” dialog…
open (path to system folder as string) & “SD letter template” as “WP”
on error
beep
activate
display dialog “Sorry, I couldn’t open the letter template: ‘SD letter template’ in the System Folder.”
See Write a Letter AppleScript Code.

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to

Exhibit E

	the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100). "Q. Can you give me a general description of what it is that Apple Data Detectors did? A. The idea at the time was to identify bits of information in user documents. For instance, phone numbers or email addresses or, you know, other sorts of things that could be easily identified and make it very easy for people to carry out actions on them." Miller Depo. at 27:2-9. "And so we were looking for ways to find those things in user documents and then make it easy for people to carry out actions on them, to do things with those, let's say, email addresses that were found." Miller Depo. at 39:23-40:2. "The phrase 'your application' refers to the applications that the attendees of WWDC were building. And the point that we wanted to make was that what we were building could provide a way for users to be able to very easily interact with those applications without having to go through many of the details of pulling down menus off of the menu bar and clicking on things and repositioning the cursor and typing stuff into text fields and clicking

Exhibit E

	submit buttons and all of that. They could just interact directly with the content and then let the software go through all of the tedious parts of actually carrying out actions on that information. Q. And when you say 'input data more easily,' do you recall what that was referring to in your presentation? A. Well, that if we could identify, say, a very long email address and provide a way for someone to act directly on it, then the user would not have to open up a text box and remember and type in that email address, possibly making mistakes along the way. Q. So what sorts of things could be done with the email address like that using the Apple Data Detectives that you're describing in this presentation? A. Well, in particular you could create a new outgoing email message to that email address. You could also put it into an address book so that you could easily use it later. Q. How would that work? Using the Apple Data Detectors or Detectives as they are called in this presentation, how would you actually take an email address in a document and put it into an address book? A. There were application programmatic interfaces, APIs, that would receive requests from outside parts of the computer system. And so the way that this would ordinarily work is that our part of the system would identify this and then send a message through this API channel over to the application and say, 'Make a new email message addressed to this person.' Q. How about to save it in their address book, how would that work? A. There would be another kind of message that could be sent to the application. This one would be 'save this email address inside of your address book.' Q And did you actually build that kind of functionality into the Apple Data Detectors? A. Yes. We would work with applications that could accept those messages and then we would write the bits of code that would take the discovered pieces of information and send them to the application." Miller Depo. at 42:23-45:7. "Then the Data Detectives CM plug-in then exchanges information, it appears, with something called the Data Detectives shared library. Do you see that? A. Yes. Q. Then within that there are two folders, one called "structures folder" and one called "actions folder." Is that right? A. Yes. Q. What is meant by "structures folder"? A. These were the patterns that would recognize different kinds of

Exhibit E

	information. So there would be a component that was able to recognize URLs, http:// followed by a bunch of stuff, or an email address. Q. And I notice in the structures folder, in addition to URLs, it also indicates email address; is that right? A. Yes. Q. And is that something then that you had disclosed that the Apple Data Detectives would be able to recognize as a structure? A. Yes. Q. And I also notice meetings is included in the structure folder. Was that another structure that you disclosed that Apple Data Detectives would be able to identify? A. Yes. Q. Then there's the separate folder called "actions folder." What is that intended to indicate? A. This contains these small bits of code that are able to talk to the applications to carry out the actions. And that after a user had -- or after we had identified a piece of information like an email address, and the user had then indicated I would like to send an email message to that -- to that person, that little application icon would send email to, that would run accept the email address and tell its targeted application to create a new email message." Miller Depo. at 46:23-48:15. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes.

Exhibit E

	Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4 "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "It was possible to add an address to some contact managers' address book. I believe it was Now Contact." Miller Depo. at 84:18-20. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes.

Exhibit E

	Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information?

Exhibit E

	A. Yes." Miller Depo. at 126:14-128:14. "Q. And so looking back at the screenshot that is Number 2 in this Miller Exhibit 14, am I correct that when Lotus – I'm sorry, when Apple Data Detectors are invoked on this document, the user is given, for this particular document, five different options of things that he or show can do? A. Yes. * * * Q. And the user can click any of these with a single click of the mouse? A. Yes." Miller Depo. at 129:17-130:18. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect,

Exhibit E

	which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "Q. When you say 'pointing at a highlight and pressing the mouse

Exhibit E

	button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the

Exhibit E

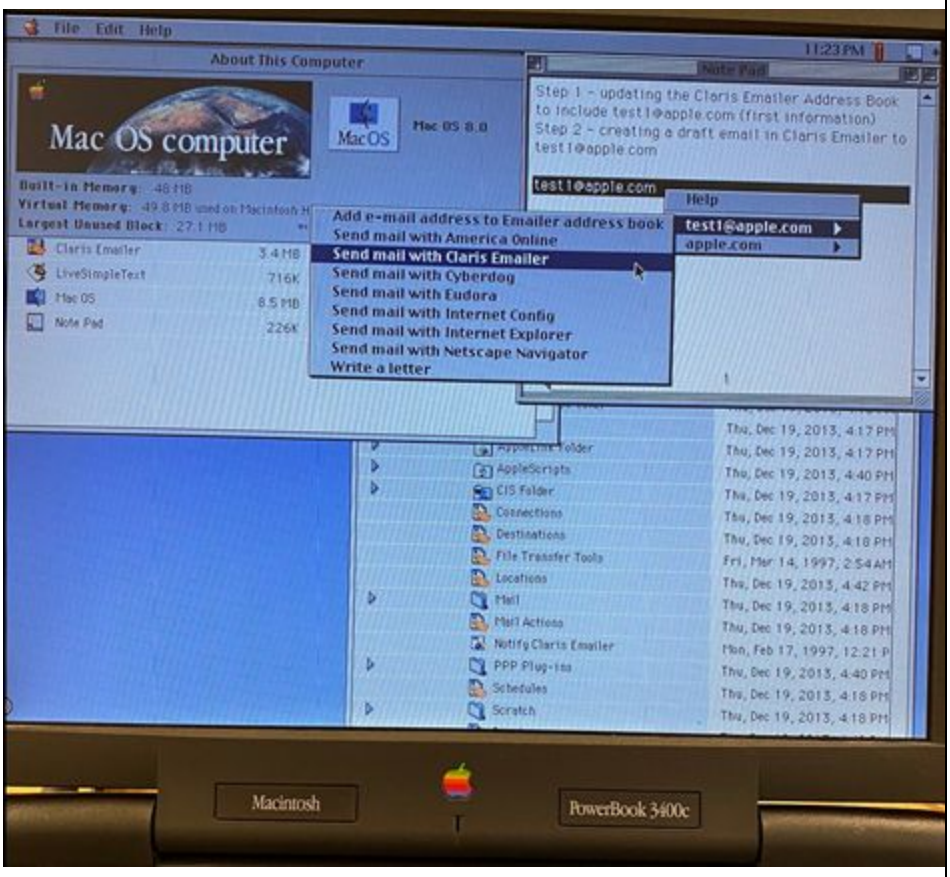
	mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Send mail with Claris Emailer" from the sub-menu, the Claris Emailer application is launched, and an "Outgoing Message" window is presented in the Claris Emailer application. In this example, "Apple Computer" is associated with "test1@apple.com" in the Claris Emailer Address Book, and is displayed in the "Outgoing Message" window in the "Destination" field. See also additional screenshots and text associated with this example or with the example "test2@generic.com", in the ("providing an input device, configured by the first computer program . . . ") element. 

Exhibit E

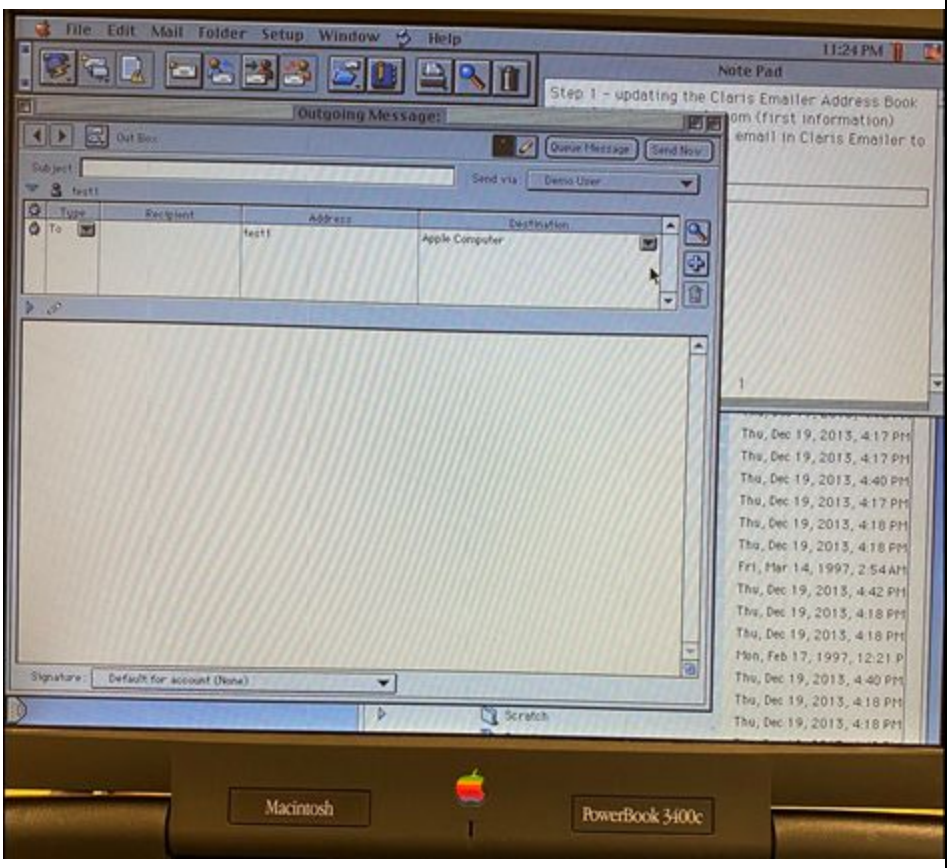
	 File Edit Mail Folder Setup Window Help 11:24 PM Note Pad Step 1 - updating the Claris Emailer Address Book email in Claris Emailer to Outgoing Message: Queue Message Send Now Send via: Demo User Type Recipient Address Destination To test1 Apple Computer Signature: Default for account (None) Scratch Macintosh PowerBook 3400c

Exhibit E

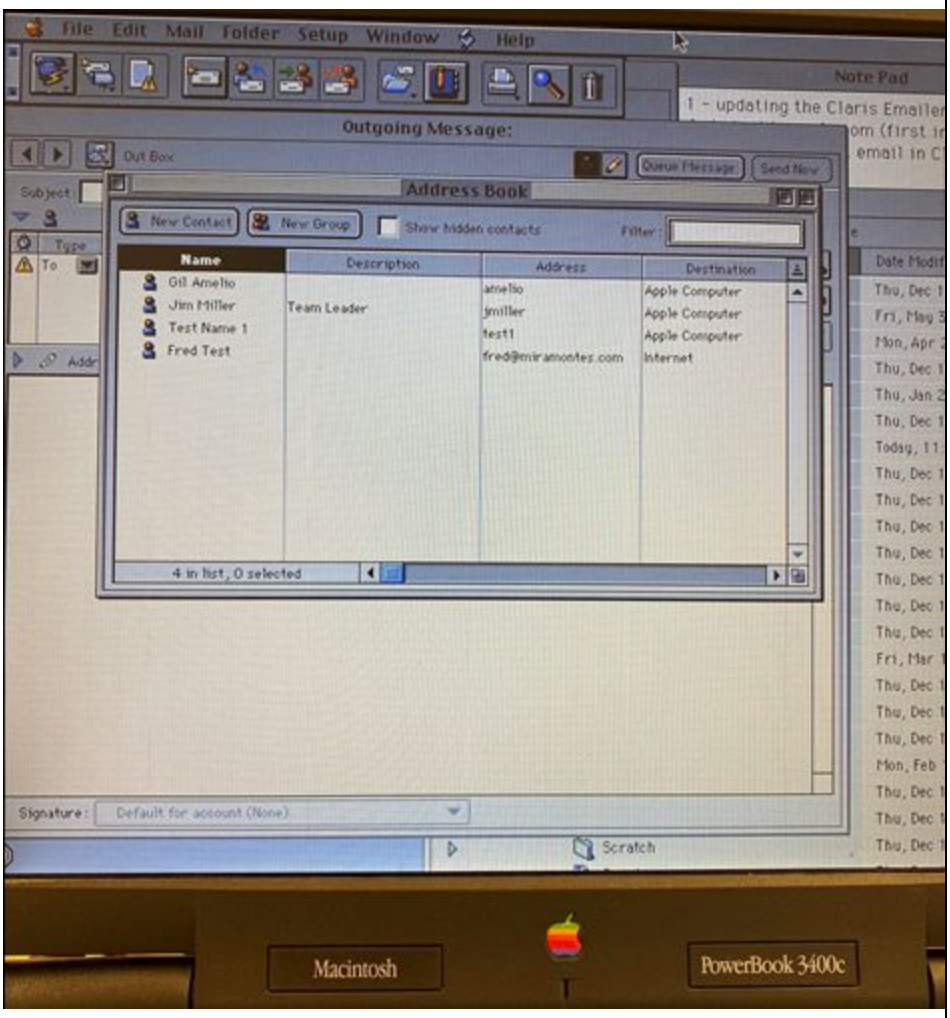
	 For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Write a letter" from the sub-menu, the Now Contact and Claris Works applications are launched, and an "untitled (WP)" window is presented in the Claris Emailer application. In this example, the first name ("Test"), last name ("One"), and physical address information ("Apple Computer 1 Infinite Loop Cupertino, CA 95014") are associated with "test1@apple.com" in the Now Contact Demo Contact File, and that is displayed in the "untitled (WP)" window. See also additional screenshots and text associated with this example in the ("providing an input device, configured by the first computer program . . . ") element.

Exhibit E

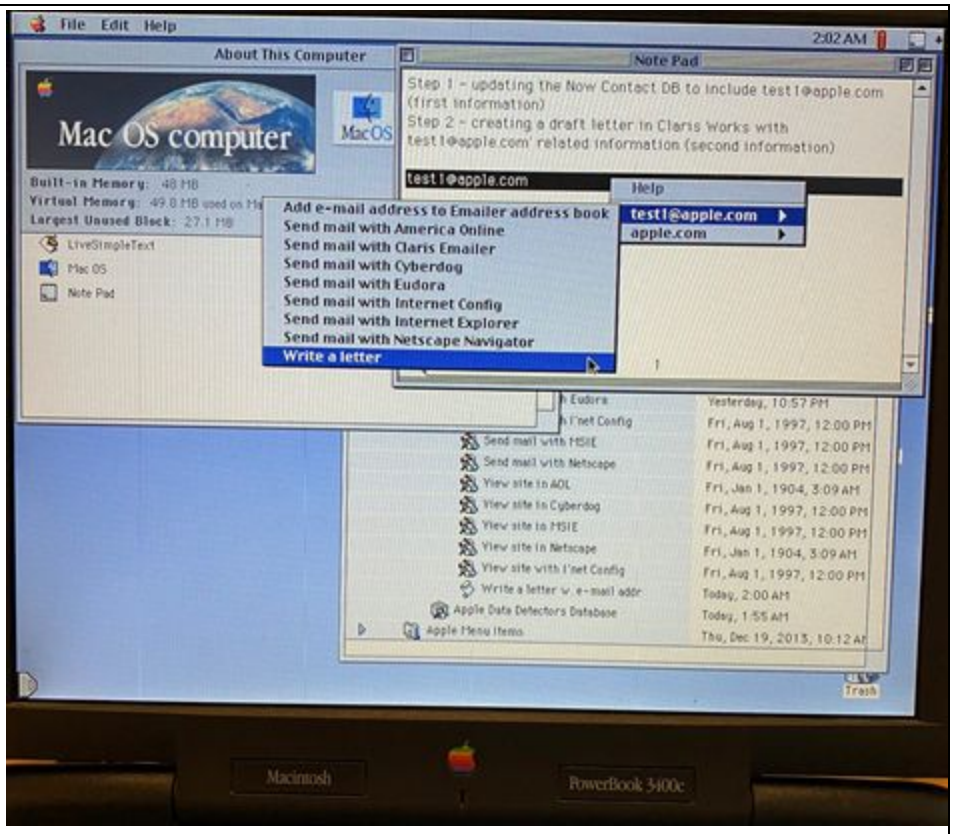
	 File Edit Help 2:02 AM About This Computer Mac OS computer MacOS Note Pad Step 1 - updating the Now Contact DB to include test1@apple.com (first information) Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information) test1@apple.com Help test1@apple.com apple.com Add e-mail address to Emailer address book Send mail with America Online Send mail with Claris Emailer Send mail with Cyberdog Send mail with Eudora Send mail with Internet Config Send mail with Internet Explorer Send mail with Netscape Navigator Write a letter Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Ha Largest Unused Block: 27.1 MB LiveSimpleText Mac OS Note Pad Send mail with MSIE Send mail with Netscape View site in AOL View site in Cyberdog View site in MSIE View site in Netscape View site with i'net Config Write a letter w. e-mail addr. Apple Data Detectors Database Apple Menu Items Yesterday, 10:57 PM Fri, Aug 1, 1997, 12:00 PM Fri, Aug 1, 1997, 12:00 PM Fri, Aug 1, 1997, 12:00 PM Fri, Jan 1, 1904, 3:09 AM Fri, Aug 1, 1997, 12:00 PM Fri, Aug 1, 1997, 12:00 PM Fri, Jan 1, 1904, 3:09 AM Fri, Aug 1, 1997, 12:00 PM Today, 2:00 AM Today, 1:55 AM Thu, Dec 19, 2013, 10:12 AM Trash Macintosh PowerBook 3400c

Exhibit E

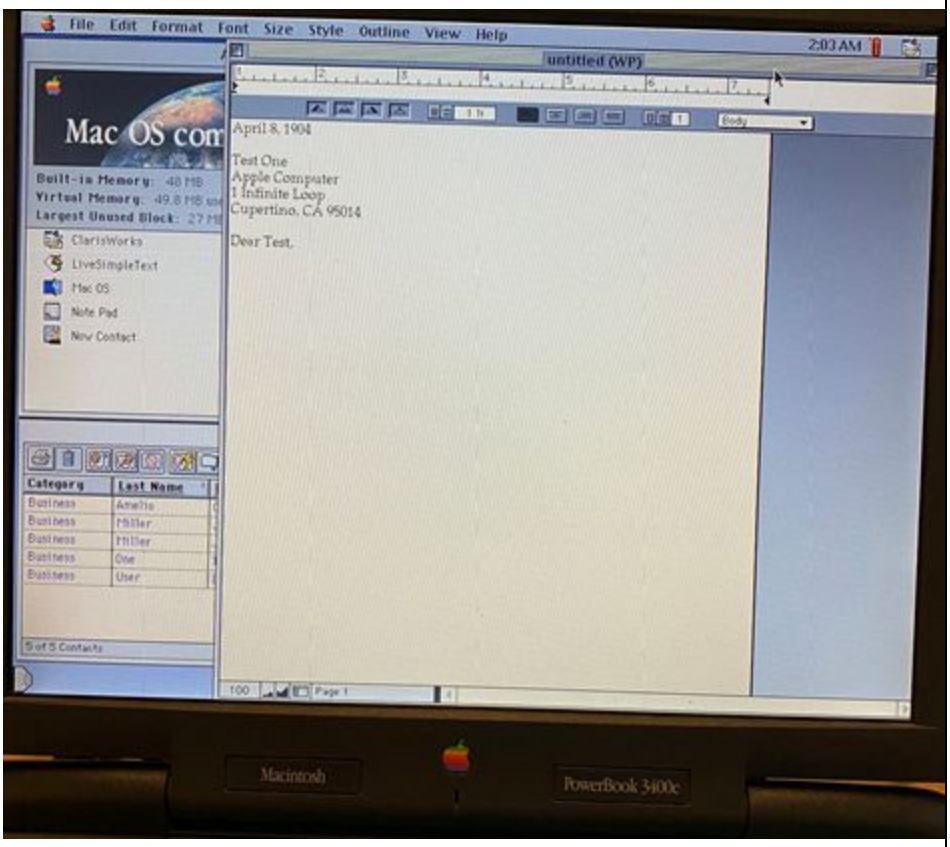
	 File Edit Format Font Size Style Outline View Help 2:03 AM untitled (WP) April 8, 1904 Test One Apple Computer 1 Infinite Loop Cupertino, CA 95014 Dear Test, Macintosh PowerBook 3400c

Exhibit E

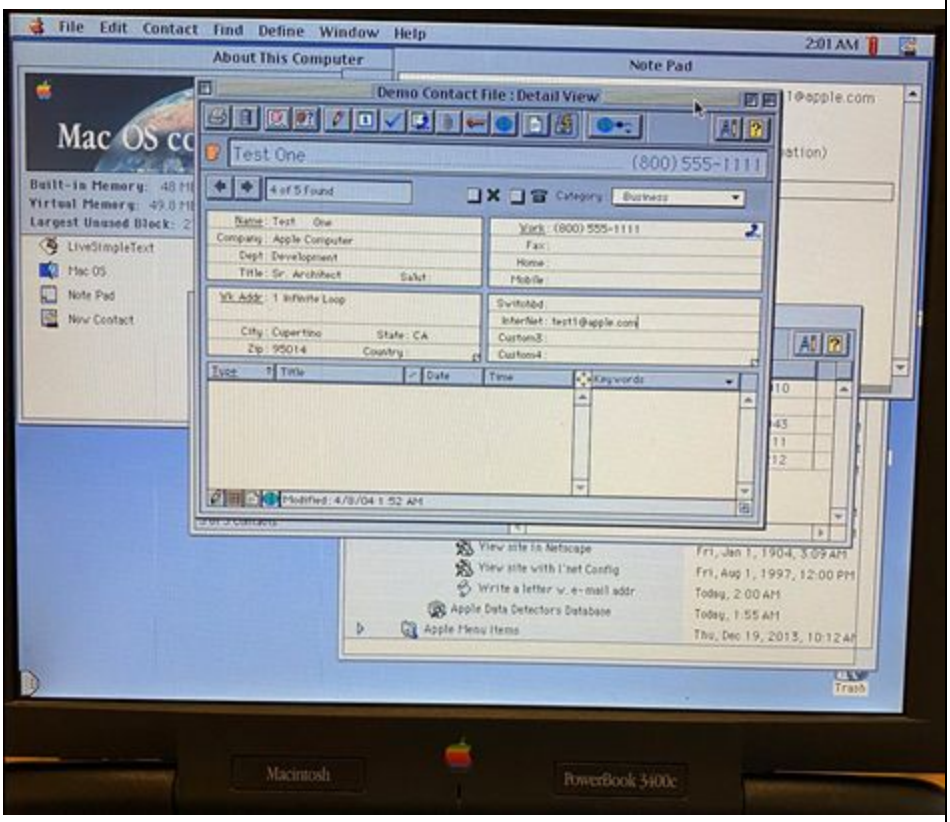
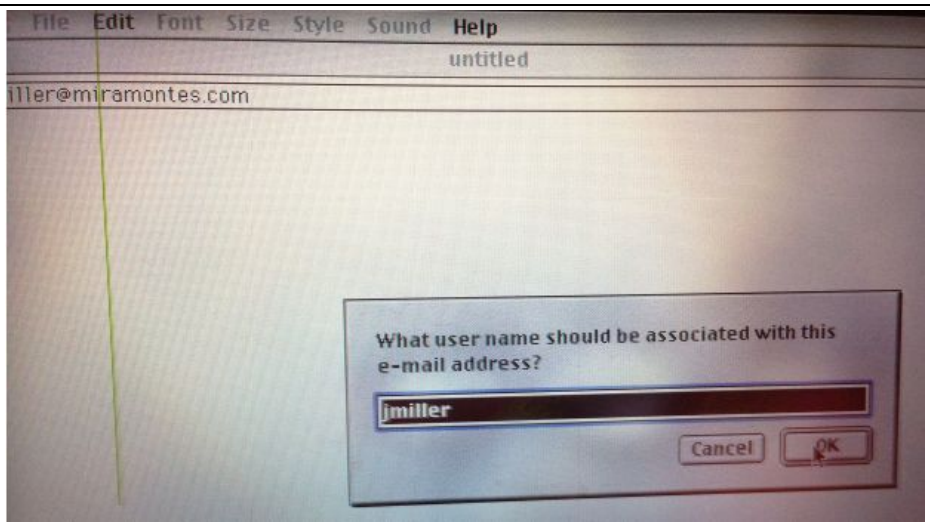
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	The Apple Data Detectors System discloses this element. If a user selects "Add email address to Emailer Address Book," IAD will ask the user for a name to associate with the email address. For example:

Exhibit E



See also MacWorld 1997 Expo at 1:11:00:

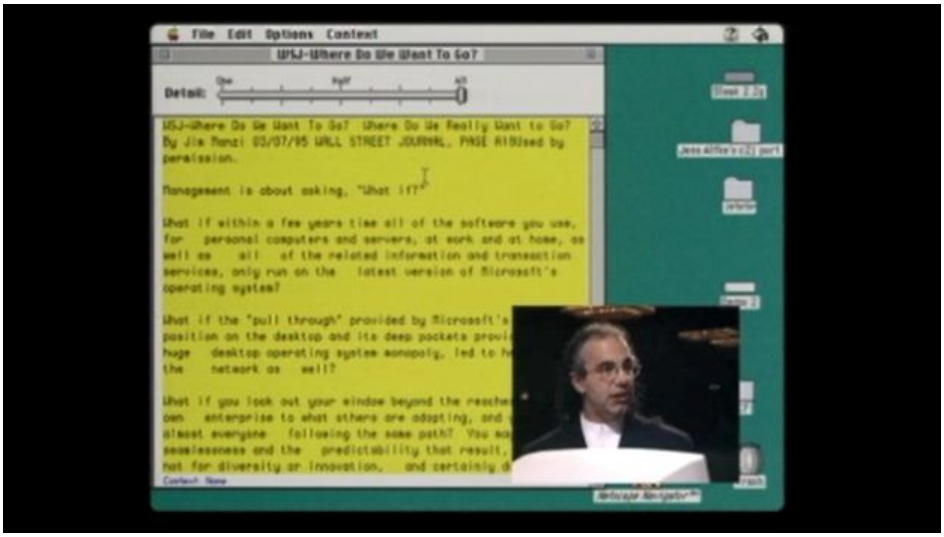
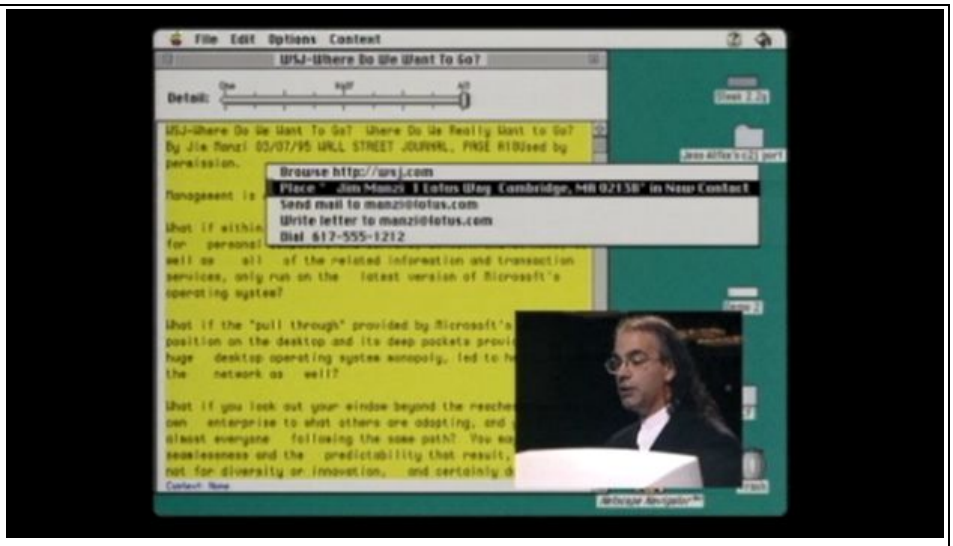
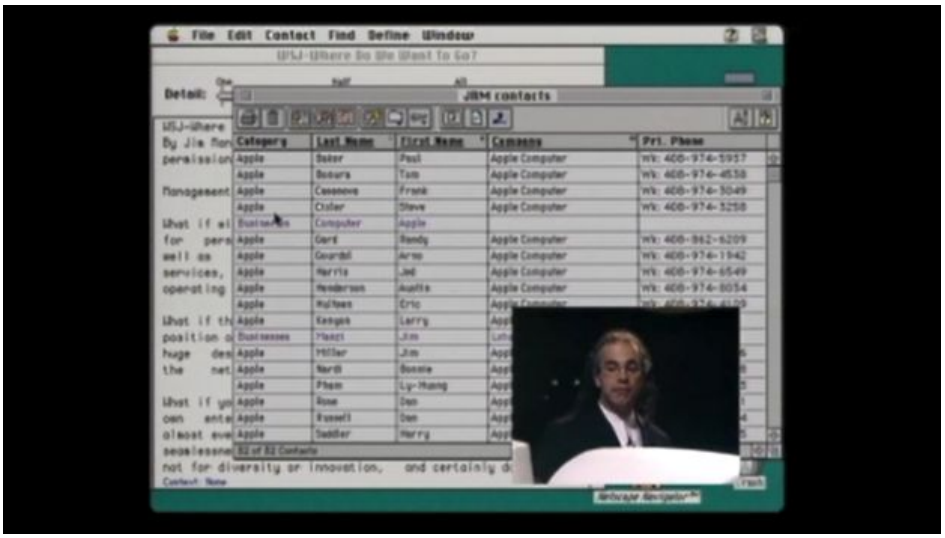


Exhibit E





See also Nardi at 101: "Apple Data Detectors assumes a world of heterogenous data (in the user's machine) that comes from different applications in different file formats. For example, it makes sense to put an address in an address book, whether the address is from a message sent in any of several mail programs, appears in a downloaded document, or is in another address book maintained by the user. Apple Data Detectors is a pervasive technology, giving users access to actions appropriate for data in an entire set of documents."

See also Nardi at 101: "The system's current user interface allows users to select data and actions, resulting in a collaborative agent. The user participates by signaling as structures of interest occur in a document and by verifying that an action is appropriate by selecting it from the menu, as shown in Figure 1. Apple Data Detectors participates by recognizing structures, offering appropriate actions, sending data to the

Exhibit E

	target application, opening the target application, and performing any other actions specified in the action scripts." See also Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." See also Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." See also Nardi at 103: "Our early contact with the product group again served us well. During the system's development, we experimented with various user interfaces to its basic technology, an effort that paid off in a number of ways. The different interfaces followed different assumptions about various aspects of the system, such as the interface techniques themselves, the demand the techniques would place on the underlying operating system, and the demand imposed on developers who wanted to adapt Apple Data Detectors to their own uses. It was important for us to understand such trade-offs, so we could advise the product group on the technology's possibilities." The system provided an option to add a name to Now Contact or to add an email address to Now Contact. *See* screenshot above, Video at 1:15-1:40. Miller teaches: When a user selects a detected structure, the system prompts the user to select a candidate action by displaying a pop-up menu of actions. *See, e.g.,* 4:27-31 ("Upon selection of this structure, user interface 240

Exhibit E

	presents and enables selection of the linked candidate actions using any selection mechanism, such as a conventional pull-down or pop-up menu."). As shown in Fig. 4, several of these actions update an information source to include first information (e.g., put in address book). *See, e.g.,* Fig. 4; 5:6-18. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book." Miller at 5:38-43. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98). "It was possible to add an address to some contact managers' address book. I believe it was Now Contact." Miller Depo. at 84:18-20. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. For example, during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Add e-mail address to Emailer address book" from the sub-menu, the Claris Emailer application is launched, and the user is asked to associate a user name with the detected email address. In this example, "Test Name 1" is associated with "test1@apple.com" and added to the Claris Emailer Address Book. See also screenshots and text associated with the example "test2@generic.com" in the ("providing an input device, configured by the first computer program . . . ") element.

Exhibit E

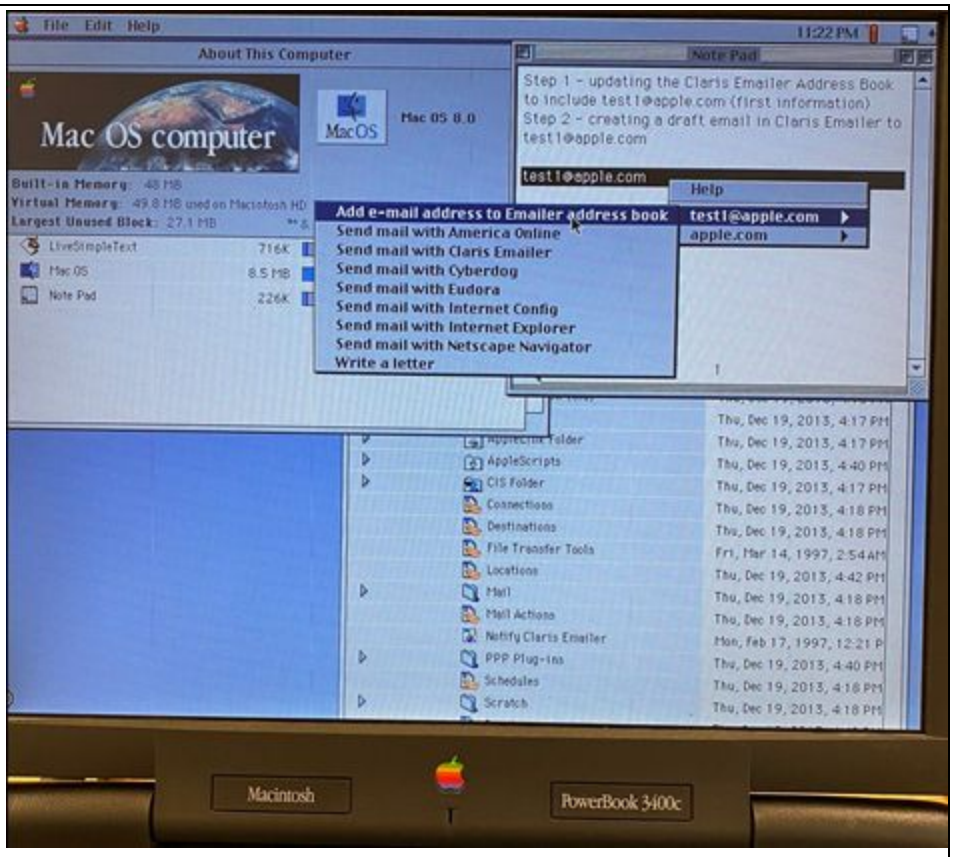
	 File Edit Help 11:22 PM About This Computer Mac OS computer MacOS Mac OS 8.0 Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 27.1 MB LiveSimpleText 716K Mac OS 8.5 MB Note Pad 226K Note Pad Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information) Step 2 - creating a draft email in Claris Emailer to test1@apple.com test1@apple.com Help Add e-mail address to Emailer address book Send mail with America Online Send mail with Claris Emailer Send mail with Cyberdog Send mail with Eudora Send mail with Internet Config Send mail with Internet Explorer Send mail with Netscape Navigator Write a letter test1@apple.com apple.com AppleScripts CIS Folder Connections Destinations File Transfer Tools Locations Mail Mail Actions Notify Claris Emailer PPP Plug-ins Schedules Scratch Macintosh PowerBook 3400c

Exhibit E

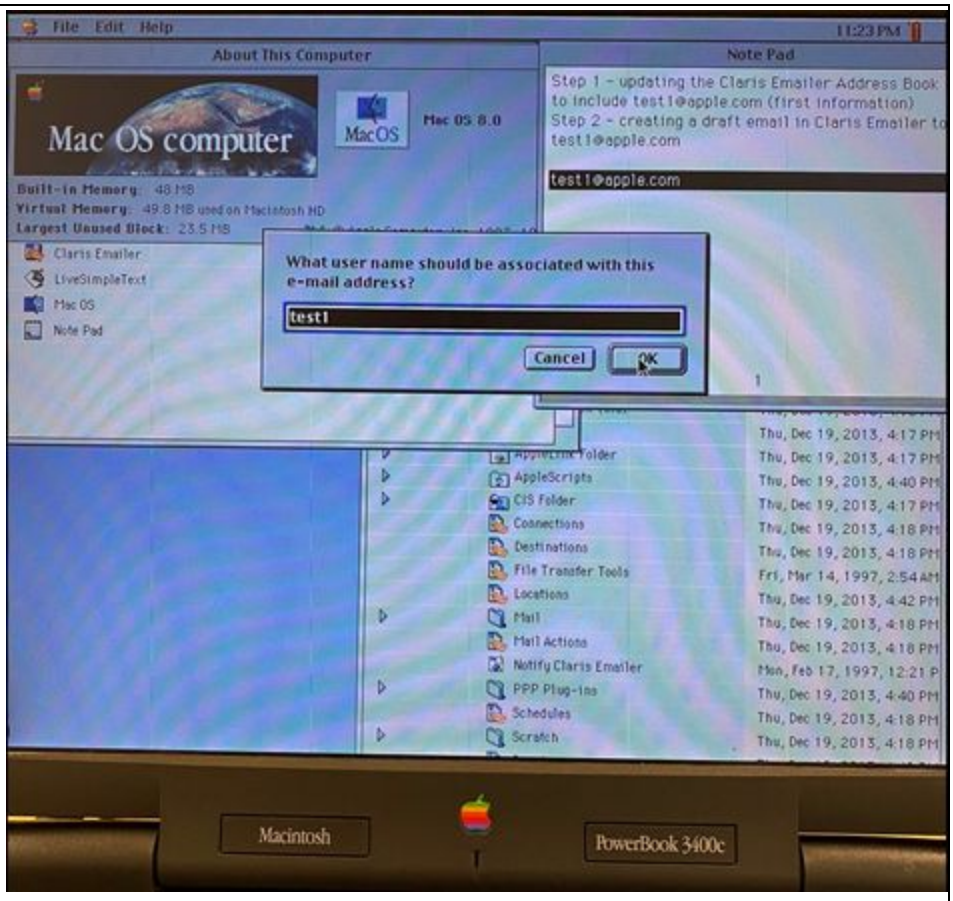

File Edit Help
11:23 PM
About This Computer
Mac OS computer
MacOS
Mac OS 8.0
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Macintosh HD
Largest Unused Block: 23.5 MB
Claris Emailer
LiveSimpleText
Mac OS
Note Pad
Note Pad
Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information)
Step 2 - creating a draft email in Claris Emailer to test1@apple.com
test1@apple.com
What user name should be associated with this e-mail address?
test1
Cancel
OK
AppleScripts
CIS Folder
Connections
Destinations
File Transfer Tools
Locations
Mail
Mail Actions
Notify Claris Emailer
PPP Plug-ins
Schedules
Scratch
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:17 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Fri, Mar 14, 1997, 2:54 AM
Thu, Dec 19, 2013, 4:42 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Mon, Feb 17, 1997, 12:21 P
Thu, Dec 19, 2013, 4:40 PM
Thu, Dec 19, 2013, 4:18 PM
Thu, Dec 19, 2013, 4:18 PM
Macintosh
PowerBook 3400c

Exhibit E

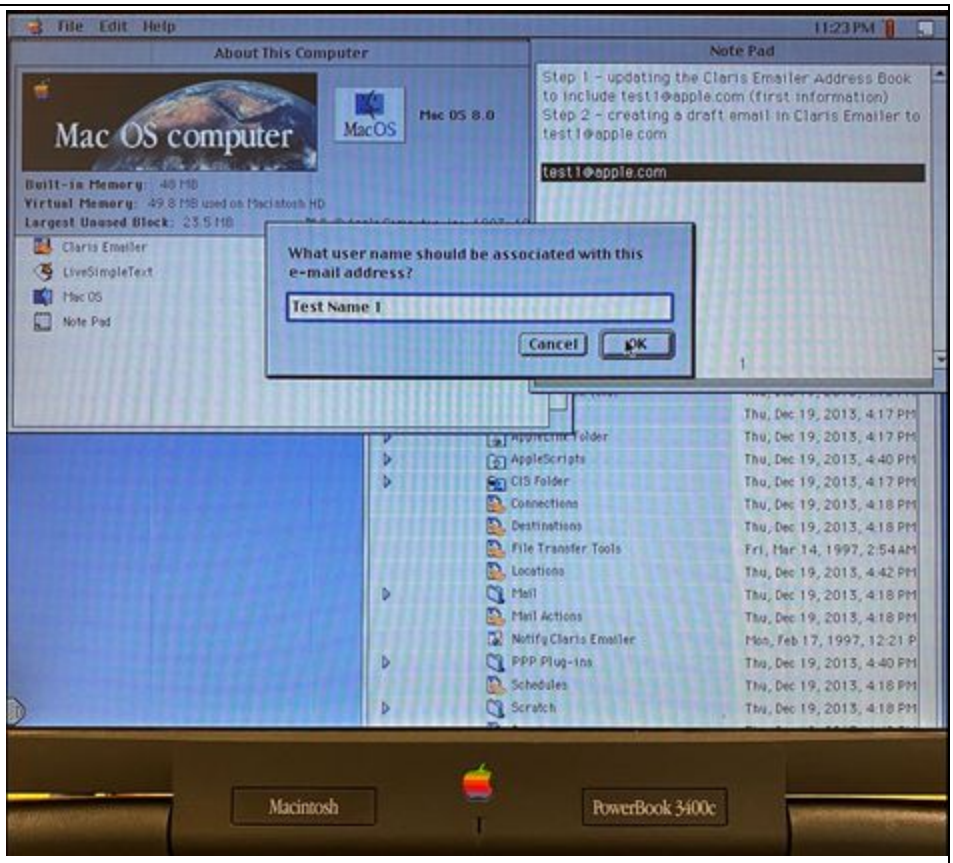
	 File Edit Help 11:23 PM About This Computer Mac OS computer MacOS Mac OS 8.0 Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 23.5 MB Claris Emailer LiveSimpleText Mac OS Note Pad Note Pad Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information) Step 2 - creating a draft email in Claris Emailer to test1@apple.com test1@apple.com What user name should be associated with this e-mail address? Test Name 1 Cancel OK AppleScripts CIS Folder Connections Destinations File Transfer Tools Locations Mail Mail Actions Notify Claris Emailer PPP Plug-ins Schedules Scratch Macintosh PowerBook 3400c

Exhibit E

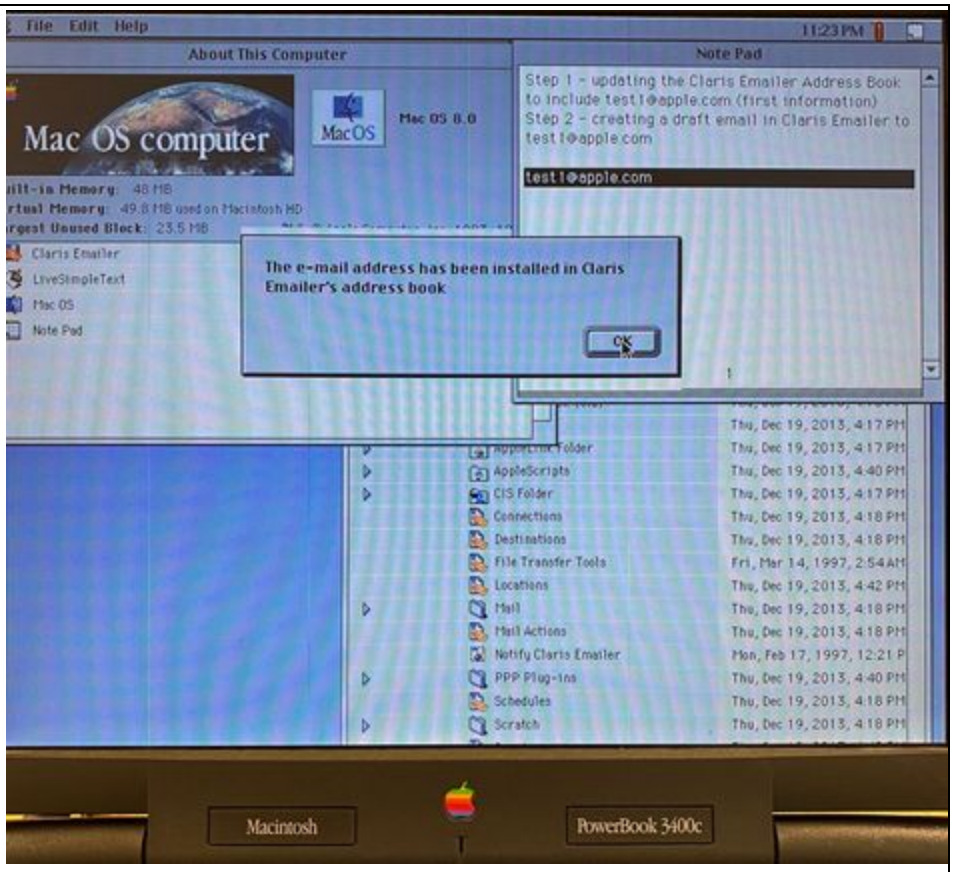
	 File Edit Help 11:23 PM About This Computer Mac OS computer MacOS Mac OS 8.0 Note Pad Step 1 - updating the Claris Emailer Address Book to include test1@apple.com (first information) Step 2 - creating a draft email in Claris Emailer to test1@apple.com test1@apple.com The e-mail address has been installed in Claris Emailer's address book OK Claris Emailer LiveSimpleText Mac OS Note Pad AppleScripts CIS Folder Connections Destinations File Transfer Tools Locations Mail Mail Actions Notify Claris Emailer PPP Plug-ins Schedules Scratch Macintosh PowerBook 3400c

Exhibit E

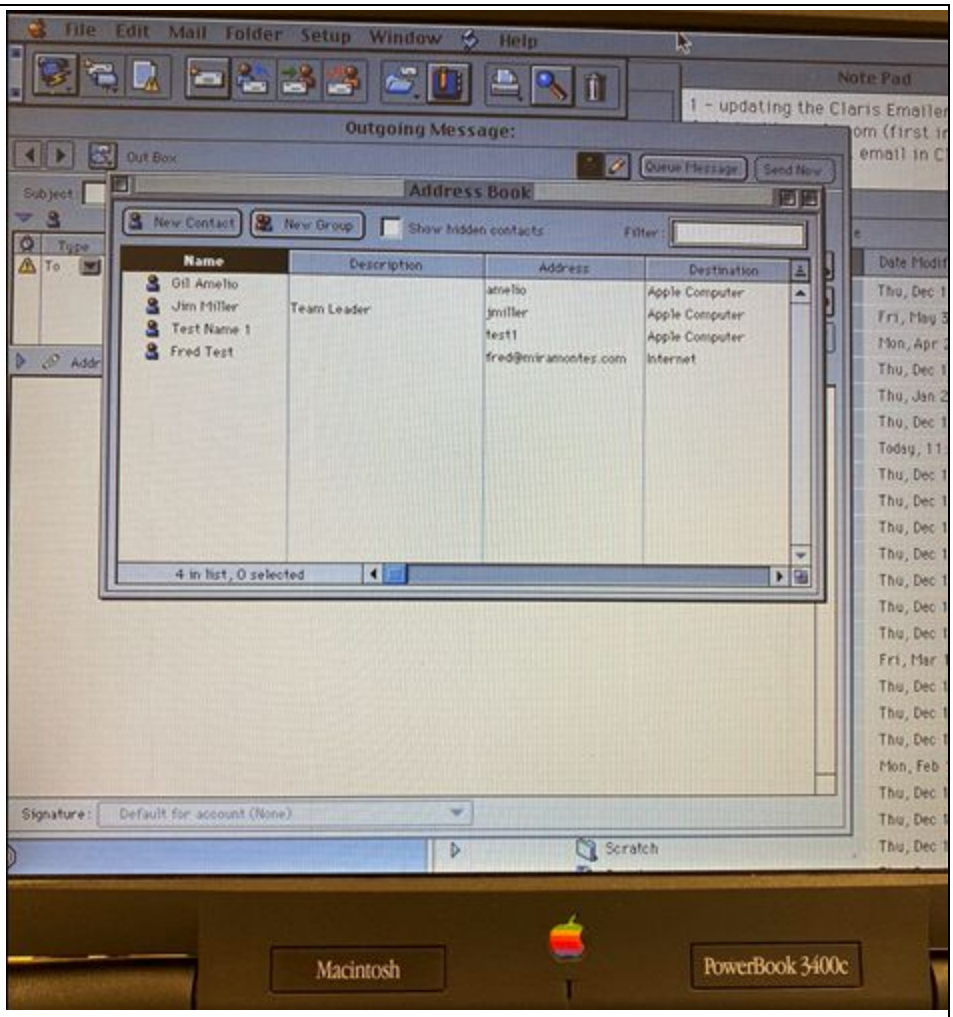
	 For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Write a letter" from the sub-menu, the Now Contact application is launched, and, if the email address is not present in the Now Contact "Demo Contact File," the user can create a "New Contact" in the "Demo Contact File." In this example, the detected email address ("test1@apple.com") along with first name ("Test"), last name ("One"), company ("Apple Computer"), department ("Development"), title ("Sr. Architect"), work address ("1 Infinite Loop"), city ("Cupertino"), state ("CA"), zip code ("95014"), and work telephone number ("800 555-1111") was saved in a contact entry in the "Demo Contact File." The "InterNet" field of the contact entry contains the email address for the contact ("test1@apple.com").

Exhibit E

	 File Edit Help 1:49 AM About This Computer Mac OS computer Mac OS Built-in Memory: 48 MB Virtual Memory: Largest Unused Block: LiveSimpleText Mac OS Note Pad Note Pad Step 1 - updating the Now Contact DB to include test1@apple.com (first information) Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information) test1@apple.com Help test1@apple.com apple.com Add e-mail address to Emailer address book Send mail with America Online Send mail with Claris Emailer Send mail with Cyberdog Send mail with Eudora Send mail with Internet Config Send mail with Internet Explorer Send mail with Netscape Navigator Write a letter AppleScripts CIS Folder Connections Destinations File Transfer Tools Locations Mail Mail Actions Notify Claris Emailer PPP Plug-ins Schedules Scratch Macintosh PowerBook 3400c

Exhibit E

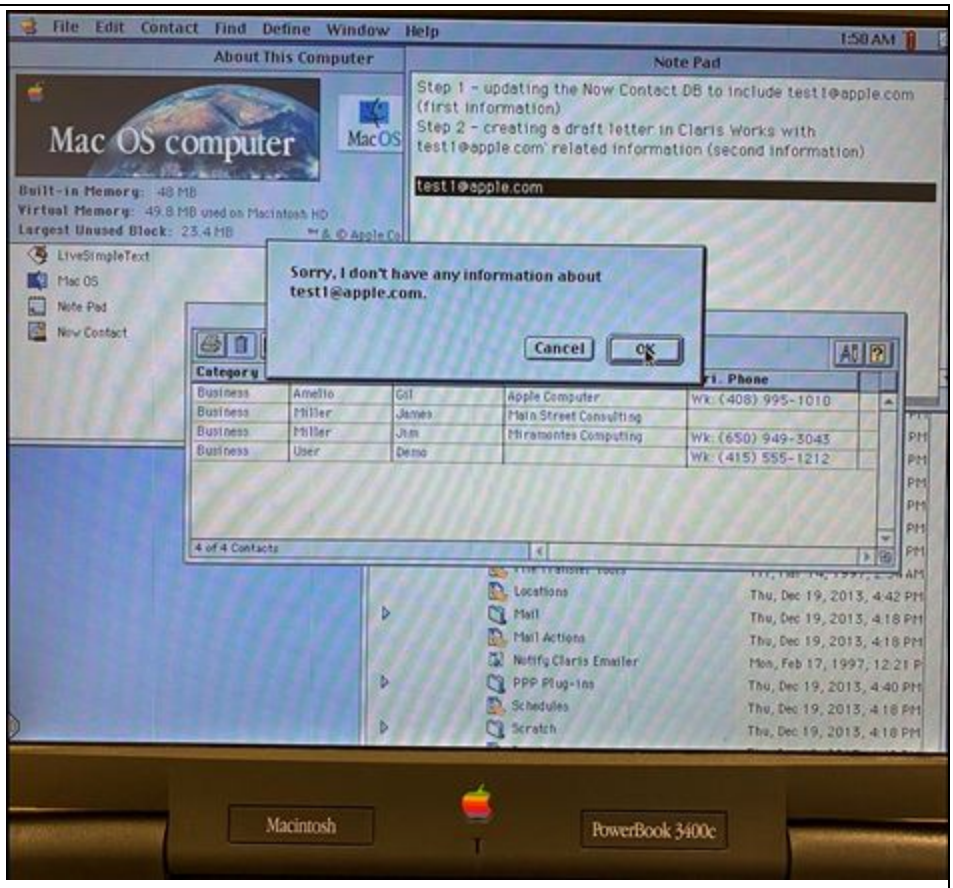
	 File Edit Contact Find Define Window Help 1:50 AM About This Computer Mac OS computer Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 23.4 MB LiveSimpleText Mac OS Note Pad New Contact Note Pad Step 1 - updating the Now Contact DB to include test1@apple.com (first information) Step 2 - creating a draft letter in Claris Works with test1@apple.com related information (second information) test1@apple.com Sorry, I don't have any information about test1@apple.com. Cancel OK Category Business Amelio Gil Apple Computer Wk: (408) 995-1010 Business Miller James Main Street Consulting Business Miller Jim Miramontes Computing Wk: (650) 949-3043 Business User Demo Wk: (415) 555-1212 4 of 4 Contacts Locations Thu, Dec 19, 2013, 4:42 PM Mail Thu, Dec 19, 2013, 4:18 PM Mail Actions Thu, Dec 19, 2013, 4:18 PM Notify Claris Emailer Mon, Feb 17, 1997, 12:21 P PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM Schedules Thu, Dec 19, 2013, 4:18 PM Scratch Thu, Dec 19, 2013, 4:18 PM Macintosh PowerBook 3400c

Exhibit E

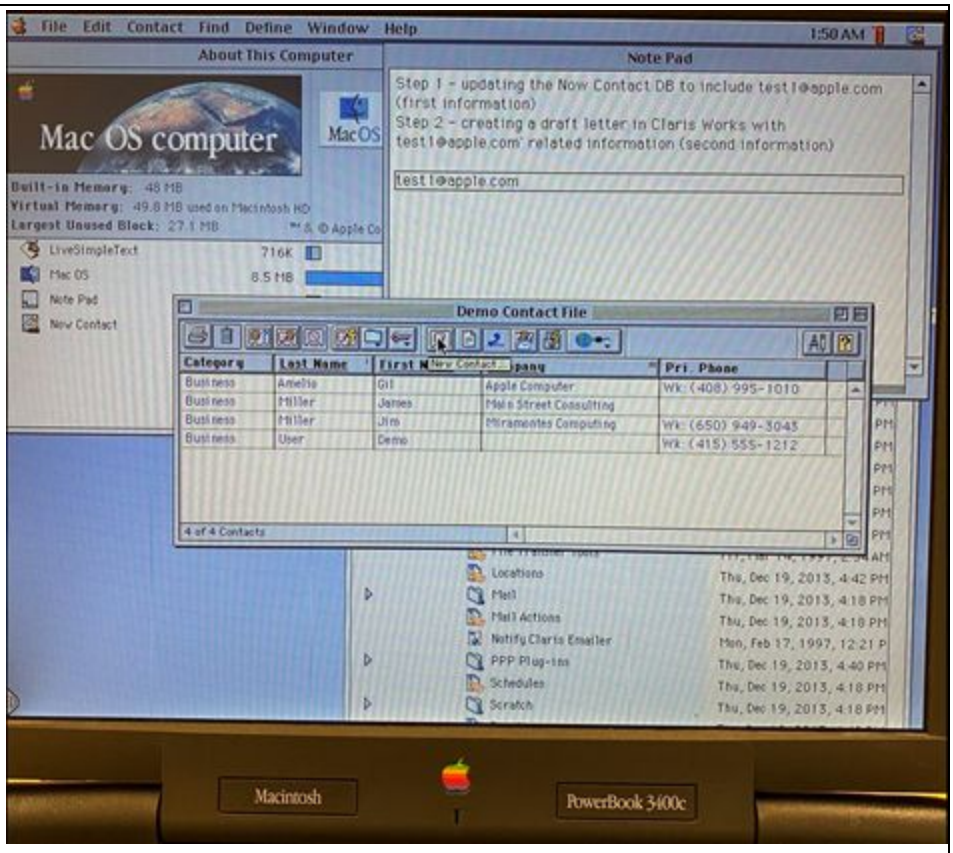
	 File Edit Contact Find Define Window Help 1:50 AM About This Computer Mac OS computer Mac OS Built-in Memory: 48 MB Virtual Memory: 49.8 MB used on Macintosh HD Largest Unused Block: 27.1 MB LiveSimpleText 716K Mac OS 8.5 MB Note Pad New Contact Note Pad Step 1 - updating the New Contact DB to include test1@apple.com (first information) Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information) test1@apple.com Demo Contact File Category Last Name First Name Company Pri. Phone Business Amelio Gil Apple Computer Wk: (408) 995-1010 Business Miller James Main Street Consulting Business Miller Jim Miramontes Computing Wk: (650) 949-3045 Business User Demo Wk: (415) 555-1212 4 of 4 Contacts Locations Thu, Dec 19, 2013, 4:42 PM Mail Thu, Dec 19, 2013, 4:18 PM Mail Actions Thu, Dec 19, 2013, 4:18 PM Notify Claris Emailer Mon, Feb 17, 1997, 12:21 P PPP Plug-ins Thu, Dec 19, 2013, 4:40 PM Schedules Thu, Dec 19, 2013, 4:18 PM Scratch Thu, Dec 19, 2013, 4:18 PM Macintosh PowerBook 3400c

Exhibit E

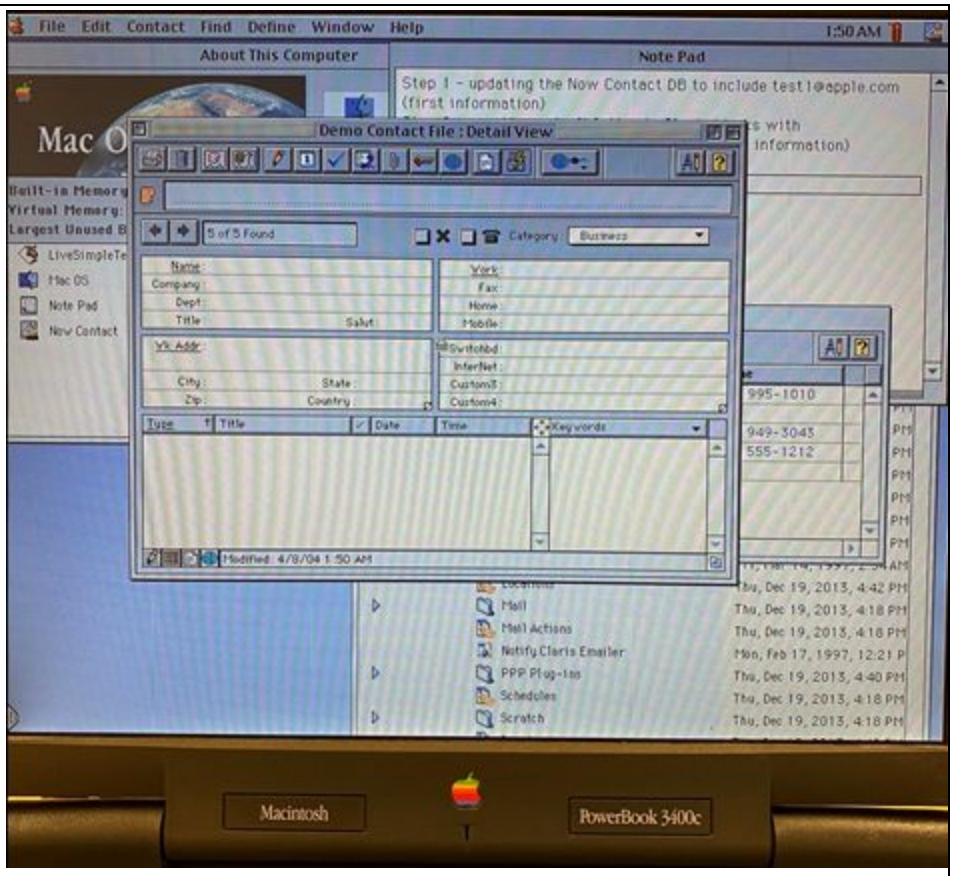
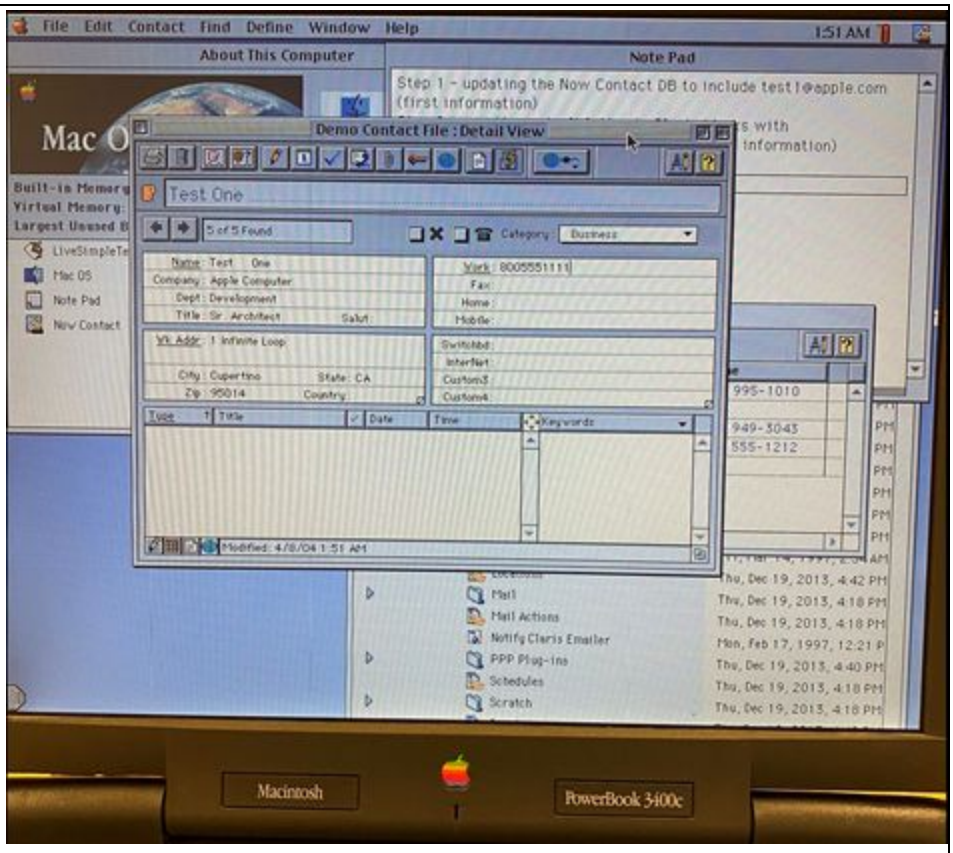
	 File Edit Contact Find Define Window Help 1:50 AM About This Computer Note Pad Step 1 - updating the Now Contact DB to include test1@apple.com (first information) Demo Contact File : Detail View Macintosh PowerBook 3400c

Exhibit E


File Edit Contact Find Define Window Help
1:51 AM
About This Computer
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com
(first information)
Demo Contact File : Detail View
Test One
Macintosh
PowerBook 3400c

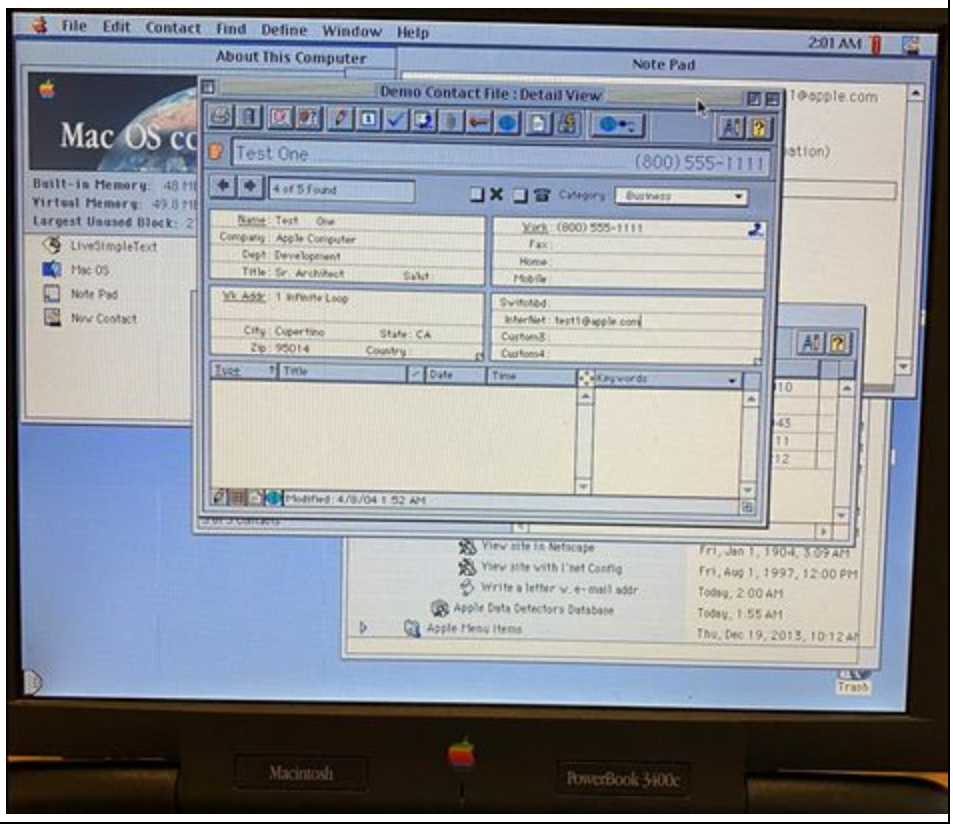

File Edit Contact Find Define Window Help
2:01 AM
About This Computer
Note Pad
Demo Contact File : Detail View
Test One
(800) 555-1111
Macintosh
PowerBook 3400c

Exhibit E

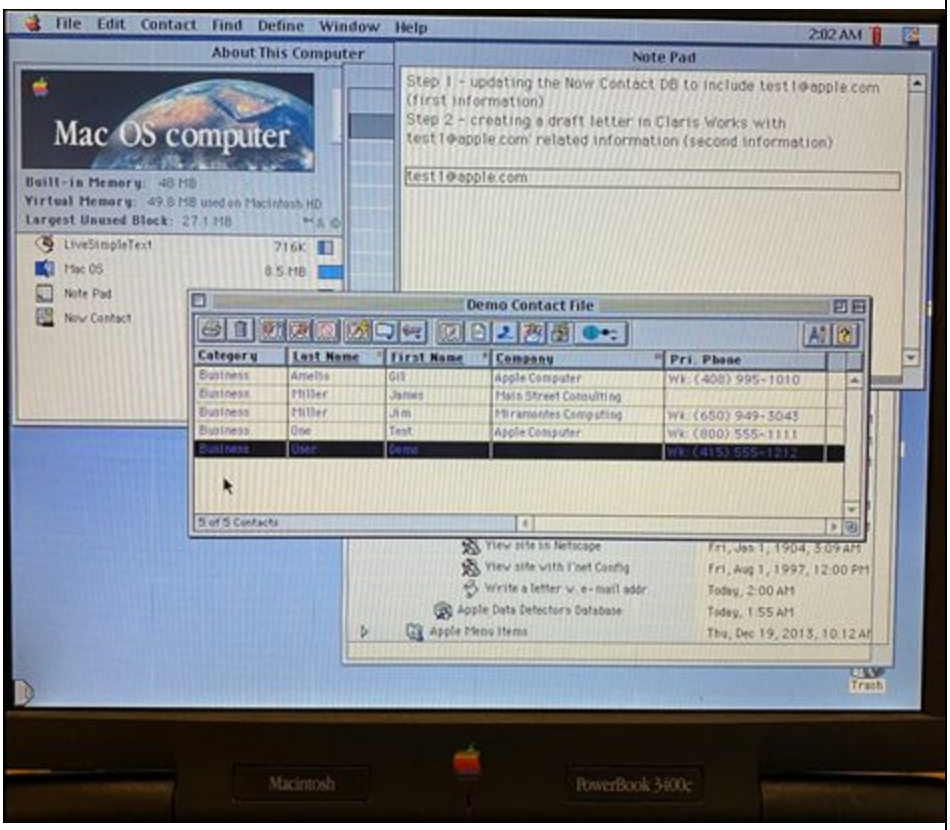
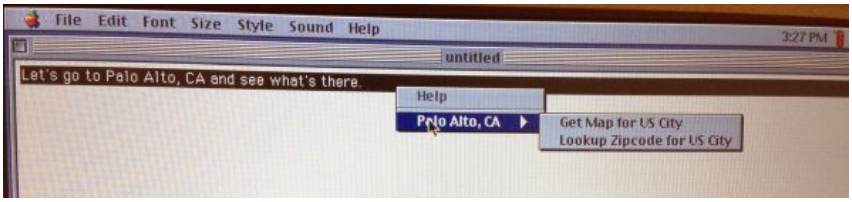
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	The Apple Data Detectors System discloses this element. When IAD detects an email address and a user selects “Send mail with Claris Emailer,” Claris Emailer will search its information source for a company name associated with the domain of the email address and will insert this into the Outgoing Message panel for a new email. For example:

Exhibit E

When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could obtain a zip code for the selected US city, for which the system would search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:



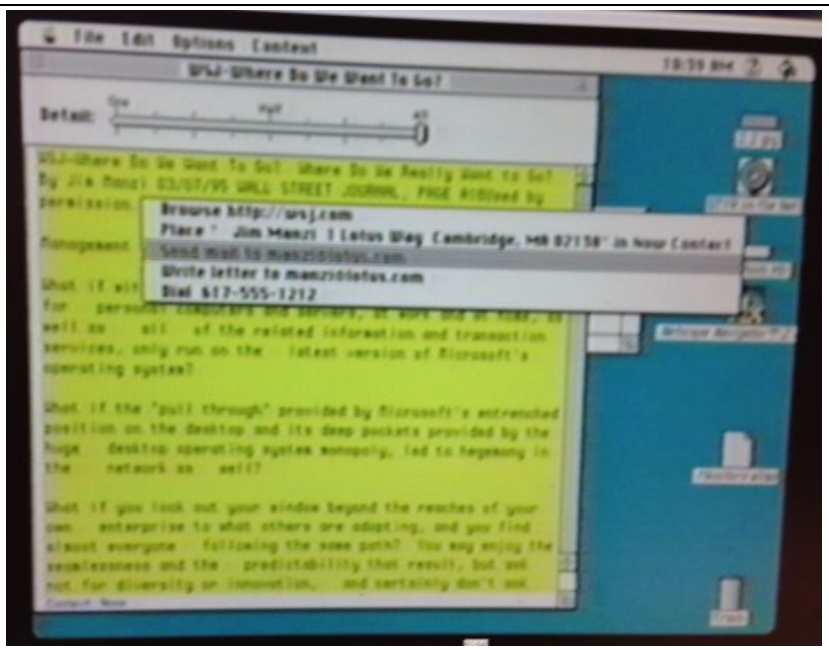
This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

See also Nardi at pp. 96-98 (including figs. 1, 2): "As Apple Computer researchers, we started from a simple but focused approach to agents: That they should have the ability to infer appropriate high-level goals from user actions and requests and take action to achieve these goals. Further, based on a study of reference librarians as exemplary human agents [9], we wanted to build a system in which the user would not have to state goals explicitly and in detail. We learned from librarians that a large part of their value to clients is in working with imprecise requests. Beyond this concern, our general design strategy was to keep the most basic user question in front of us at all times: Will this software

Exhibit E

	do something useful for users in an intelligent way that makes them more productive? The system we describe here—Apple Data Detectors—meets our criteria of being unobtrusive, being able to infer user needs, and doing useful work. Apple Data Detectors shipped as a product in 1997. Earlier work on intelligent agents was multifaceted, to the point where it is difficult to find a consensus among researchers on exactly what constitutes an 'agent' or even 'intelligence.' However, in nearly all cases, systems described as 'agent-based' rely on some explicitly represented knowledge about relevant aspects of the world—the objects or concepts being addressed by the software, the tasks relevant to the user, and the user's own knowledge about the world. Researchers have used machine learning techniques to track user actions and construct models of user preferences [7], create explicit models of user knowledge and skill levels in an attempt to anticipate user actions, misconceptions, and information needs [2], and implement planning systems to leap from a user's stated intention to the specific actions required to achieve that intention [3]. The locality of agents also varies across different agent-based systems; some act only within one's own machine, find others autonomously crawl the Web, searching for interesting content [4]. We tried to find a middle ground by using explicit representations of user-relevant information as a means of identifying actions users might wish to take but to leave the choice of these actions to users." See also Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." See also Nardi at fig. 1. Mac World Data Detectors System could detect data structures in a written document and provide options for the detected structures. For example:

Exhibit E



Video.

See also MacWorld Expo 1997 at 1:11:00:

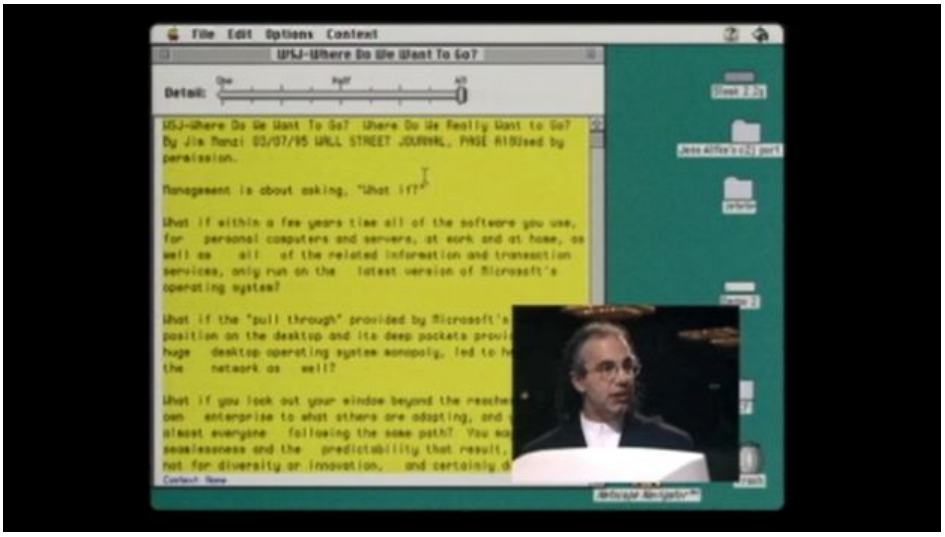
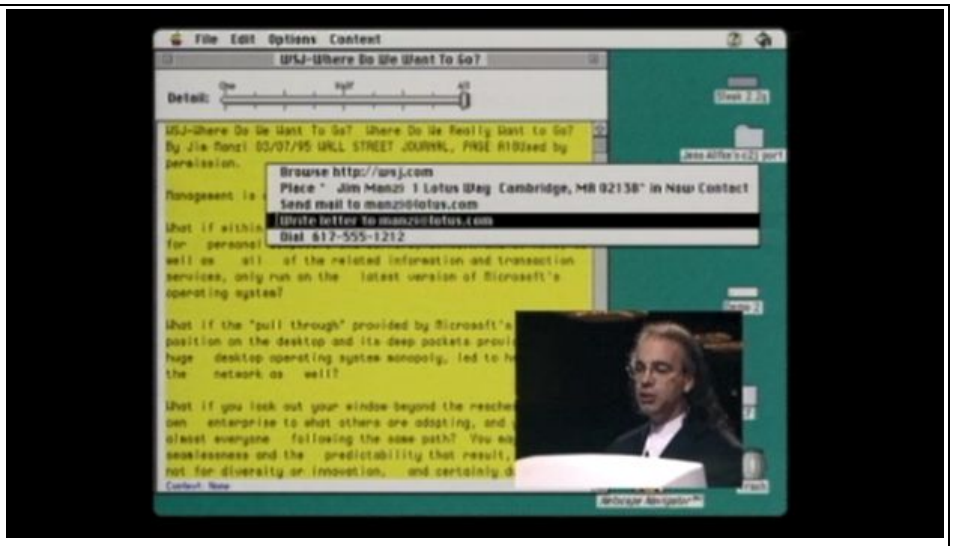
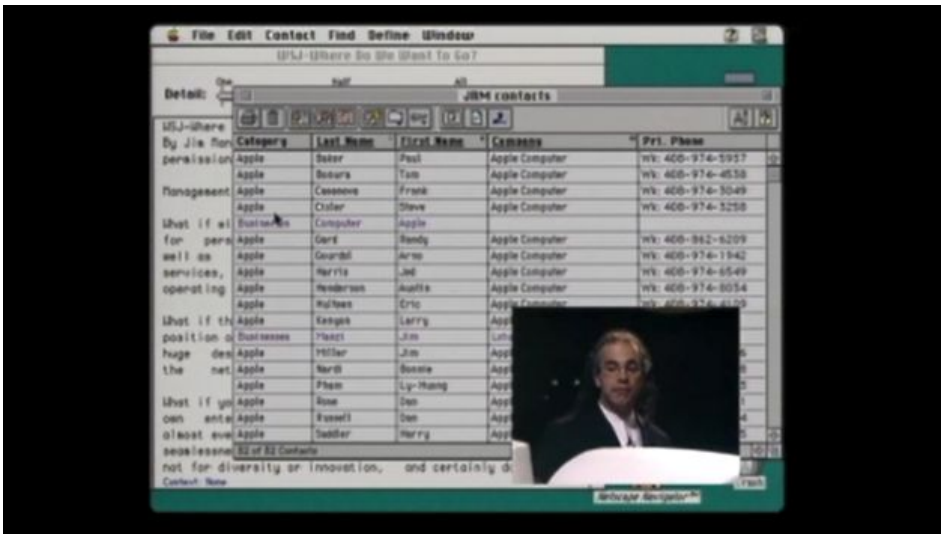


Exhibit E


File Edit Options Content
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212


File Edit Contact Find Define Window
WSJ-Where Do We Want To Go?
Detail:
JBM contacts
Category
Last Name
First Name
Company
Pri. Phone

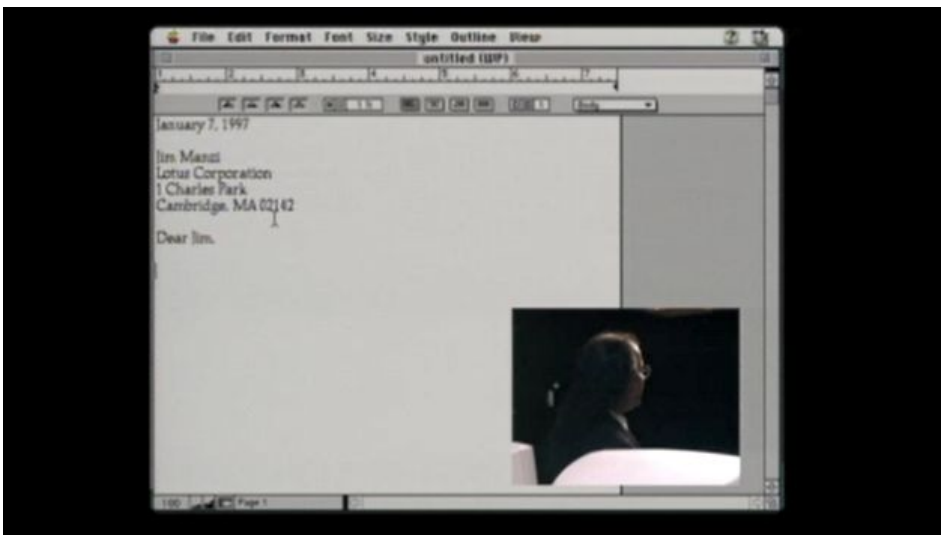

File Edit Format Font Size Style Outline View
untitled (WP)
January 7, 1997
Jim Manzi
Lotus Corporation
1 Charles Park
Cambridge, MA 02142
Dear Jim,

Exhibit E

	“**on** addressLetterTo(emailAddress) **set** emailAddress **to** removeTextStyling(emailAddress) **tell** application “Now Contact 3.5” launch activate run *--find the person in the Now Contact database* **try** **set** thePerson **to the first** person **whose** (customer 2 is emailAddress) **on error** beep display dialog “Sorry, I don’t have any information about “ & emailAddress & “.” **return** **end try** *--get the address information for the person* **try** **set** firstAndLastName **to** (**the** first name **of** thePerson) & “ ” & (**the** last name **of** thePerson) **set** theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (the work address **of** thePerson) & return & (**the** work city **of** thePerson) & “, ” & (**the** work state **of** thePerson) & “ ” & (**the** work zip of thePerson) & return **on error** beep display dialog “Sorry, I couldn’t get the address of “ & emailAddress 7 “.” **return** **end try** *--write the address information into ClarisWorks* **tell**application “ClarisWorks” **try** launch activate *--run --don’t do a “run here, or you’ll get the “New document” dialog…* open (path to system folder as string) & “SD letter template” as “WP” **on error** beep

Exhibit E

activate
display dialog “Sorry, I couldn’t open the letter template: ‘SD letter template’ in the System Folder.”
See Write a Letter AppleScript Code.

"As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250."

Miller at 5:38-47.

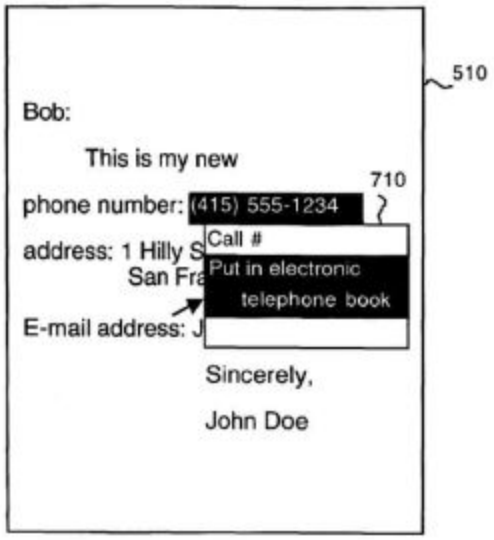


FIG. 7

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an

Exhibit E

	address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98). "And why was it important that – to allow users to do otherwise time-consuming actions on documents with a single click of the mouse? A. It makes their interaction with their information much easier." Miller Depo. at 70:14-19. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things,"

Exhibit E

	who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple

Exhibit E

	event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And so looking back at the screenshot that is Number 2 in this Miller Exhibit 14, am I correct that when Lotus – I'm sorry, when Apple Data Detectors are invoked on this document, the user is given, for this particular document, five different options of things that he or show can do? A. Yes. * * * Q. And the user can click any of these with a single click of the mouse? A. Yes." Miller Depo. at 129:17-130:18. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get

Exhibit E

	me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at

Exhibit E

	161:16-162:4. "Q. When you say 'pointing at a highlight and pressing the mouse button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that

Exhibit E

	are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Send mail with Claris Emailer" from the sub-menu, the Claris Emailer application is launched, and an "Outgoing Message" window is presented in the Claris Emailer application. In this example, "Apple Computer" is associated with "test1@apple.com" in the Claris Emailer Address Book, and is displayed in the "Outgoing Message" window in the "Destination" field. See also additional screenshots and text associated with this example or with the example "test2@generic.com" in the ("providing an input device, configured by the first computer program . . . ") element. 

Exhibit E

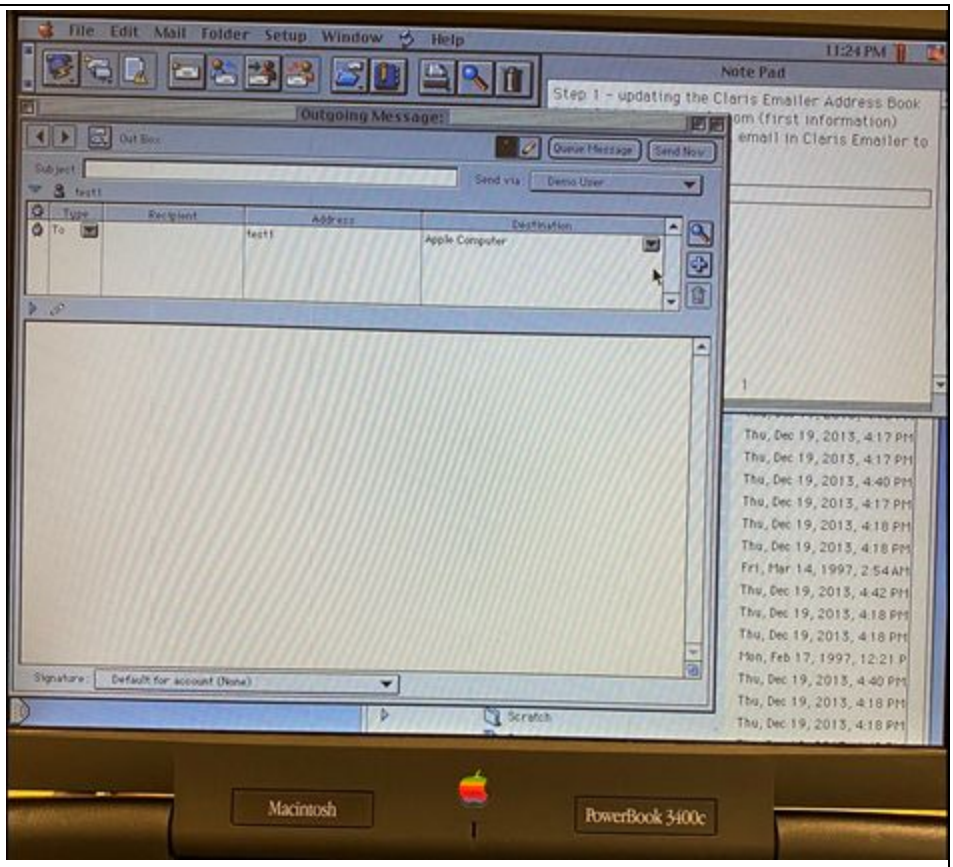
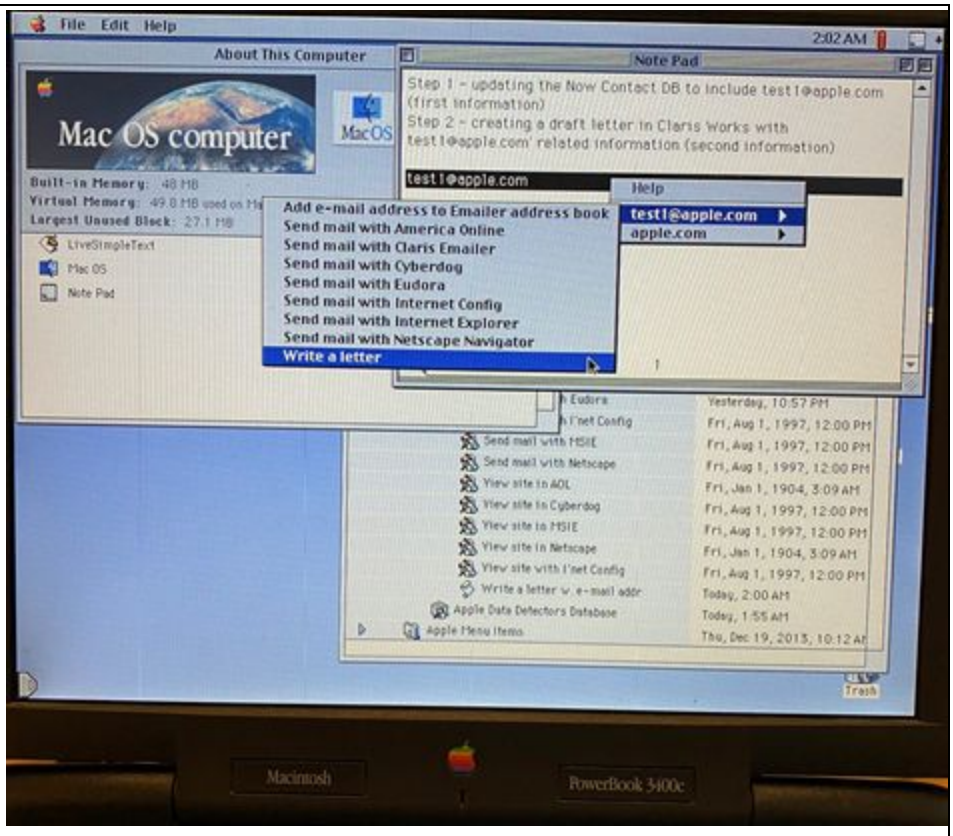
	 For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Note Pad application, and the user selects the associated action of "Write a letter" from the sub-menu, the Now Contact and Claris Works applications are launched, and an "untitled (WP)" window is presented in the Claris Emailer application. In this example, the first name ("Test"), last name ("One"), and physical address information ("Apple Computer 1 Infinite Loop Cupertino, CA 95014") are associated with "test1@apple.com" in the Now Contact Demo Contact File, and is displayed in the "untitled (WP)" window. See also additional screenshots and text associated with this example in the ("providing an input device, configured by the first computer program . . . ") element.

Exhibit E


File Edit Help
2:02 AM
About This Computer
Mac OS computer
MacOS
Note Pad
Step 1 - updating the Now Contact DB to include test1@apple.com (first information)
Step 2 - creating a draft letter in Claris Works with test1@apple.com' related information (second information)
test1@apple.com
Help
test1@apple.com
apple.com
Add e-mail address to Emailer address book
Send mail with America Online
Send mail with Claris Emailer
Send mail with Cyberdog
Send mail with Eudora
Send mail with Internet Config
Send mail with Internet Explorer
Send mail with Netscape Navigator
Write a letter
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB used on Ha
Largest Unused Block: 27.1 MB
LiveSimpleText
Mac OS
Note Pad
Macintosh
PowerBook 3400c

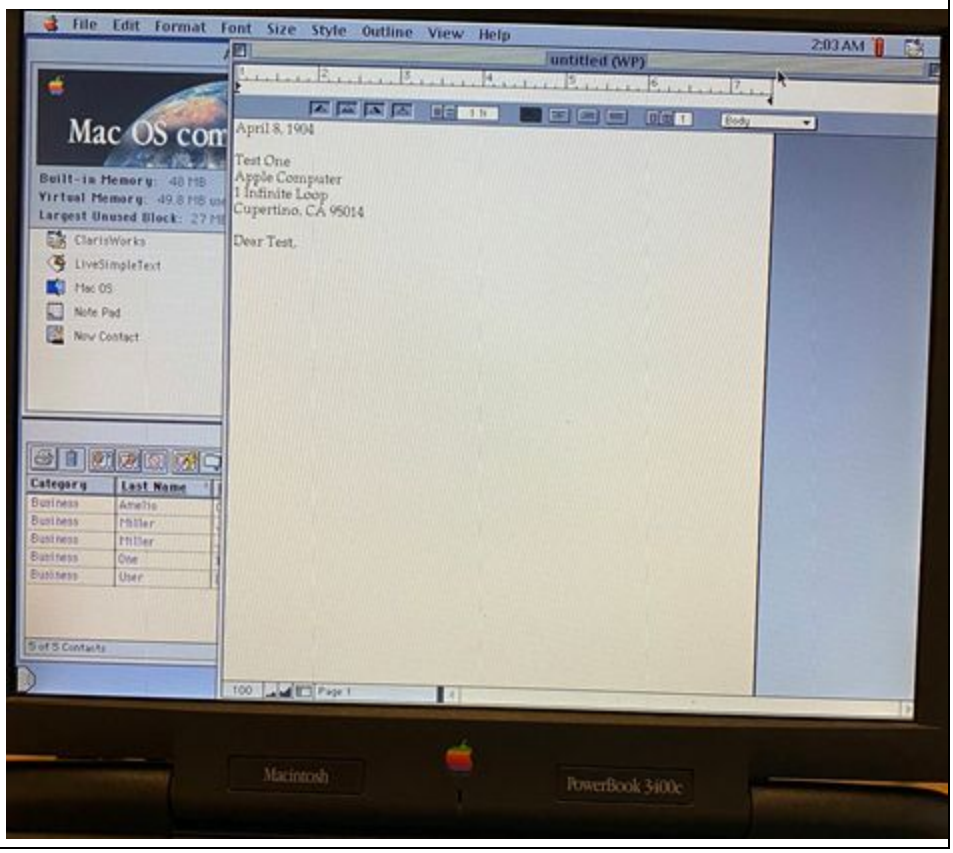

File Edit Format Font Size Style Outline View Help
2:03 AM
untitled (WP)
Mac OS com
Built-in Memory: 48 MB
Virtual Memory: 49.8 MB us
Largest Unused Block: 27 MB
ClarisWorks
LiveSimpleText
Mac OS
Note Pad
New Contact
April 8, 1904
Test One
Apple Computer
1 Infinite Loop
Cupertino, CA 95014
Dear Test,
Category
Last Name
Business
Amelio
Business
Miller
Business
Miller
Business
One
Business
User
5 of 5 Contacts
Macintosh
PowerBook 3400c

Exhibit E

	For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures. *See, e.g.*, Nardi at 101: "Apple Data Detectors assumes a world of heterogenous data (in the user's machine) that comes from different applications in different file formats. For example, it makes sense to put an address in an address book, whether the address is from a message sent in any of several mail programs, appears in a downloaded document, or is in another address book maintained by the user. Apple Data Detectors is a pervasive technology, giving users access to actions appropriate for data in an entire set of documents." *See also* Nardi at 101: "The system's current user interface allows users to select data and actions, resulting in a collaborative agent. The user participates by signaling as structures of interest occur in a document and by verifying that an action is appropriate by selecting it from the menu, as shown in Figure 1. Apple Data Detectors participates by recognizing structures, offering appropriate actions, sending data to the target application, opening the target application, and performing any other actions specified in the action scripts." *See also* Nardi at 101: "Apple Data Detectors therefore has the ability to infer appropriate high-level goals from user actions and requests and take appropriate action to achieve these goals. When users invoke it on a region of text in a document, they are saying, in effect, 'Find the important stuff in here and help me do reasonable things to it.' Users can be imprecise, throwing the system a broad hint that there is something of interest, then let the system use its knowledge to do the right thing. Users work on their tasks in terms of high-level goals, such as 'put this address in my address book'—not by opening folders, clicking on icons, cutting, and pasting. Direct manipulation is a wasteful, frustrating way for users to interact with machines capable of showing more intelligence." *See also* Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their

Exhibit E

	being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." *See also* Nardi at 103: "Our early contact with the product group again served us well. During the system's development, we experimented with various user interfaces to its basic technology, an effort that paid off in a number of ways. The different interfaces followed different assumptions about various aspects of the system, such as the interface techniques themselves, the demand the techniques would place on the underlying operating system, and the demand imposed on developers who wanted to adapt Apple Data Detectors to their own uses. It was important for us to understand such trade-offs, so we could advise the product group on the technology's possibilities." "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." Miller at 5:38-47.

Exhibit E

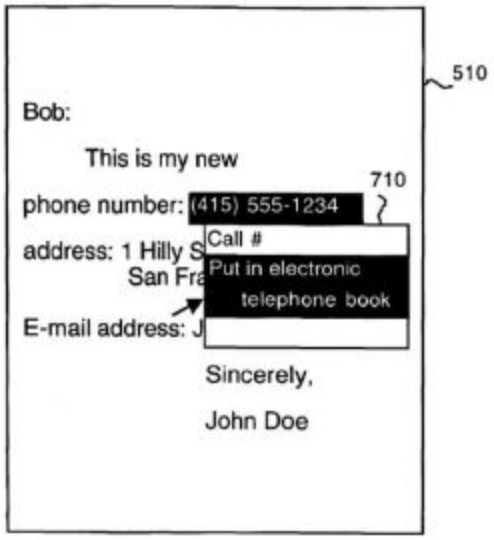


FIG. 7

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

Miller at 5:6-5:18.

See also, e.g., FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18.

The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... ").

“Am I correct that as we saw on the demo, you could, in fact, select the entire document to have Data Detectors review it?

Exhibit E

	A. Yes. Q. And am I correct that Data Detectors would work even if a user did not specifically designate, for example, the particular phone number that they were interested in acting on? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. Q. Why is it that you wanted to allow users to be able to analyze as much as all of the text in a document? A. Well, it's much easier to just very quickly select a region of text on the screen and let the computer figure out what things can be acted on in it as opposed to having to go in, in this case with a mouse, and very carefully only select the characters that you're interested in. It just requires less precise action on the part of the user. Q. What happens if there were multiple different data objects within the text of the letter, let's say, an email address and a phone number and a map address? Could the Data Detectors detect all of those things? A. Yes, that's what is shown in the figure at the bottom of, I guess, Page 2 of Miller 7. There is a highlighted region of text in the window and the hierarchical menus showing up on the page are showing a phone number and an email address and a URL." Miller Depo. at 71:13-72:20. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number

Exhibit E

	was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "Q. When you say 'pointing at a highlight and pressing the mouse

Exhibit E

	button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the

Exhibit E

	mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7 (*e.g.*, Luciw 735, Siitonen, Domini, and Schabes). A POSITA would have reasonable expectation of success to achieve predictable results in combining Apple Data Detectors and the prior art references and systems in Exhibit U, Table 7. For example, a POSITA would have recognized advantages and benefits to have Apple Data Detectors display the plurality of distinct instances of second information from search results to enable user selection of one of them for use in performing the action.
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	The Apple Data Detectors System discloses this element. The Address Book and the stored company names were associated with Claris Emailer and were available through the computer described for claim 1. *See* screenshots in claim 1. The Now Contact File ("Demo Contact File") and the stored company name, name, and address were associated with Now Contact and were available through the computer described for claim 1. *See* screenshots in claim 1. See also Nardi at 101: "Apple Data Detectors assumes a world of heterogenous data (in the user's machine) that comes from different applications in different file formats. For example, it makes sense to put an address in an address book, whether the address is from a message sent in any of several mail programs, appears in a downloaded document, or is in another address book maintained by the user. Apple Data Detectors is a pervasive technology, giving users access to actions appropriate for data in an entire set of documents." See also Nardi at fig. 4: "This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter. See also Nardi at 102: "We also see evidence that end users with varying degrees of programming skill are extending actions much as we

Exhibit E

had expected. One user—a skilled programmer—used the basic detectors and actions that ship with the product as the basis for a new detector/action pair for a personally relevant task. He wanted to look up software bug reports in a database based on the ID numbers of the reports commonly found in other bug reports, email messages, and other program management documents. He distributed this detector/action throughout his work group, and a colleague—a marketing manager with much less programming experience than the original programmer—was able to adapt the action so the detector could run on his laptop computer (where his email and other documents reside) and display the bug reports on his desktop machine (where is program management tools reside). This extension required changes to only a few lines of code in the action, something well within his technical ability."

See also Nardi at p. 99 ("Figure 4. An action script, demonstrating the generality of Apple Data Detectors' use of a scripting language and external applications as information repositories and as end-user tools. This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a "personal information manager" (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter.").

Now Contact was used to access the address book on the computer. *See* Video at 1:22-1:40.

See also MacWorld Expo 1997 at 1:11:00:

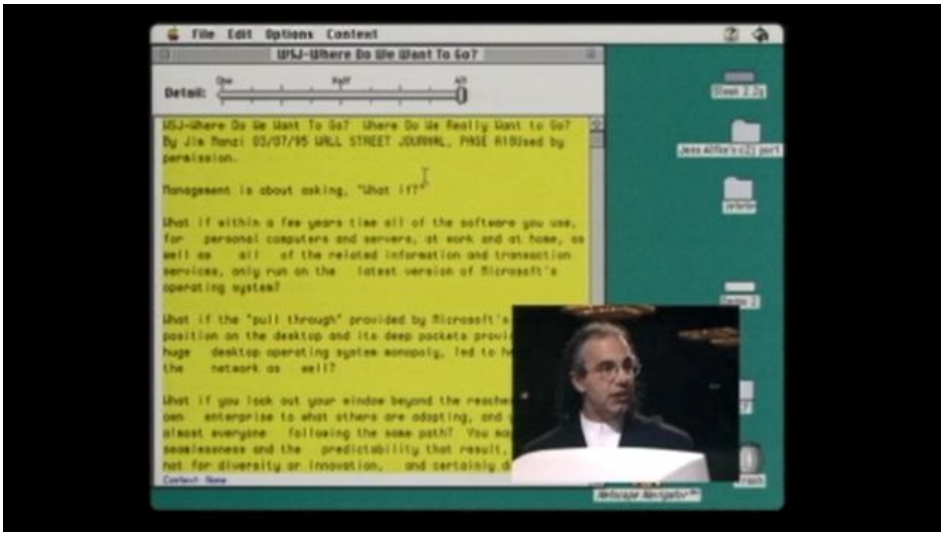


Exhibit E

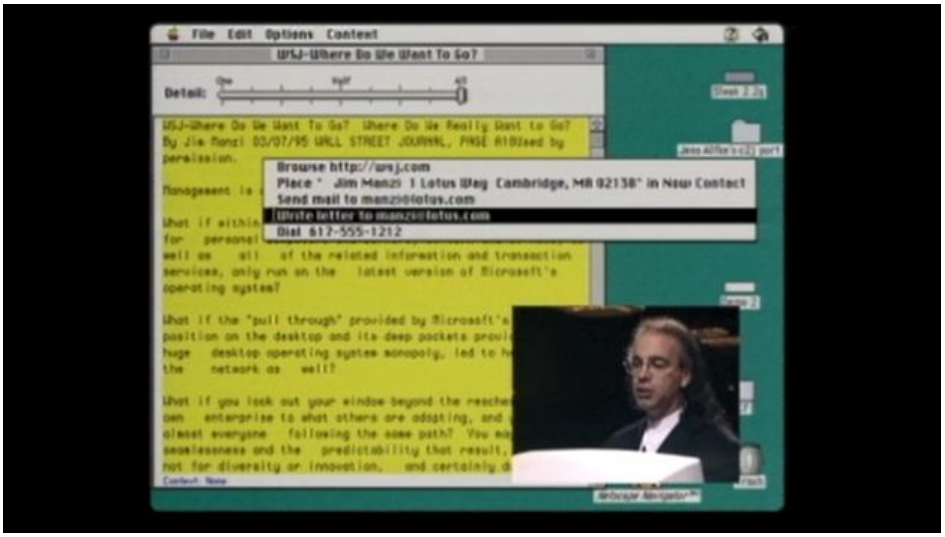
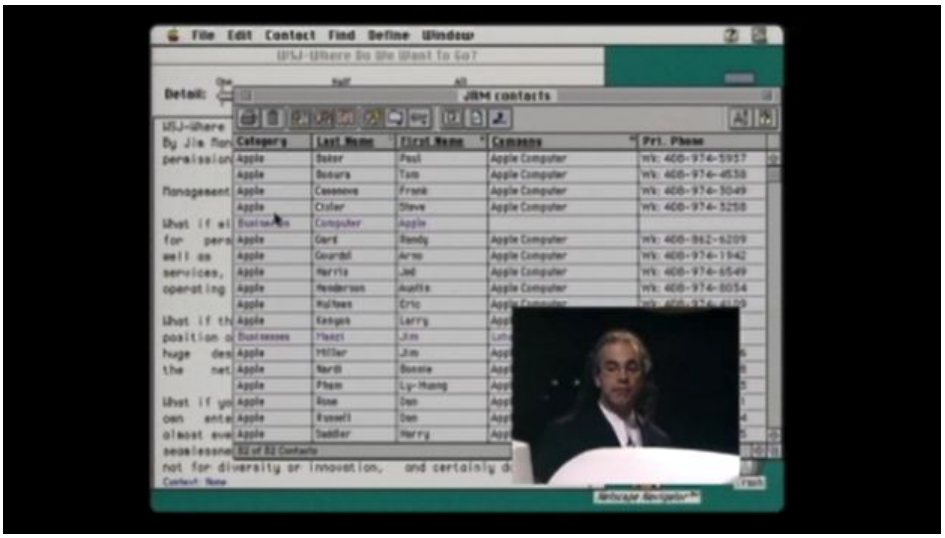
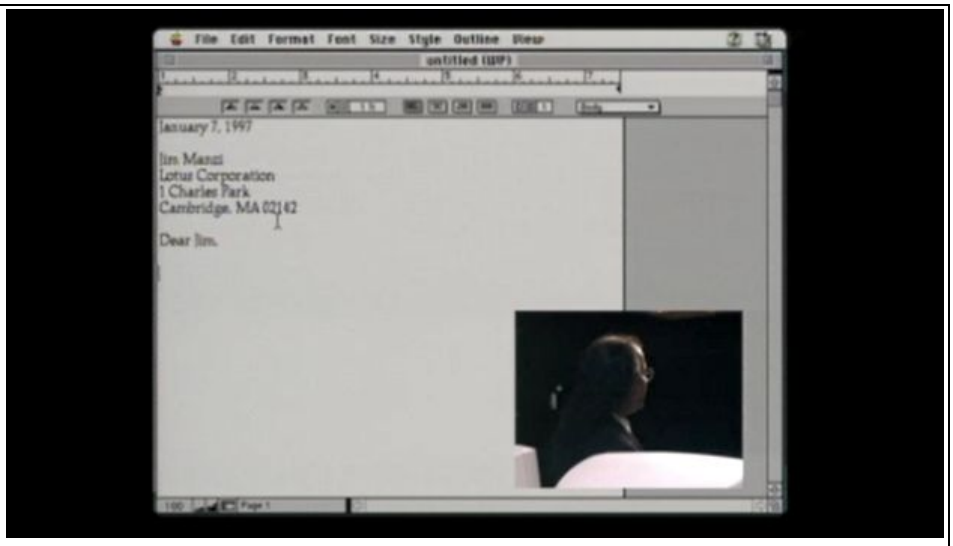
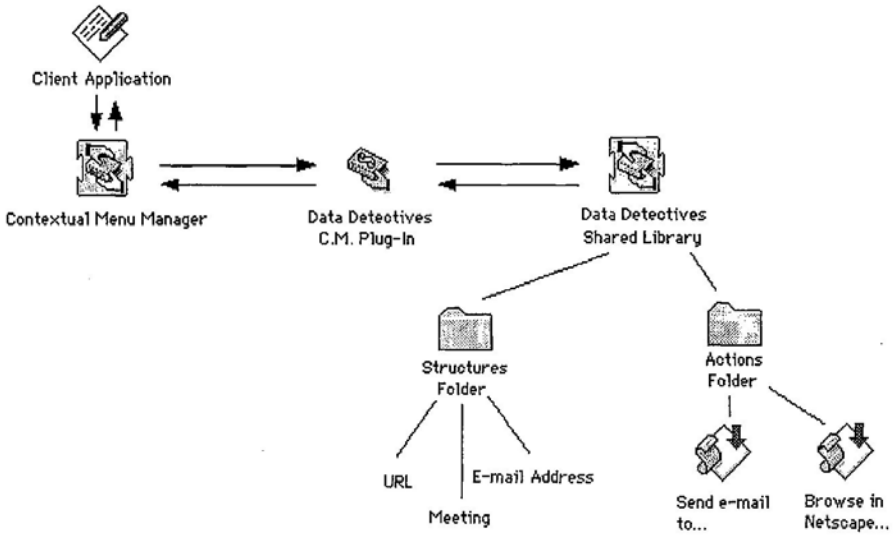
	 File Edit Options Context WSJ-Where Do We Want To Go? Detail: Browse http://wsj.com Send mail to manzi@lotus.com Dial 617-555-1212  File Edit Contact Find Define Window WSJ-Where Do We Want To Go? Detail: JRM contacts Category Last Name First Name Company Pri. Phone

Exhibit E



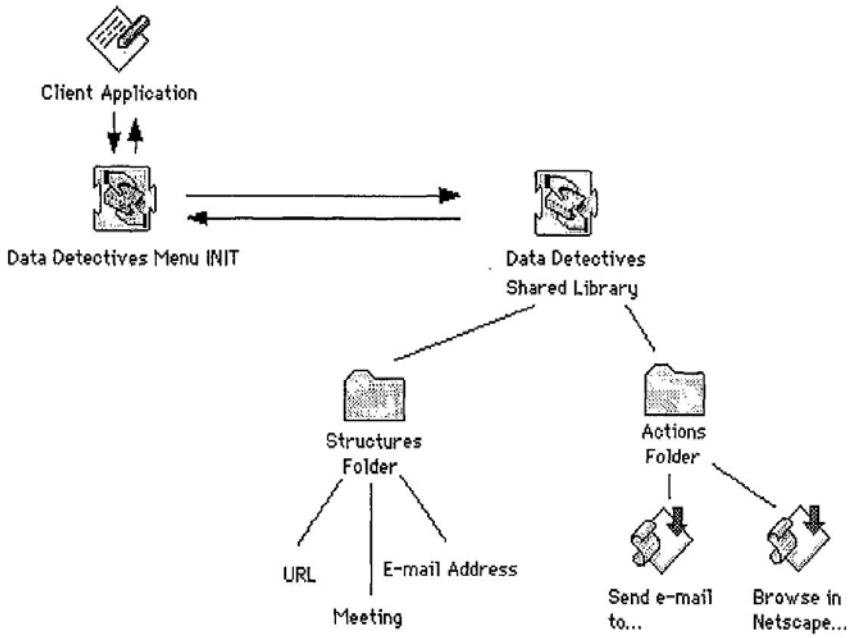
See WWDC Presentation at 4-5:

[Technical Overview - with Contextual Menus in Copland:]



[Technical Overview - Current]

Exhibit E



"**on** addressLetterTo(emailAddress)
set emailAddress **to** removeTextStyling(emailAddress)
tell application "Now Contact 3.5"
launch
activate
run
--find the person in the Now Contact database
try
set thePerson **to the first** person **whose** (customer 2 is emailAddress)
on error
beep
display dialog "Sorry, I don't have any information about " & emailAddress & "."
return
end try

--get the address information for the person
try
set firstAndLastName **to** (**the** first name **of** thePerson) & " " & (**the** last name **of** thePerson)
set theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (the work address **of** thePerson) & return & (**the** work city **of** thePerson)
& ", " & (**the** work state **of** thePerson) & " " & (**the** work zip of thePerson) & return
on error

Exhibit E

beep
display dialog “Sorry, I couldn’t get the address of “ & emailAddress 7 “.”
return
end try

--write the address information into ClarisWorks”
See Write a Letter AppleScript Code.

"As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250."

Miller at 5:38-47.

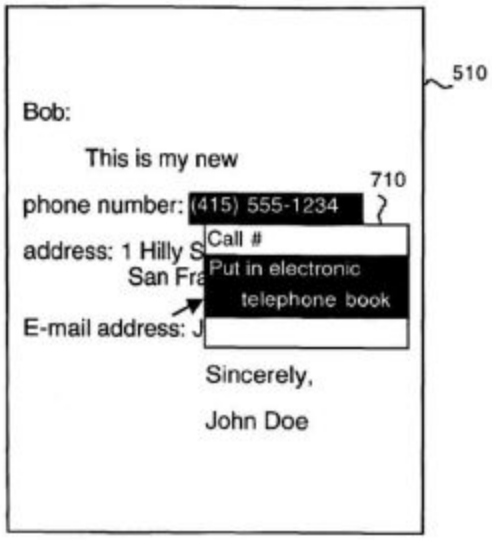


FIG. 7

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the

Exhibit E

	telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). The Apple Data Detectors' architecture separates detectors and actions so that more than one action can exist for any detector (without having to duplicate the detector for each action). Hence, a detector written by one person to support one task can be used by another detector for another task. Detectors are thus reusable and easily shared. (Nardi, page 100). "The phrase 'your application' refers to the applications that the attendees of WWDC were building. And the point that we wanted to make was that what we were building could provide a way for users to be able to very easily interact with those applications without having to go through many of the details of pulling down menus off of the menu bar and clicking on things and repositioning the cursor and typing stuff into text fields and clicking submit buttons and all of that. They could just interact directly with the content and then let the software go through all of the tedious parts of actually carrying out actions on that information. Q. And when you say 'input data more easily,' do you recall what that was referring to in your presentation? A. Well, that if we could identify, say, a very long email address and provide a way for someone to act directly on it, then the user would not

Exhibit E

	have to open up a text box and remember and type in that email address, possibly making mistakes along the way. Q. So what sorts of things could be done with the email address like that using the Apple Data Detectives that you're describing in this presentation? A. Well, in particular you could create a new outgoing email message to that email address. You could also put it into an address book so that you could easily use it later. Q. How would that work? Using the Apple Data Detectors or Detectives as they are called in this presentation, how would you actually take an email address in a document and put it into an address book? A. There were application programmatic interfaces, APIs, that would receive requests from outside parts of the computer system. And so the way that this would ordinarily work is that our part of the system would identify this and then send a message through this API channel over to the application and say, 'Make a new email message addressed to this person.' Q. How about to save it in their address book, how would that work? A. There would be another kind of message that could be sent to the application. This one would be 'save this email address inside of your address book.' Q And did you actually build that kind of functionality into the Apple Data Detectors? A. Yes. We would work with applications that could accept those messages and then we would write the bits of code that would take the discovered pieces of information and send them to the application." Miller Depo. at 42:23-45:7. "Q. And again, what were you trying to accomplish with this presentation to developers? A. Well, this would point out to developers that their applications could be accessed directly from user information. As opposed to the user having to go back into the system finder and open up a sequence of folders until they finally got to the user's application and then launch that and then do activities inside of the application." Miller Depo. at 49:15-24. "Q. What about a word processing program, could Apple Data Detectors be used in conjunction with a word processing program at the time? A Yes. Q. What was an example of a word processing program available on Macintosh that Apple Data Detectors could work in conjunction with? A. ClarisWorks is one." Miller Depo. at 74:5-14. "Q When it says that "Apple Data Detectors analyze the text you've

Exhibit E

	selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16.

Exhibit E

	"It was possible to add an address to some contact managers' address book. I believe it was Now Contact." Miller Depo. at 84:18-20. "A more interesting version of the – of that small piece of code might first look to see if that address existed in the address book and if it did, then it could offer, this already is here, would you like to overwrite it with the new information or not?" Miller Depo. at 86:13-18. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support.

Exhibit E

	Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. Why is it that you separated the apple Data Detector from the application? MR. WILLIAMS: Objection to form. THE WITNESS: We would not be able to build Apple Data Detectors into all of the applications that were out there. Those were other companies' applications with their own proprietary interests and their work. BY MR. UNIKEL: Q. And so why create it as a separate piece of code from the application itself? A. So that those applications, by making use of the Data Detector technologies, could still get access to the functionality." Miller Depo. at 148:6-20. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,'

Exhibit E

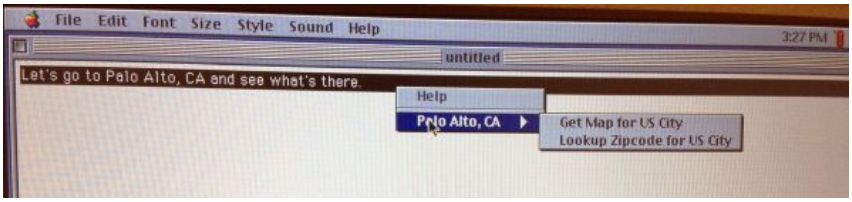
	unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that

Exhibit E

	I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "[W]hy did you make LiveDoc its own separate piece of code where it had to exchange information with the application rather than just making it a part of every application? A. I think for much the same reason – I mean, this is going back quite a ways. But I think it was the same sort of logic as for making the Data Detector engine a separate thing. That by keeping the LiveDoc manager as an external component, we would be more able to connect into potentially more applications. We wouldn't be building LiveDoc capability into all of those applications. All we would – all we would need would be for that application to be able to tell the LiveDoc manager where on the screen are these characters." Miller Depo. at 180:11-181:3. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosure. When IAD detects an email address and a user selects "Send mail with Claris Emailer," Claris Emailer will search its information source for a company name associated with the domain of the email address and will insert this into the Outgoing Message panel for a new email. For example:

Exhibit E

When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could obtain a zip code for the selected US city, for which the system would search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:



This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In

Exhibit E

	addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information." *See also* Nardi at fig. 4: "This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date and salutation information. This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address and salutation into it, leaving the user ready to write the letter." *See also* Nardi at 99: "While Apple and third-party developers will provide many detectors and actions, it is clear that enabling end users to write their own detectors and actions will make Apple Data Detectors much more powerful and useful by providing domain-specific programming capabilities appropriate for the specific needs of specific users [8]." *See also* Nardi at 101: "Unlike systems based on predefined recognizer/action pairs, such as the Selection Recognition Agent [10], the Apple Data Detectors' scripting ability allows any set of arbitrary actions to be executed when the user selects a particular action (see Figure 3). Scripting is not limited to the parameters of a command line interface to the application; it can do anything that can be expressed in the scripting language, including manipulation of data structures inside the application, if the application's scripting model makes that possible." *See also* Nardi at 101: "The ability to work within existing documents provides immediate user value and leverages the data that the user is already using."

Exhibit E

	See also Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." *See also* Nardi at 102: "We also see evidence that end users with varying degrees of programming skill are extending actions much as we had expected. One user—a skilled programmer—used the basic detectors and actions that ship with the product as the basis for a new detector/action pair for a personally relevant task. He wanted to look up software bug reports in a database based on the ID numbers of the reports commonly found in other bug reports, email messages, and other program management documents. He distributed this detector/action throughout his work group, and a colleague—a marketing manager with much less programming experience than the original programmer—was able to adapt the action so the detector could run on his laptop computer (where his email and other documents reside) and display the bug reports on his desktop machine (where is program management tools reside). This extension required changes to only a few lines of code in the action, something well within his technical ability." The system could detect an email address, and in response to the "Write letter" command, obtain the mailing address for that person from Now Contact, and address a letter to that person and mailing address in Claris Works. See screenshot above, Video at 1:22-1:40. " *--write the address information into ClarisWorks* **tell**application "ClarisWorks" **try** launch activate *--run --don't do a "run here, or you'll get the "New document" dialog…* open (path to system folder as string) & "SD letter template" as "WP" **on error** beep activate display dialog "Sorry, I couldn't open the letter template: 'SD letter template' in the System Folder."

Exhibit E

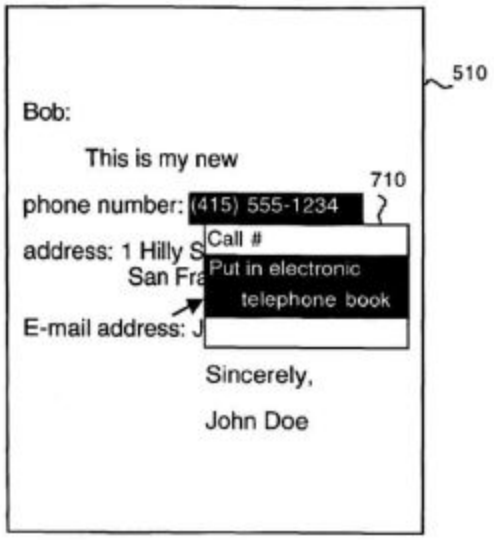
	See Write a Letter AppleScript Code. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." Miller at 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to

Exhibit E

	the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector.

Exhibit E

	(Nardi, pages 99-100). "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that

Exhibit E

	allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number.

Exhibit E

	Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,' unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition

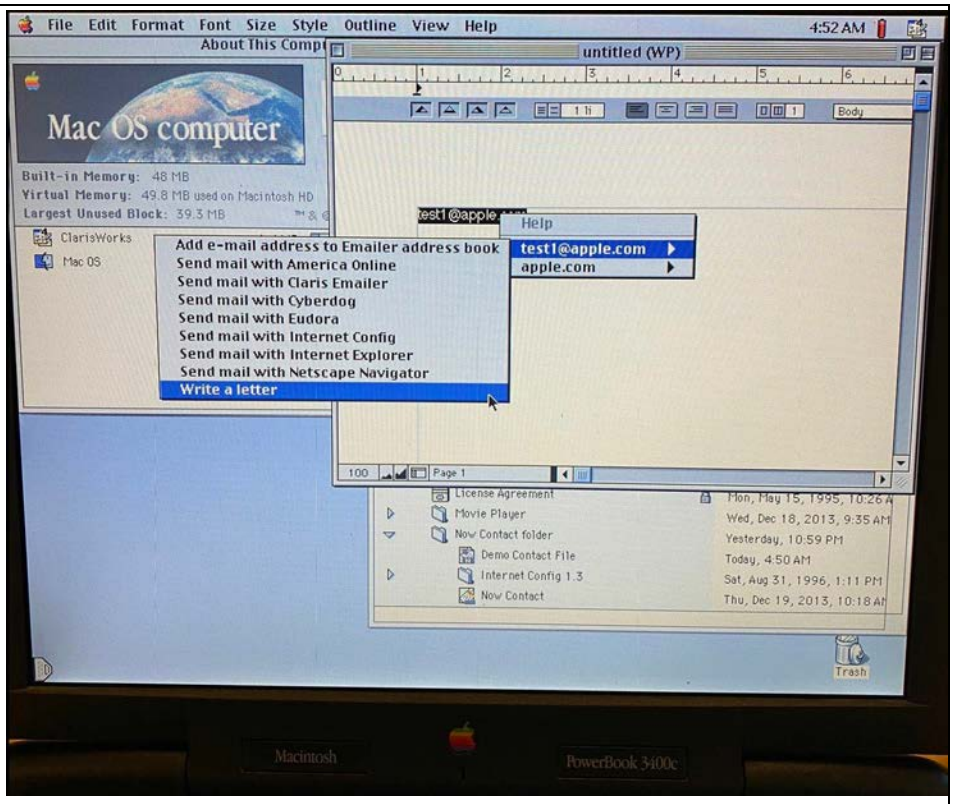
Exhibit E

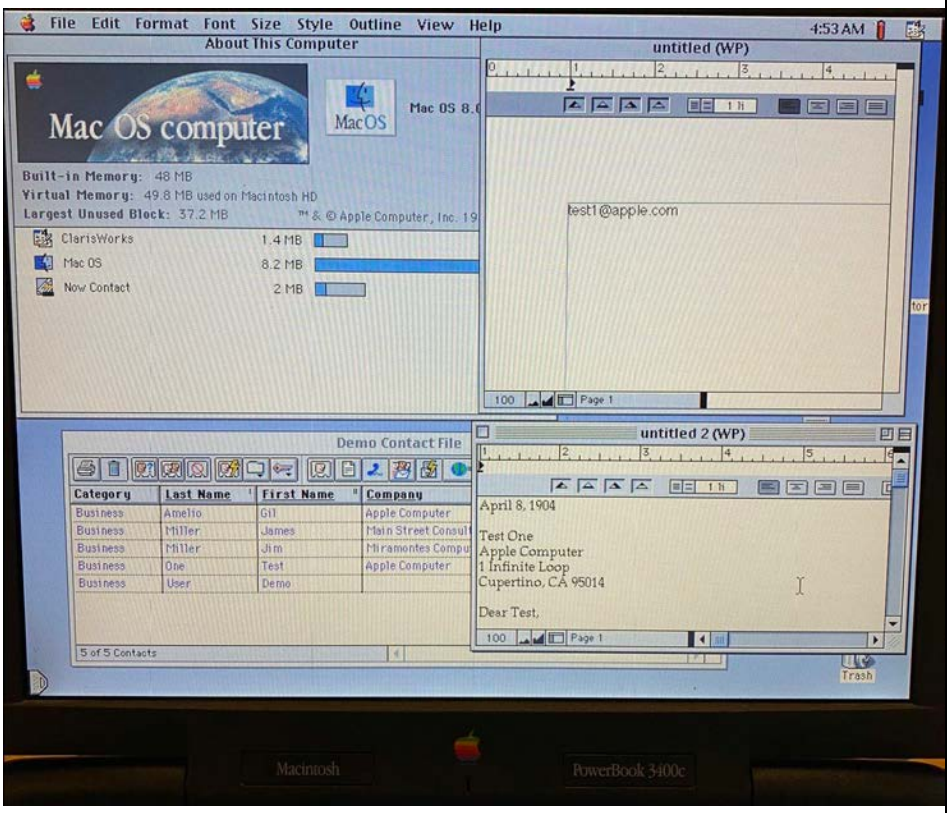
	component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted.

Exhibit E

	You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Claris Works application's "untitled (WP)" window, and the user selects the associated action of "Write a letter" from the sub-menu, the Now Contact application is launched, and an "untitled 2 (WP)" window is presented in the Claris Works application. In this example, the first name ("Test"), last name ("One"), and physical address information ("Apple Computer 1 Infinite Loop Cupertino, CA 95014") are associated with "test1@apple.com" in the Now Contact Demo Contact File, and displayed in the "untitled 2 (WP)" window.

Exhibit E





For example, during my Inspection of the Powerbook 3400C, I located

Exhibit E

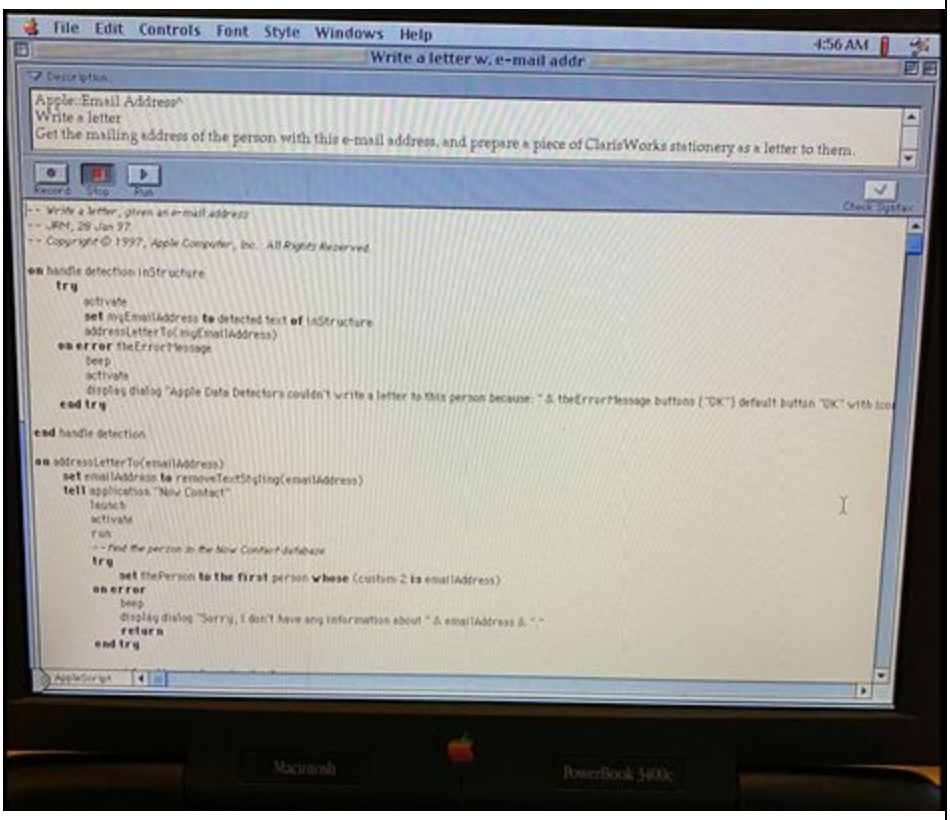
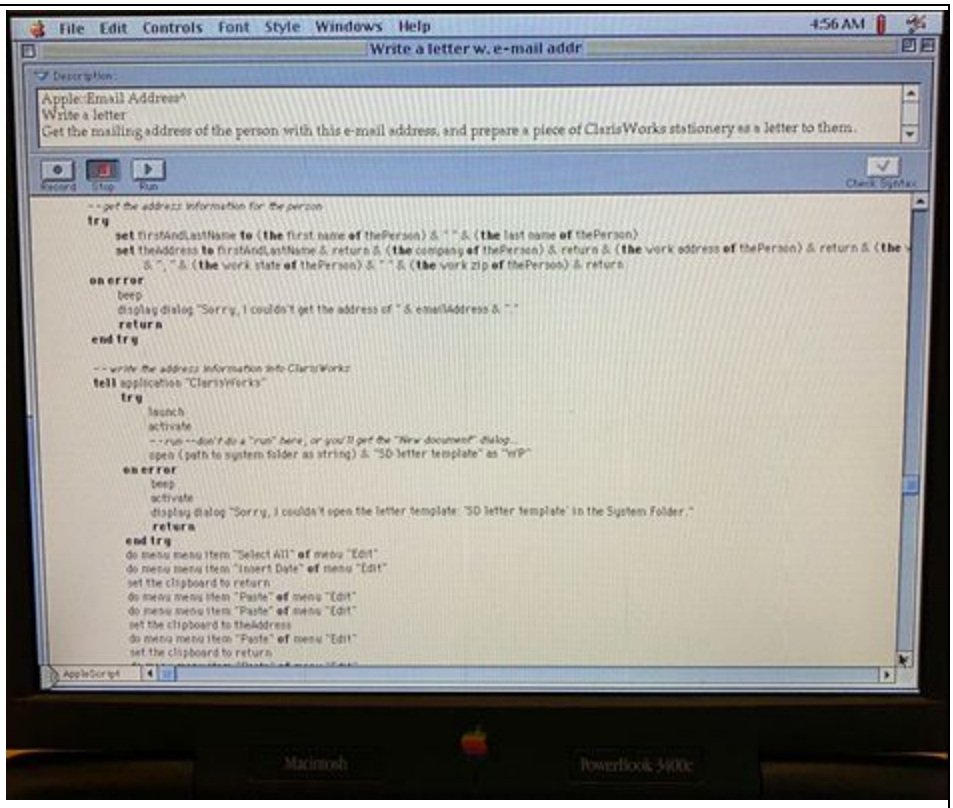
	and observed the written code associated with the "Write a letter" action. After having reviewed the code associated with the "Write a letter" action, it is my opinion that it would have taken a POSITA no more than an hour to modify that code to insert the information contained in the "untitled 2 (WP)" window back into the "untitled (WP)" window as part of the "Write a letter" action in the existing Apple Data Detectors System. 

Exhibit E


File Edit Controls Font Style Windows Help
4:56 AM
Write a letter w. e-mail addr
Apple::Email Address^
Write a letter
Get the mailing address of the person with this e-mail address, and prepare a piece of ClarisWorks stationery as a letter to them.
Record
Stop
Run
Check Syntax
AppleScript
Macintosh
PowerBook 3400c

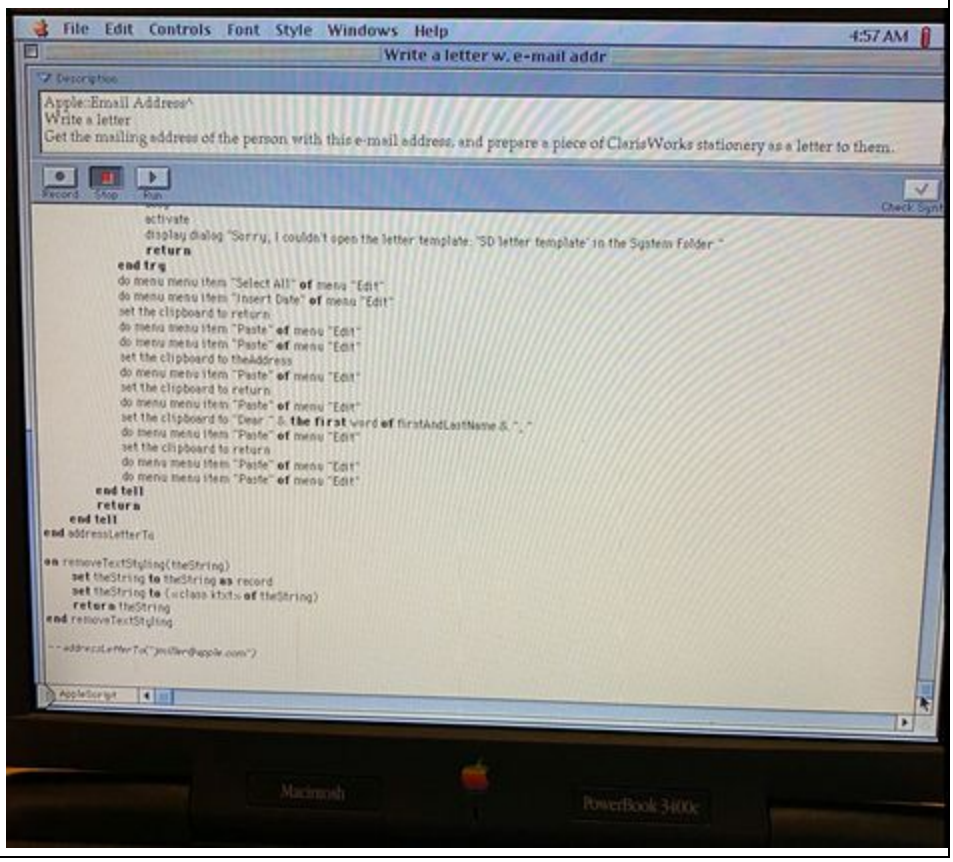

File Edit Controls Font Style Windows Help
4:57 AM
Write a letter w. e-mail addr
Apple::Email Address^
Write a letter
Get the mailing address of the person with this e-mail address, and prepare a piece of ClarisWorks stationery as a letter to them.
Record
Stop
Run
Check Syntax
AppleScript
Macintosh
PowerBook 3400c

Exhibit E

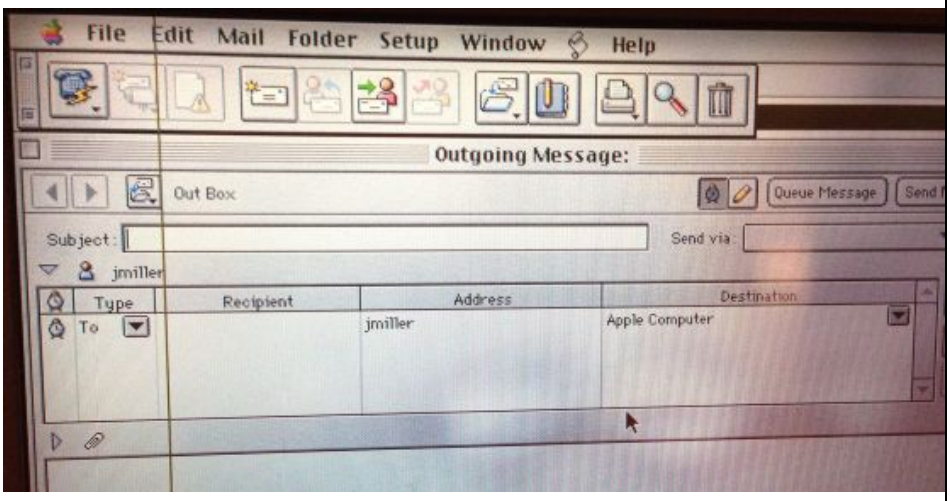
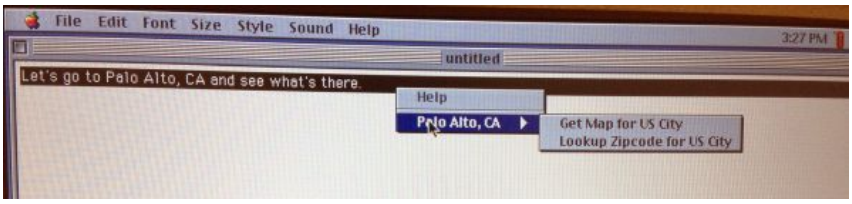
	For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining Apple Data Detectors and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have Apple Data Detectors cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures. When IAD detects an email address and a user selects "Send mail with Claris Emailer," Claris Emailer will search its information source for a company name associated with the domain of the email address and will insert this into the Outgoing Message panel for a new email. For example:  When a user selects text and right-clicks in Simple Text, IAD can detect occurrences of a city name followed by a state name or US Postal abbreviation and occurrences of the name of a state, territory, or protectorate of the United States and provide options. The user could then obtain a map of the selected US city, for which the system would search the Yahoo map website for a map of the city, or the user could

Exhibit E

obtain a zip code for the selected US city, for which the system would search the US Postal Service website for a list of zip codes for the city. *See* US Geographic Detectors 1.0 Read Me file. For example:



This feature was available in a Claris Emailer email as well, and starting with Claris Emailer version 2.0v3, a user needed to only right-click (and not select text) to obtain a list of actions for geographic information. (*See* Second Miller Affidavit, at paras. 7, 26, 48.)

See also Nardi at 98: "Our first step was to find a user problem that needed solving in which intelligent agents would add value. In an investigation of how people file information on their computer desktops [1], we discovered that a common user complaint is that they cannot easily take action on the structured information found in everyday documents (structured information being data-recognizable by a grammar). Ordinary documents are full of such structured information: phone numbers, fax numbers, street addresses, email addresses, email signatures, abstracts, tables of contents, lists of references, tables, figures, captions, meeting announcements, Web addresses, and more. In addition, there are countless domain-specific structures . . . . These structures are not only relevant to users, but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts.

Apple Data Detectors supports a wide range of uses. Think of all the structured information in the documents you work with; in addition to those mentioned already, add bibliography items, forms (such as travel expensive reports and non-disclosure agreements), executive summaries, and most important, such domain-specific kinds of data as legal boilerplate, customer orders, and library search requests. Specific detectors can be created for each of these types of information."

See also Nardi at fig. 4: "This script can be activated when the system detects a telephone number. It then generates word processor letterhead addressed to the person possessing that number, with appropriate date

Exhibit E

	and salutation information. This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address and salutation into it, leaving the user ready to write the letter." *See also* Nardi at 99: "While Apple and third-party developers will provide many detectors and actions, it is clear that enabling end users to write their own detectors and actions will make Apple Data Detectors much more powerful and useful by providing domain-specific programming capabilities appropriate for the specific needs of specific users [8]." *See also* Nardi at 101: "Unlike systems based on predefined recognizer/action pairs, such as the Selection Recognition Agent [10], the Apple Data Detectors' scripting ability allows any set of arbitrary actions to be executed when the user selects a particular action (see Figure 3). Scripting is not limited to the parameters of a command line interface to the application; it can do anything that can be expressed in the scripting language, including manipulation of data structures inside the application, if the application's scripting model makes that possible." *See also* Nardi at 101: "The ability to work within existing documents provides immediate user value and leverages the data that the user is already using." *See also* Nardi at 101: "Having to choose a particular action is actually an artifact of Apple Data Detectors' ability to find more than one structure in a selection and the need to offer more than one action for the same structure (such as 'open URL' and 'place URL in hot list'). But multiple structures and actions are of benefit to users in terms of their being able to choose the structure to be manipulated and to choose the action to be employed. Users therefore remain in control of their work with the computer at all times." *See also* Nardi at 102: "We also see evidence that end users with varying degrees of programming skill are extending actions much as we had expected. One user—a skilled programmer—used the basic detectors and actions that ship with the product as the basis for a new detector/action pair for a personally relevant task. He wanted to look up software bug reports in a database based on the ID numbers of the reports commonly found in other bug reports, email messages, and other program management documents. He distributed this detector/action throughout his work group, and a colleague—a marketing manager with much less programming experience than the original programmer—was

Exhibit E

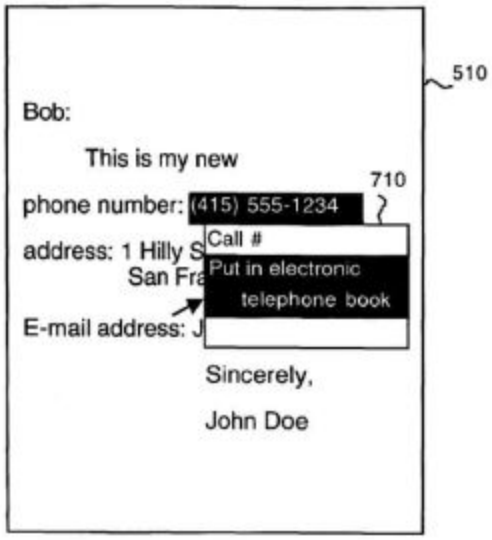
	able to adapt the action so the detector could run on his laptop computer (where his email and other documents reside) and display the bug reports on his desktop machine (where is program management tools reside). This extension required changes to only a few lines of code in the action, something well within his technical ability." "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." Miller at 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library

Exhibit E

	420 containing important names. When analyzer server 220 identifies an address using the .e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a

Exhibit E

	more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100). " *--write the address information into ClarisWorks* **tell**application "ClarisWorks" **try** launch activate *--run --don't do a "run here, or you'll get the "New document" dialog…* open (path to system folder as string) & "SD letter template" as "WP" **on error** beep activate display dialog "Sorry, I couldn't open the letter template: 'SD letter template' in the System Folder." *See* Write a Letter AppleScript Code. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number

Exhibit E

	was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "Well, you are sitting in this email program and you find that while you're working with the email program, you want to make a phone call. So here you can make that phone call and you've never left the email program. You're still in it as opposed to having to leave the program, go out into the finder, search through folders trying to find your telephone application, finally find it, launch it, find the part of the application that allows you to specify where you can enter a phone number and, you know, finally end up making your phone call." Miller Depo. at 78:8-16. "I see there is, in fact, a pop-up menu with a variety of options that are shown. Do you see that? A. Yes. Q. What are those options – what are those? What's being shown in that pop-up box? A. Those are actions that can be applied to various things that have been found in that document. Q. For example, the third one down says, "Address new email to Kawasaki@applelink.apple.com in Claris Emailer." Do you see that? A. Yes. Q. so it that an option a user could choose? A. Yes. Q. And what would happen if the user clicked on that option? A. Claris Emailer would be launched and a new email would be created and it would be given the Kawasaki@applelink.apple.com in the to field of the new email." Miller Depo. at 98:24-99:21. "So to your recollection, could Apple Data Detectors actually pass information in 1996 to Eudora email program? A. Yes." Miller Depo. at 100:12-15.

Exhibit E

	"Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And, in fact, two sentences down you say, quote, 'These structures are not only relevant to users but because of their structure, are also recognizable by parsing technologies. Once identified, the structure's type can be used to identify appropriate actions that might be carried out like placing a meeting on a schedule, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number and compiling a list of abstracts,'

Exhibit E

	unquote. Do you see that? A. Yes. * * * Q. And so would Apple Data Detectors be able to determine the type of information in addition to just detecting it just existed in a document? A. Yes. Part of the recognition process is saying that this is a thing of a certain type like an email address. Q. And once that determination of the type is made, what then does the Apple Data Detectors do with that determination? A. That determines what actions could be applied to it." Miller Depo. at 150:24-152:16. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25. "Q. And so if I'm understanding correctly, if a phone number was detected, then the Apple Data Detectors would allow that phone number to be looked up in a contact database, a mailing address retrieved and then the mailing address inserted into a word processing document? MS. FRANKLIN: Objection to form. THE WITNESS: Yes." Miller Depo. at 156:2-9. "Is it possible to write a script for the action that would instead of inserting that mailing address information into a new word processing document, to insert it into the open word processing email document that

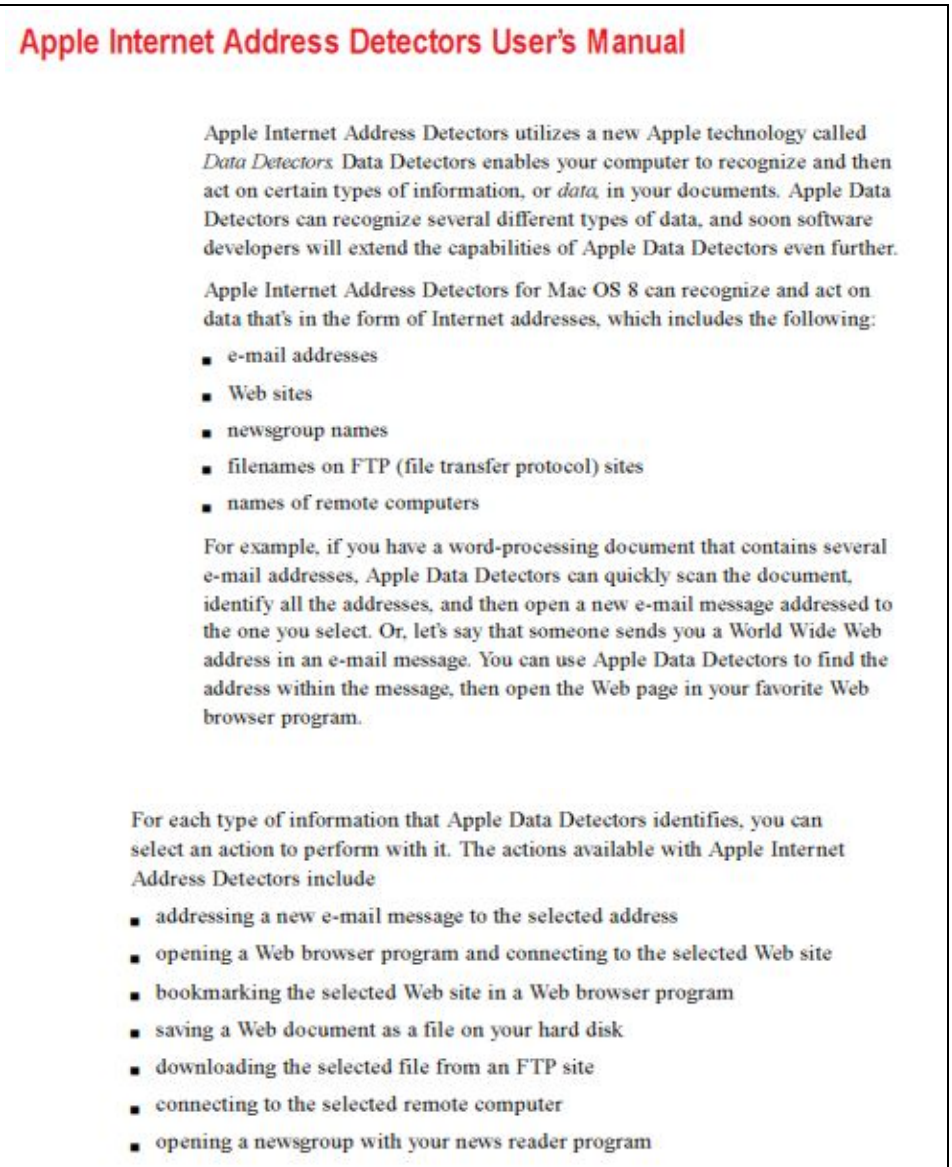
Exhibit E

	I started with? A. I believe so. It depends on the capabilities offered by the word processing application. But if the word processing application allowed you to insert text at some arbitrary place in it, I believe it would be able to. Q. And am I correct that that would be a relatively simple program to write? A. I believe so. Q. Approximately how long do you think it would take to write that program assuming that the word processing program itself would allow for insertion? A. Yes. Half hour." Miller Depo. at 157:18-158:12. "There would be an application showing you the contents of the document with some text in it. And so you would, with your mouse, drag out a region of that – of a document. You know, put the mouse cursor down, click the button, hold it down, drag it and you get the visual highlighting. And then you can let go and the section remains highlighted. You would then hold down either the right button of a mouse, if you had the right kind of mouse, or hold down the control key and press the mouse button with the mouse cursor over the region of selected text. And that would display the hierarchal pop-up menu. It would be a list of the structures that Data Detectors had found, the email addresses, the phone numbers, things like that. And if you then with your mouse pointed at one of those structures, a new menu would be drawn off to the side showing the actions that could be carried out on that detected thing. If you picked one of those, a bit of code would run that would, for instance, launch an email program and open up a new email message addressed to that email message that had been found." Miller Depo. at 184:23-185:25. For example, as observed during my Inspection of the Powerbook 3400C, when an email address is detected in the Claris Works application's "untitled (WP)" window, and the user selects the associated action of "Write a letter" from the sub-menu, the Now Contact application is launched, and an "untitled 2 (WP)" window is presented in the Claris Works application. In this example, the first name ("Test"), last name ("One"), and physical address information ("Apple Computer 1 Infinite Loop Cupertino, CA 95014") is associated with "test1@apple.com" in the Now Contact Demo Contact File, and is displayed in the "untitled 2 (WP)" window. See screenshots in claim 18 above.

Exhibit E

	For example, during my Inspection of the Powerbook 3400C, I located and observed the written code associated with the "Write a letter" action. After having reviewed the code associated with the "Write a letter" action, it is my opinion that it would have taken a POSITA no more than an hour to modify that code to insert the information contained in the "untitled 2 (WP)" window back into the "untitled (WP)" window as part of the "Write a letter" action in the existing Apple Data Detectors System. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining Apple Data Detectors and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have Apple Data Detectors cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures. The User Manual states at pp. 1-2:

Exhibit E



Apple Internet Address Detectors User's Manual

Apple Internet Address Detectors utilizes a new Apple technology called *Data Detectors*. Data Detectors enables your computer to recognize and then act on certain types of information, or *data*, in your documents. Apple Data Detectors can recognize several different types of data, and soon software developers will extend the capabilities of Apple Data Detectors even further.

Apple Internet Address Detectors for Mac OS 8 can recognize and act on data that's in the form of Internet addresses, which includes the following:

- e-mail addresses
- Web sites
- newsgroup names
- filenames on FTP (file transfer protocol) sites
- names of remote computers

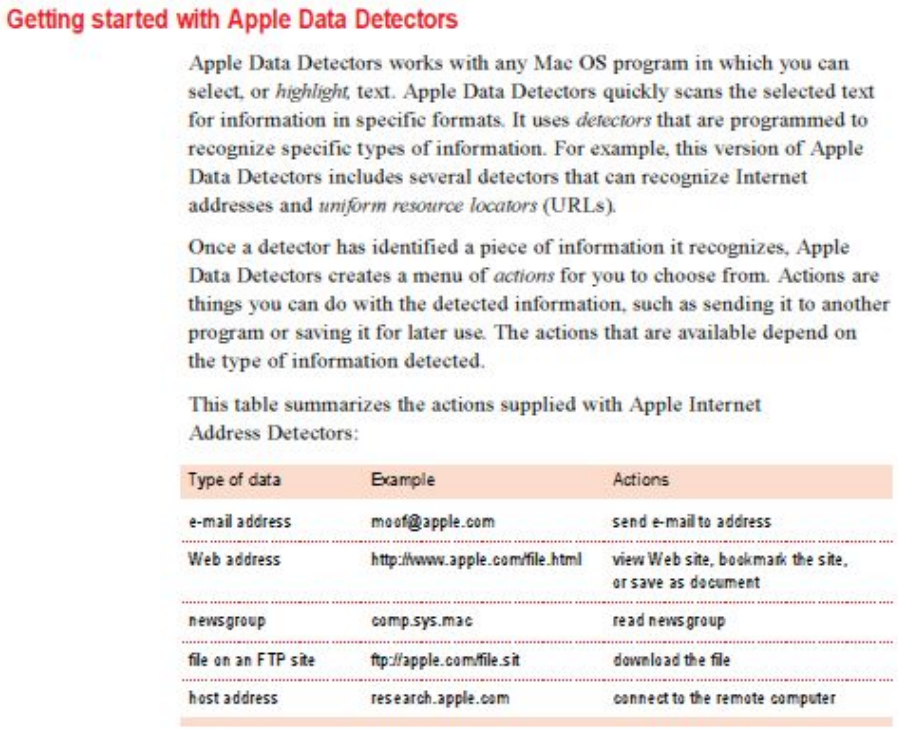
For example, if you have a word-processing document that contains several e-mail addresses, Apple Data Detectors can quickly scan the document, identify all the addresses, and then open a new e-mail message addressed to the one you select. Or, let's say that someone sends you a World Wide Web address in an e-mail message. You can use Apple Data Detectors to find the address within the message, then open the Web page in your favorite Web browser program.

For each type of information that Apple Data Detectors identifies, you can select an action to perform with it. The actions available with Apple Internet Address Detectors include

- addressing a new e-mail message to the selected address
- opening a Web browser program and connecting to the selected Web site
- bookmarking the selected Web site in a Web browser program
- saving a Web document as a file on your hard disk
- downloading the selected file from an FTP site
- connecting to the selected remote computer
- opening a newsgroup with your news reader program

The User Manual states at p. 4:

Exhibit E



Getting started with Apple Data Detectors

Apple Data Detectors works with any Mac OS program in which you can select, or *highlight*, text. Apple Data Detectors quickly scans the selected text for information in specific formats. It uses *detectors* that are programmed to recognize specific types of information. For example, this version of Apple Data Detectors includes several detectors that can recognize Internet addresses and *uniform resource locators* (URLs).

Once a detector has identified a piece of information it recognizes, Apple Data Detectors creates a menu of *actions* for you to choose from. Actions are things you can do with the detected information, such as sending it to another program or saving it for later use. The actions that are available depend on the type of information detected.

This table summarizes the actions supplied with Apple Internet Address Detectors:

Type of data	Example	Actions
e-mail address	moof@apple.com	send e-mail to address
Web address	http://www.apple.com/file.html	view Web site, bookmark the site, or save as document
newsgroup	comp.sys.mac	read newsgroup
file on an FTP site	ftp://apple.com/file.sit	download the file
host address	research.apple.com	connect to the remote computer

The User Manual states at pp. 5-6:

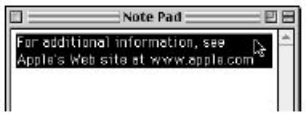
Exhibit E

Using Apple Data Detectors

To use Apple Data Detectors, follow these steps:

1 **Select some text in any application that allows you to highlight text.**

Make sure the selected text contains at least one type of data that Apple Data Detectors recognizes. (In this example, the selected text contains one complete Internet address.)



2 **Hold down the Control key, then press and hold down the mouse button.**

A "contextual menu" appears. It lists all the recognized data found in the selection.

Tip: If the contextual menu is empty, or the message "no structures found in selection" appears, the text you selected did not contain any recognizable data.

After the contextual menu appears, you can release the Control key. Be sure to keep holding down the mouse button or the menu will disappear.

3 **Choose a recognized data item from the contextual menu, then choose an action from the submenu that appears.**

IAD could operate on text entered by a user in a Simple Text file, a Claris Emailer email, or text entered using any other application.

See also Nardi at pp. 96-98 (including figs. 1, 2): "As Apple Computer researchers, we started from a simple but focused approach to agents: That they should have the ability to infer appropriate high-level goals from user actions and requests and take action to achieve these goals. Further, based on a study of reference librarians as exemplary human agents [9], we wanted to build a system in which the user would not have to state goals explicitly and in detail. We learned from librarians

Exhibit E

	that a large part of their value to clients is in working with imprecise requests. Beyond this concern, our general design strategy was to keep the most basic user question in front of us at all times: Will this software do something useful for users in an intelligent way that makes them more productive? The system we describe here—Apple Data Detectors—meets our criteria of being unobtrusive, being able to infer user needs, and doing useful work. Apple Data Detectors shipped as a product in 1997. Earlier work on intelligent agents was multifaceted, to the point where it is difficult to find a consensus among researchers on exactly what constitutes an 'agent' or even 'intelligence.' However, in nearly all cases, systems described as 'agent-based' rely on some explicitly represented knowledge about relevant aspects of the world—the objects or concepts being addressed by the software, the tasks relevant to the user, and the user's own knowledge about the world. Researchers have used machine learning techniques to track user actions and construct models of user preferences [7], create explicit models of user knowledge and skill levels in an attempt to anticipate user actions, misconceptions, and information needs [2], and implement planning systems to leap from a user's stated intention to the specific actions required to achieve that intention [3]. The locality of agents also varies across different agent-based systems; some act only within one's own machine, find others autonomously crawl the Web, searching for interesting content [4]. We tried to find a middle ground by using explicit representations of user-relevant information as a means of identifying actions users might wish to take but to leave the choice of these actions to users." *See also* Nardi at fig. 4: "This script uses two applications: First, a 'personal information manager' (Now Contact 3.5) is opened and used as a database. Then the script opens an empty word processor document (via Corel WordPerfect) and writes the date, name, address, and salutation into it, leaving the user ready to write the letter." *See also* Nardi at 103-04: "Apple Data Detectors is a first step toward extracting semantics from everyday documents without asking users to create documents in new ways. Such an intelligent agent redefines 'document' from a stream of characters to a data structure containing specific, known kinds of structures that can play specific, known roles in user interactions. Such an approach can provide a foundation for more powerful analyses beyond our current recognition and parsing technology. Future work could explore the use of more sophisticated kinds of recognition and parsing, including those that rely on finite state technology and linguistically informed context analysis [5], as well as integration with statistical techniques of data analysis, such as

Exhibit E

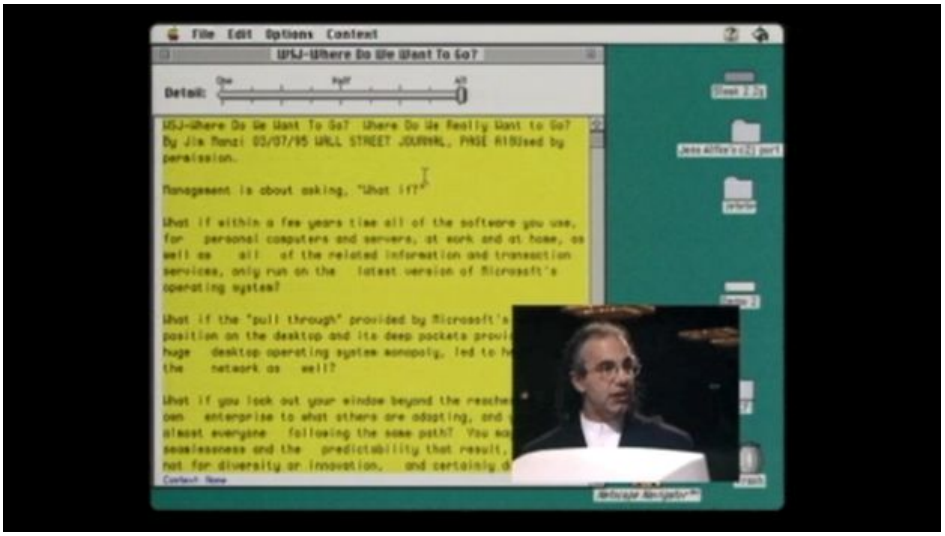
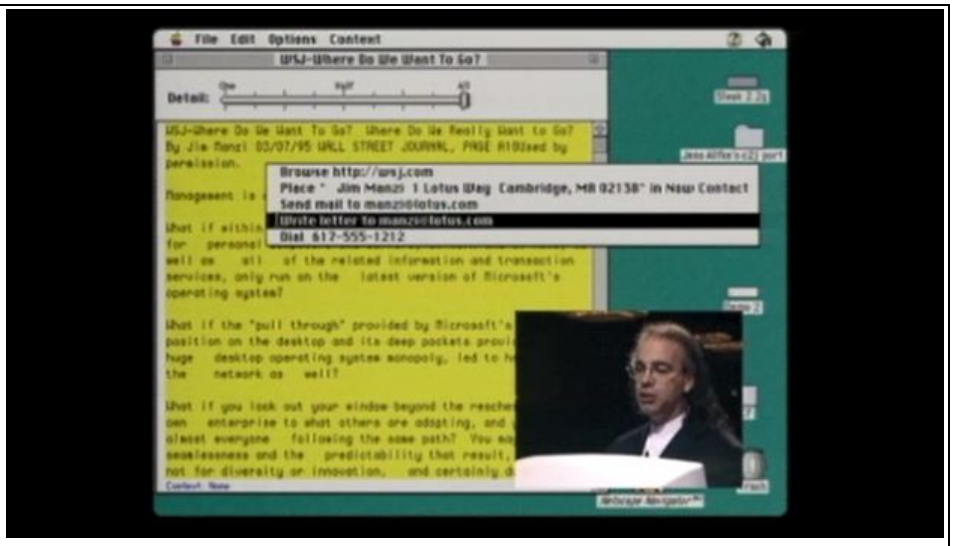
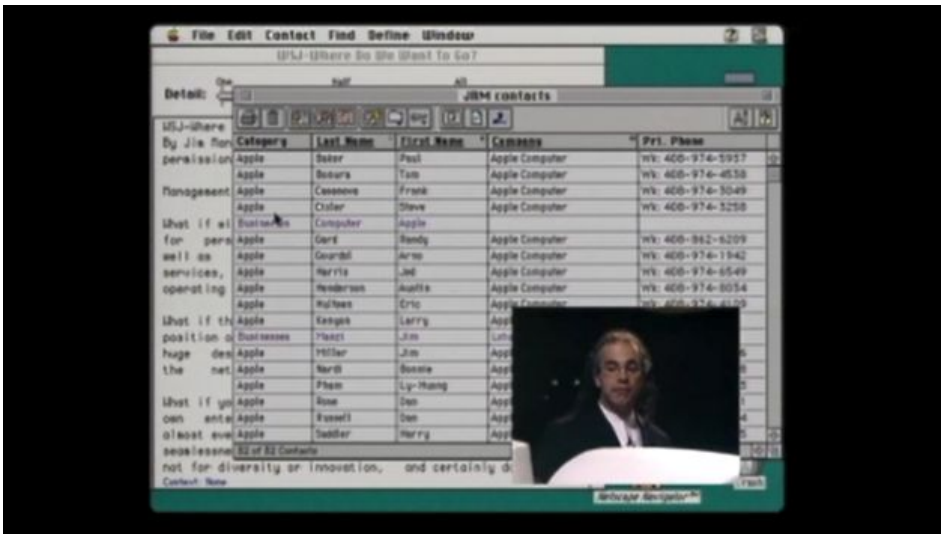
	relevance-based techniques. One important future step for research will be to build knowledge into the system about the structures being recognizing and how these structures are related to user goals and tasks. Doing so will provide the basis for more flexible and powerful task support. For instance, if we can attribute some reasonable set of email address semantics to the textual presentation of an email address in a document, the system can use the address as a pointer to the person with that address. We can then carry out system actions intended for people (such as 'Place a phone call to this person') on an email address and let the system figure out the person implied by the address. (This can already be done through Apple Data Detectors but requires writing a script, rather than relying on inferencing as suggested here.) Such interaction might require a different user interface from the one used today. One can certainly imagine many different kinds of user interfaces to the basic structure-detection technology underlying the current system." The system included a computer containing a medium (such as a hard disk drive or memory) containing the recited instructions. *See* Video at 0:30-1:40. *See also* MacWorld Expo 1997 at 1:11:00: 

Exhibit E


File Edit Options Content
WSJ-Where Do We Want To Go?
Detail:
Browse http://wsj.com
Place " Jim Manzi 1 Lotus Way Cambridge, MA 02138" in New Contact
Send mail to manzi@lotus.com
Write letter to manzi@lotus.com
Dial 617-555-1212


File Edit Contact Find Define Window
WSJ-Where Do We Want To Go?
Detail:
JBM contacts
Category
Last Name
First Name
Company
Pri. Phone

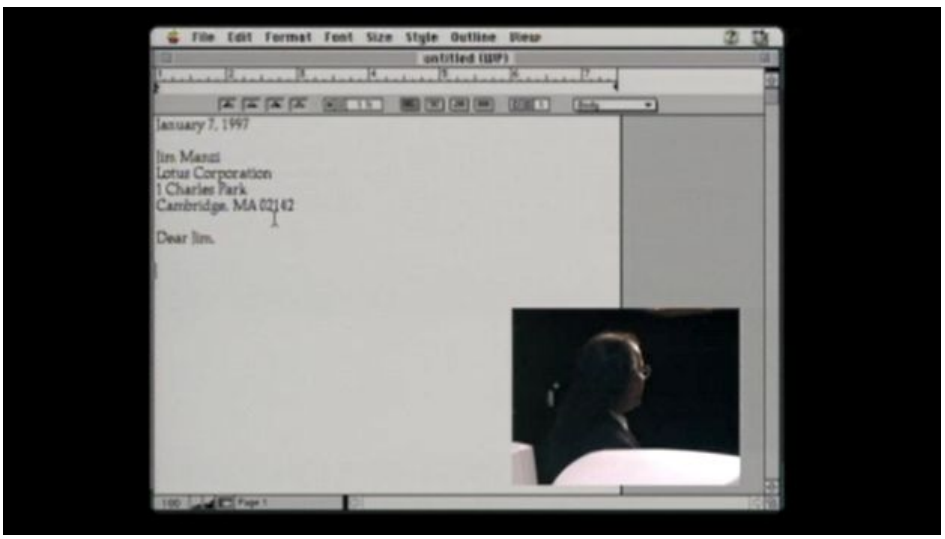

File Edit Format Font Size Style Outline View
untitled (WP)
January 7, 1997
Jim Manzi
Lotus Corporation
1 Charles Park
Cambridge, MA 02142
Dear Jim,

Exhibit E

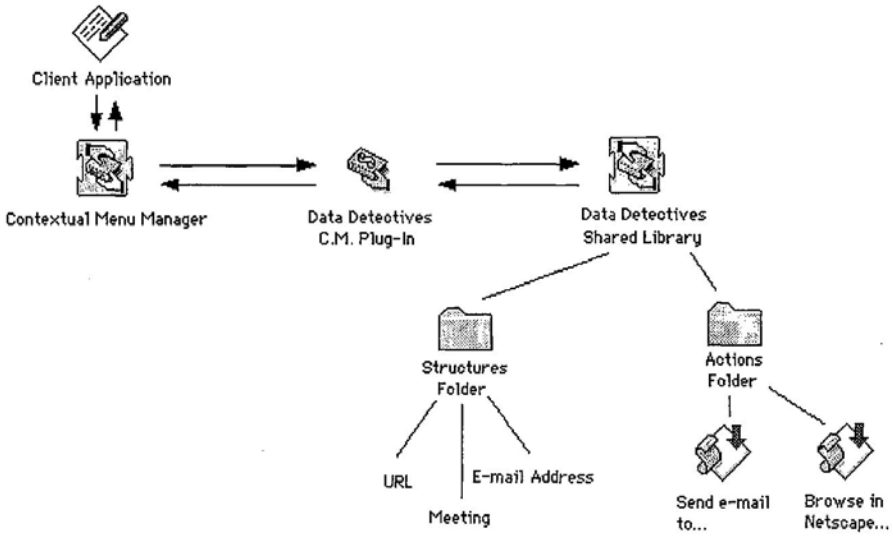
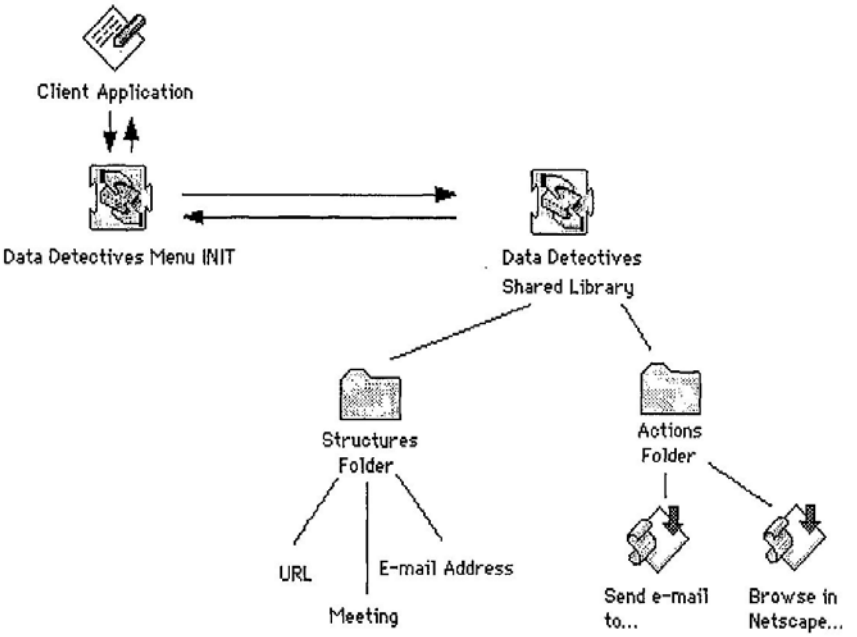
	"Find inherently structured data, let user take action on data" *See* WWDC Presentation at 2. *See* WWDC Presentation at 4-5: [Technical Overview - with Contextual Menus in Copland:]  [Technical Overview - Current]  "**on** addressLetterTo(emailAddress) **set** emailAddress **to** removeTextStyling(emailAddress) **tell** application "Now Contact 3.5"

Exhibit E

launch
activate
run
--*find the person in the Now Contact database*
try
set thePerson **to the first** person **whose** (customer 2 is emailAddress)
on error
beep
display dialog "Sorry, I don't have any information about " & emailAddress & "."
return
end try

--*get the address information for the person*
try
set firstAndLastName **to** (**the** first name **of** thePerson) & " " & (**the** last name **of** thePerson)
set theAddress **to** firstAndLastName & return & (**the** company **of** thePerson) & return & (the work address **of** thePerson) & return & (**the** work city **of** thePerson)
& ", " & (**the** work state **of** thePerson) & " " & (**the** work zip of thePerson) & return
on error
beep
display dialog "Sorry, I couldn't get the address of " & emailAddress 7 "."
return
end try

--*write the address information into ClarisWorks*
tellapplication "ClarisWorks"
try
launch
activate
--*run --don't do a "run here, or you'll get the "New document" dialog…*
open (path to system folder as string) & "SD letter template" as "WP"
on error
beep
activate
display dialog "Sorry, I couldn't open the letter template: 'SD letter template' in the System Folder."
See Write a Letter AppleScript Code.

Exhibit E

	Miller discloses or renders obvious this element. See element [1p] above. Miller further discloses a computer readable medium including program instructions (*see, e.g.*, Fig. 1 at 170). "Q. Can you give me a general description of what it is that Apple Data Detectors did? A. The idea at the time was to identify bits of information in user documents. For instance, phone numbers or email addresses or, you know, other sorts of things that could be easily identified and make it very easy for people to carry out actions on them." Miller Depo. at 27:2-9. "And so we were looking for ways to find those things in user documents and then make it easy for people to carry out actions on them, to do things with those, let's say, email addresses that were found." Miller Depo. at 39:23-40:2. "Q When it says that "Apple Data Detectors analyze the text you've selected," what do you mean by that? A. The system receives the text that the user has selected and then analyzes it with the various kinds of detectors that are installed in the system looking for the kinds of information that they are designed to identify, e-mail addresses or phone numbers or what have you. Q. It then goes on to say in this document, Miller 7, quote, "You're then presented with a menu of the things found by Apple Data Detectors. Point at one and you'll see a hierarchical menu of the actions that you can carry out on that thing. For a phone number, one of these things -- one of these would surely be to call that number," unquote. Do you see that? A. Yes. Q. When you say that "you're presented with the menu of the things," who's the "you're"? A. The user. Q. And why was -- what was the point of presenting the user with a menu that would allow them to choose particular actions? A. Well, that was how we were allowing the user to operate on the contents of their document. Q. And so, for example, for a phone number here, it identifies that one of the actions that could be carried out would be to call that number? A. Yes. Q. How would that work with Apple Data Detectors? If a phone number was detected, how would a user be given the option to call that number? MS. FRANKLIN: Objection to form. THE WITNESS: At the time there was a third-party product called

Exhibit E

	Megaphone that was a combination of a software application and a physical box that would plug into your Mac and then also allow your phone to plug into it. You could then have that application do simple voicemail responses to phone calls and it could also place outgoing phone calls from an address book. That application supported Apple events and so we were able to send that Apple events message to the Megaphone application saying, 'dial this phone number' and it would do it." Miller Depo. at 75:1-77:4. "Q So tell me how it is that clicking on "write a letter to manzi@lotus.com" results in the finding of a mailing address. A. Right. That mailing address is in the user's Now Contact address book. So the first thing that the Data Detector script does is to ask Now Contact for the mailing address of the person whose email address is manzi@lotus.com and – Q. So I assume to do that, it has to search for manzi@lotus.com? MS. FRANKLIN: Objection to form. THE WITNESS: Yes. BY MR. UNIKEL: Q. How does it find out whether or not there is a contact with manzi@lotus.com in the Now Contact database? A. That's an internal service that Now Contact provides through Apple event support. Q. What is that service? A. You can give it search parameters, in this case an email address, and say, give me back the information that you have about this person. We happened to use email address. I believe it would also operate off of a phone number. Q. Okay. So in this case, the search parameter was what? A The email address, manzi@lotus.com. Q. And so Now Contact was given that search parameter, and what would happen next? A. And it would -- Now Contact would look for an entry whose email address is manzi@lotus.com. If it found it, it would gather up the mailing address information and return it to the calling bit of code. Q. And then would it do anything with that information -- that address information that was returned? A. Now Contact would not. Now Contact's work is done. But the script would now receive that information and then start to talk to ClarisWorks, which is a word processor. It would open up a new document -- or the script would then open a new document in ClarisWorks and paste the information it had gotten from Now Contact into that new document. Q. Is that what we see then in the screenshot that is Page 4 of Miller

Exhibit E

	Exhibit 14, the pasted information? A. Yes." Miller Depo. at 126:14-128:14. "Q. And again, you described for me how this would work with an email address before. But just to be clear, can you please take me through the steps starting with what would a user see when the phone number was detected that would give them the option? A. Sure. The phone number gets recognized by the recognition component of Data Detectors and based on the user's request for writing a letter, I guess here it's referred to as address letter to, it will run the script. So it takes the phone number and opens up mail contact and says, get me – get the person whose work phone is this phone number. That information comes back from Now Contact. It continues on, it kind of breaks it up into different pieces and then opens WordPerfect, which was another word processing application that supported enough Apple events that we could work with it. And it opens a template document, writes in the date, writes the salutation and then the other information. Q. And when you say the other information, what other information would be inserted into the document as part of that? A. The date and the address, which is the address that had been retrieved from Now Contact." Miller Depo at 154:22-155:25.
displaying the document electronically using the first computer program;	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.
retrieving the first information;	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.

Exhibit E

operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	The Apple Data Detectors System discloses this element. *See* Claim 1 disclosures.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising:	The Apple Data Detectors System discloses this element. *See* Claim 23 disclosures.
providing a prompt for updating the information source to include the first information.	The Apple Data Detectors System discloses this element. *See* Claim 8 disclosures.

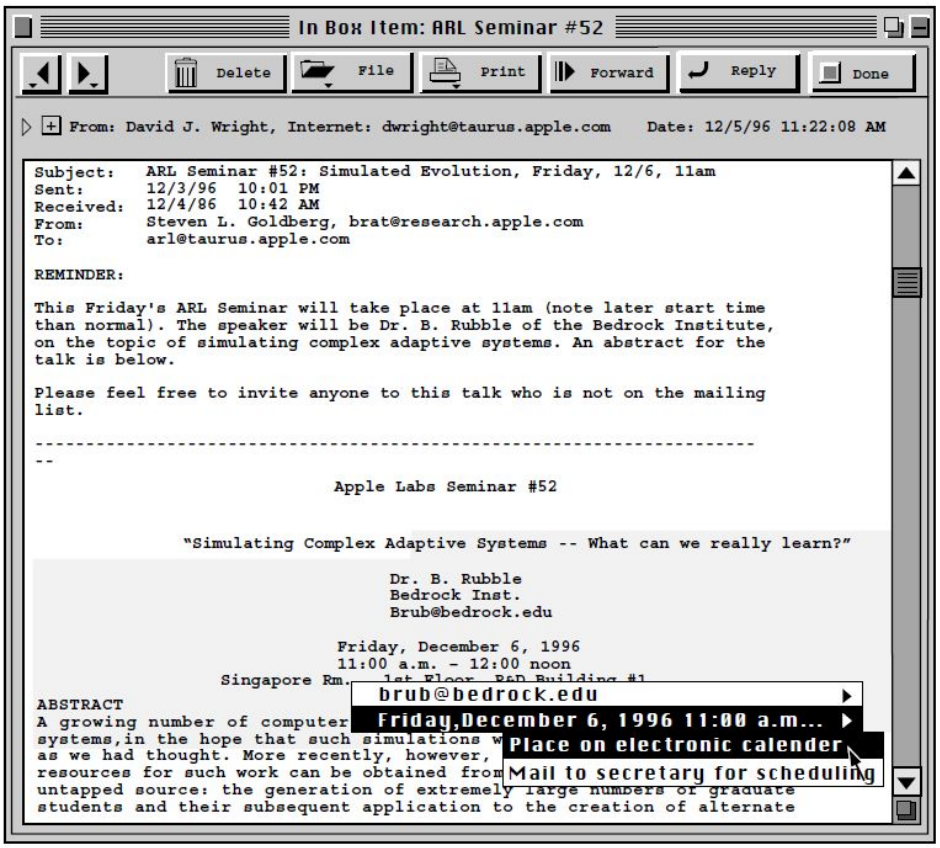
Exhibit F

Claim Chart Applying "Collaborative, Programmable Intelligent Agents" by Nardi, Bonnie A. et al., Against the '843 Patent

Collaborative Programmable Intelligent Agents ("Nardi"), describing the functionalities of Apple Data Detector, was published on March 1998. It therefore constitutes prior art under pre-AIA 35 U.S.C. § 102. As shown below, Nardi anticipates and/or renders obvious claims 1, 8, 13, 15, 17, 18, 19, 23 and 30 of the '843 Patent.

"Obviousness Statement" - To the extent that the Judge or Jury finds that Nardi does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, Nardi discloses the preamble. **Architecture and implementation.** Apple Data Detectors is an open extensible system allowing the recognition and parsing of complex structures. (Nardi, page 99) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Nardi discloses this element.

	 **Figure 1.** Sample invocation of Apple Data Detectors. The user has selected a portion of an email message describing an upcoming seminar. Two patterns are found: an email address (brub@bedrock.edu) and the announcement of the meeting (the sequence of date and time information starting Friday, Dec. 6, 1996). These patterns are presented in the pop-up menu; by pointing at the date information in the menu; a second pop-up menu offers a choice of actions: place an entry for the meeting on the user's electronic calendar or mail the selection to the user's secretary. The user can select one, thereby running a small application, or move the cursor off the menu, eliminating the pop-up menu and canceling any actions. (Nardi, page 97) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Nardi discloses this element. **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The element and the claim are further rendered obvious for the reasons stated in Exhibit U, Table 15. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered

	obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	Nardi discloses this element. A user can select a whole document or part of a document without having to make a careful selection; the grammars find any embedded structures they know about within the selection and offer an appropriate set of actions from which to choose. (Nardi, page 98)
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Nardi discloses this element. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.

in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Nardi discloses this element. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Nardi discloses this element. The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered

	obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element. **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 7, 17, and 20.
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element. The Apple Data Detectors' architecture separates detectors and actions so that more than one action can exist for any detector (without having to duplicate the detector for each action). Hence, a detector written by one person to support one task can be used by another detector for another task. Detectors are thus reusable and easily shared. (Nardi, page 100) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element.

into the document.	Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	Nardi discloses claim 1. *See* claim 1. Nardi further discloses this element. Once identified, the structure's type can be used to identify appropriate actions that might be carried out, like placing a meeting on a calendar, adding an address to an address book, dialing a phone number, opening a URL, finding the current price of a stock, filing an ISBN number, and compiling a list of abstracts. (Nardi, page 98) **User interface.** To use Apple Data Detectors, users select a region of a

	document with some information of interest. Pressing a modifier key and the mouse button instructs the system to analyze the data within the selected region and to find all structures for which it has grammars. It then offers appropriate actions for each structure (see Figure 1). (Nardi, page 98) The system parses the selected text according to the grammars associated with them. For each structure found by a detector, a data record is produced describing the structure. This record can then be passed to an action script for execution, in much the same way a subroutine is invoked with a specific set of parameters. These parameters depend, of course, on the kind of structure found by a given detector. Detectors for strings and atomic patterns typically create a record containing only the structure that was found—such as the name of a conference room or an email address. Detectors for complex patterns, such as a meeting announcement, produce records containing each of the components playing a part in the recognition of the pattern. Note that a detector can have an action associated with it and also play a part in a more complex detector. For example, a conference room detector could have a "Show on map" location indicating where that room is located and also play a part in defining the more complex meeting detector. (Nardi, pages 99-100) For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	To the extent the preamble is limiting, Nardi discloses the preamble. *See* claim 1 above.
displaying the document electronically using the first computer program;	Nardi discloses this element. *See* claim 1 above.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to	Nardi discloses this element. *See* claim 1 above.

find second information related to the first information;	
retrieving the first information;	Nardi discloses this element. *See* claim 1 above.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Nardi discloses this element. *See* claim 1 above.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Nardi discloses this element. *See* claim 1 above.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Nardi discloses this element. *See* claim 1 above.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising: providing a prompt for updating the information source to	Nardi discloses claim 23. *See* claim 23. Nardi further discloses this element. *See* claim 8 above.

include the first information.	

Exhibit G

Claim Chart Applying U.S. Patent No. 5,946,647 ("Miller") Against the '843 Patent

U.S. Patent No. 5,946,647 to Miller et al. ("Miller") was filed on February 1, 1996. It therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(e). As shown below, Miller anticipates and/or renders obvious claims 1, 8, 13, 15, 17, 18, 19, 23, and 30 of the '843 patent.

"Obviousness Statement" - To the extent that the Judge or Jury finds that Miller does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent this preamble is limiting, Miller discloses this preamble. For example, Miller states: "A system and method causes a computer to detect and perform actions on structures identified in computer data." Miller at Abstract

Exhibit G

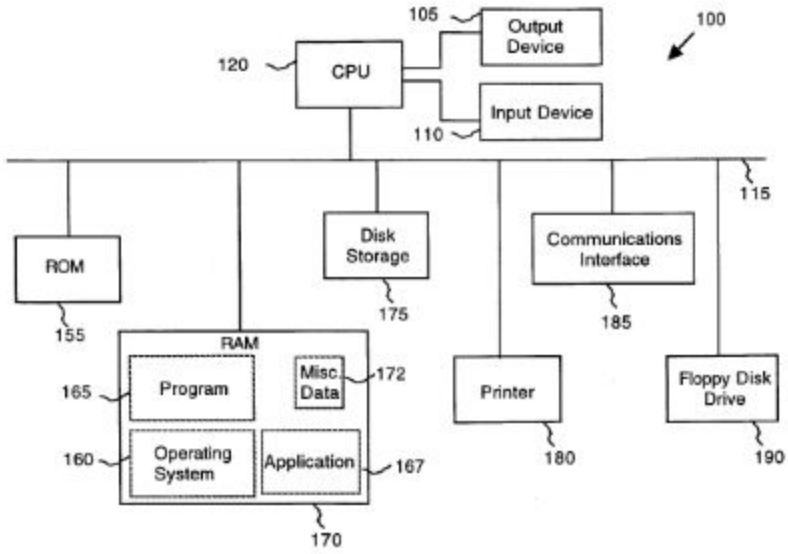


FIG. 1

Fig.1

See also FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18.

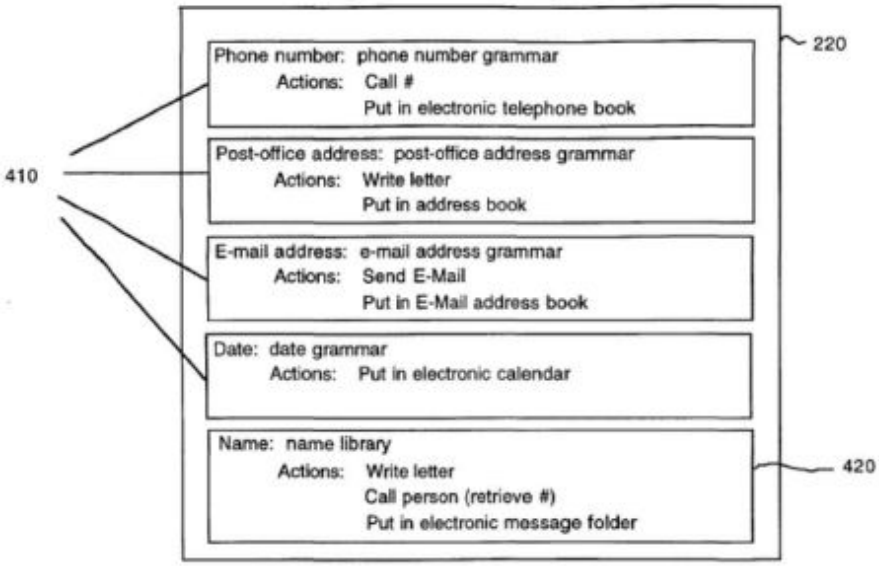


FIG. 4

"The analyzer server receives data from a document."

2:28.

Exhibit G

	"Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." 2:42-46. "Application 167 is a word-processor or e-mail program, that presents data on output device 105 to a user. The program 165 of the present invention…identif[ies] structures in the data presented by application 167, to associate actions with the structures identified in the data, to enable the user to select a structure and an action, and to automatically perform the selected action on the identified structure." 3:38-44. "Upon identification of a structure in the text, parser 310 links the actions associated with the grammar to the identified structure." 4:64-66. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5:18. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Miller discloses this element. Documents are displayed using a first computer program, such as a word processor (application 167 in Fig. 1). *See, e.g.,* Figs. 1 and 5; 5:19-22 ("FIG. 5 shows a window 510 presenting an exemplary document 210 ... "); 3:36-38 ("Application 167 is a program, such as a word processor or e-mail program, that presents data on output device 105 to user."). Output device 105 (Fig. 1) can be a CRT display device. 3:26-38. For example (and without limitation to the Obviousness Statement that is

Exhibit G

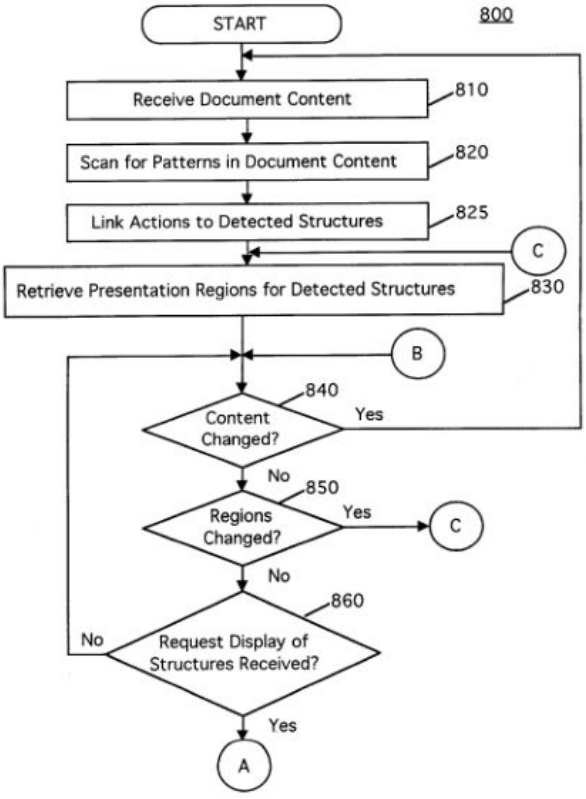
	incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Miller discloses this element. "Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." 2:42-46. "FIGS. 8 and 9 display a flowchart illustrating preferred method 800 for recognizing patterns in documents and performing actions. This method is carried out during the run-time of application 167." 5:51-54.  FIG. 8 *See also* FIGS. 5 and 6. "If the document content being displayed on output device 105 is changed 840, for example by the user adding or modifying text, method 800 restarts."

Exhibit G

	5:64-66. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 *See also, e.g.,* Figs. 5 and 6; 3:61-64 ("Analyzer server 220 ... uses patterns to parse document 210 for recognizable structures."); 5: 19-36 (" ... As illustrated in FIG. 6, analyzer server 220 identifies the phone number, post-office address, e-mail address and name ... ").  FIG. 5 Fig. 5. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.

Exhibit G

retrieving the first information;	Miller discloses this element. "The analyzer server receives data from a document." 2:28. *See also, e.g.,* 3:8-10 ("FIG. 6 illustrates a window with the identified structures in the example document of FIG. 5 highlighted based on the analyzer server of FIG. 4.").
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Miller discloses this element. "An input device 110, such as a keyboard and mouse, and an output device 105, such as a CRT or voice module, are coupled to CPU 120." 3:26-28. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic

Exhibit G

	telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Miller discloses this element. *See* claim 1 above. "Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." 2:42-46. "Window 510 includes a button 520 for initiating program 165 . . . . Upon initiation of program 165, system 100 transmits the contents of document 210 to analyzer server 220." 5:22-26. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 *See also, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "); 5:6-37; 3:52-4:10 ("After identifying structures and linking actions, application program interface 230 [of program 165] communicates with application 167 to obtain information on the identified structure so that user interface 240 can successfully present and enable selection of the actions"). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered

Exhibit G

	obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Miller discloses this element. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	Miller discloses claim 1. *See* claim 1 above. Miller further discloses this element. When a user selects a detected structure, the system prompts the user to select a candidate action by displaying a pop-up menu of actions. *See, e.g.,* 4:27-31 ("Upon selection of this structure, user interface 240 presents and enables selection of the linked candidate actions using any selection mechanism, such as a conventional pull-down or pop-up menu."). As shown in Fig. 4, several of these actions update an information source to include first information (e.g., put in address book). *See, e.g.,* Fig. 4; 5:6-18. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book." 5:38-43. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered

Exhibit G

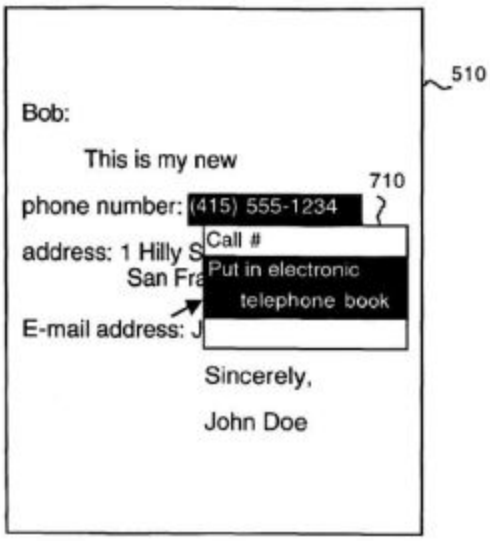
	obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	Miller discloses claim 1. *See* claim 1 above. Miller further discloses this element. *See, e.g.*: Claim 1 disclosure. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for

Exhibit G

	post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	Miller discloses claim 1. *See* claim 1 above. Miller further discloses this element. *See, e.g.*: Claim 1 disclosure. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.

Exhibit G

<table>
<tr><td></td><td> 

FIG. 7

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

5:6-5: 18.

See also, e.g., FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18.

The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... ").

For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7, 17, and 20.</td></tr>
<tr><td>Claim 17</td><td></td></tr>
<tr><td>A method according to claim 1, wherein the information source is</td><td>Miller discloses claim 1. See claim 1 above.</td></tr>
</table>

Exhibit G

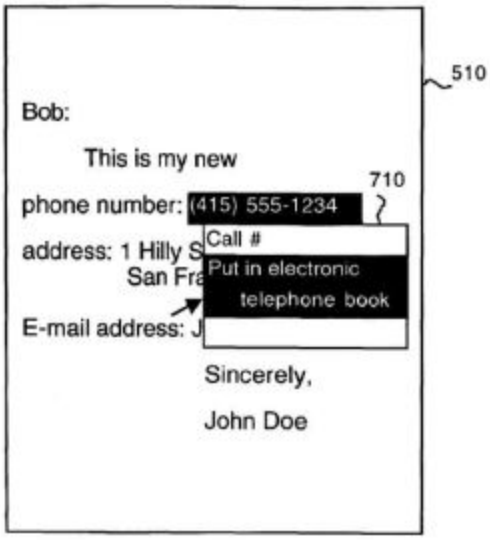
associated with the second computer program and is available through the computer.	Miller further discloses this element. *See, e.g.*: Claim 1 disclosure. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to

Exhibit G

	the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	Miller discloses claim 1. *See* claim 1 above. Miller further discloses this element. *See, e.g.*: Claim 1 disclosure. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.

Exhibit G

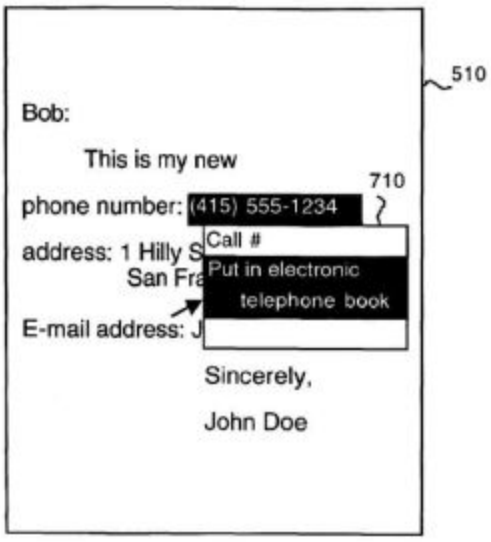


FIG. 7

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

5:6-5: 18.

See also, e.g., FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18.

The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... ").

See also, e.g., Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920.").

For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.

Claim 19	

Exhibit G

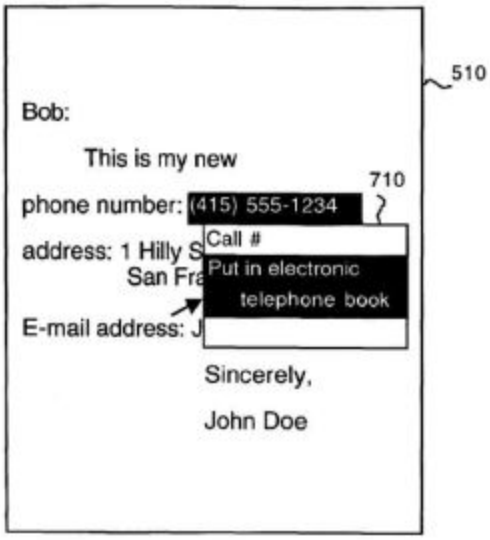
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	Miller discloses claim 1. *See* claim 1 above. Miller further discloses this element. *See, e.g.*: Claim 1 disclosure. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." 5:38-47.  FIG. 7 "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an

Exhibit G

	address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5: 18. *See also, e.g.,* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.,* Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the process comprising:	Miller discloses this element. *See* claim 1 above. Miller further discloses this element. *See, e.g.*, Fig. 1 at 170.
displaying the document electronically using the first computer program;	Miller discloses this element. *See* claim 1 above.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Miller discloses this element. *See* claim 1 above.
retrieving the first information;	Miller discloses this element. *See* claim 1 above.

Exhibit G

providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Miller discloses this element. *See* claim 1 above.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Miller discloses this element. *See* claim 1 above.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Miller discloses this element. *See* claim 1 above.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising: providing a prompt for updating the information source to include the first information.	Miller discloses claim 23. *See* claim 23 above. Miller further discloses this element. *See* claim 8.

Exhibit G

Exhibit H

Claim Chart Applying LiveDoc Version 0.8 System Against the '843 Patent

LiveDoc Version 0.8 and its associated Data Detectors (aka Structure Detectors) technology (together, "LiveDoc Version 0.8 System") were invented in the United States by employees of Apple Computer at least by December 5, 1995. The inventions of LiveDoc Version 0.8 were later described in U.S. Patent 5,946,647 filed on February 1, 1996, and the articles "From Documents to Object: An Overview of LiveDoc" and "Drop Zones: An Extension of LiveDoc" in the April 1998 issue of SIGCHI Bulletin at pages 53-63 and were publicly demonstrated at Mac World on or around August 7, 1996 in the United States, and the inventions were not abandoned, suppressed, or concealed. LiveDoc Version 0.8 therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(g).

As shown below, an Apple computer system running LiveDoc Version 0.8 and the applications Now Contact, Claris Emailer, and Claris Works ("LiveDoc Version 0.8 System") also constituted prior art under pre-AIA 35 U.S.C. § 102(g) and anticipated claims 1, 8, 13, 15, 17-19, 23, and 30 of the '843 patent.

"Obviousness Statement" - To the extent that the Judge or Jury finds that the LiveDoc Version 0.8 System does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc Version 0.8 System (including specific publications describing aspects of the LiveDoc Version 0.8 System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

Evidence of the design and operation of LiveDoc Version 0.8 and LiveDoc Version 0.8 System include the following:

- U.S. Patent No. 5,946,647 to Miller et al. ("Miller")
- Deposition of James Miller ("Miller Depo.")
- The April 1998 issue of the SIGCHI Bulletin, including articles entitled "From Documents to Object: An Overview of LiveDoc" ("LiveDoc") and "Drop Zones: And Extension of LiveDoc" ("Drop Zones")
- "Read me! 0.8" file for LiveDoc Version 0.8 ("Read Me File"), available for inspection at DLA Piper
- External Requirements Specification for LiveDoc for Applications, version 1.0a3, June 12, 1995
- Structure Detectors/LiveDoc for Maxwell: A Starting Point, December 6, 1995

Exhibit H

- "LiveDoc / Structure Detectors, v. 0.8" (Dec. 5, 1995)
- Mac World Demonstration video captured on or around August 7, 1996 ("Video")
- Executable version of LiveDoc Version 0.8, available for inspection at DLA Piper
- Source code for LiveDoc Version 0.8, available for inspection at DLA Piper
- Screenshots of executable version of LiveDoc Version 0.8 and LiveDoc Version 0.8 System, contained below
- U.S. Patent No. 5,946,647 to Miller, filed on February 1, 1996, as charted in Exhibit G and incorporated here by reference
- "From Documents to Object: An Overview of LiveDoc" and "Drop Zones: An Extension of LiveDoc", published in an April 1998 issue of the SIGCHI Bulletin, as charted in and incorporated here by reference

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, LiveDoc Version 0.8 System discloses the preamble. LiveDoc Version 0.8 System included LiveSimpleText. "LiveSimpleText is a version of SimpleText that has been modified to use the LiveDoc API and, thereby, the Structure Detectors background application. When 'LiveDoc' mode is activated (via the *Edit* menu), the text shown in the window is analyzed by Structure Detectors to find meaningful patterns and phrases." Read Me File. *See, e.g.*, LiveDoc at 53 ("There is a real opportunity to advance the computing field here, by bringing these two worlds together: by enabling an ordinary document, built with any application, to automatically offer users access to some of the meaningful bits of its content, and by helping users carry out appropriate actions on these objects."); at 58 ("Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."); Drop Zones at 61 ("Another call to the address book application, guided by another mapping rule, will return the email address for the identified person.").

Exhibit H

Figure 1: Drop zone is shown in the window labeled 'Activities'. The window at the top called 'Test' is a LiveDoc window showing proper names, e-mail addresses phone number, URL, date and stock market ticker codes

Drop Zones at 60 (Figure 1).

"A system and method causes a computer to detect and perform actions on structures identified in computer data."

Abstract

Exhibit H

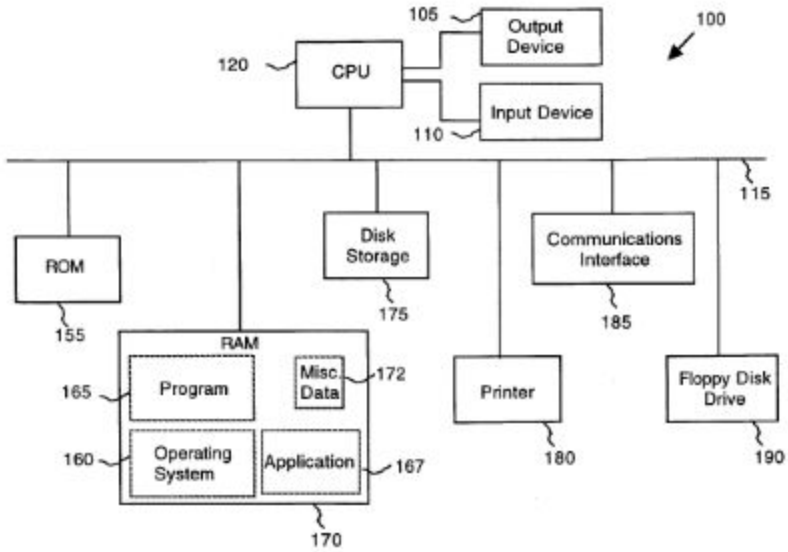


FIG. 1

Fig.1

See also FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18.

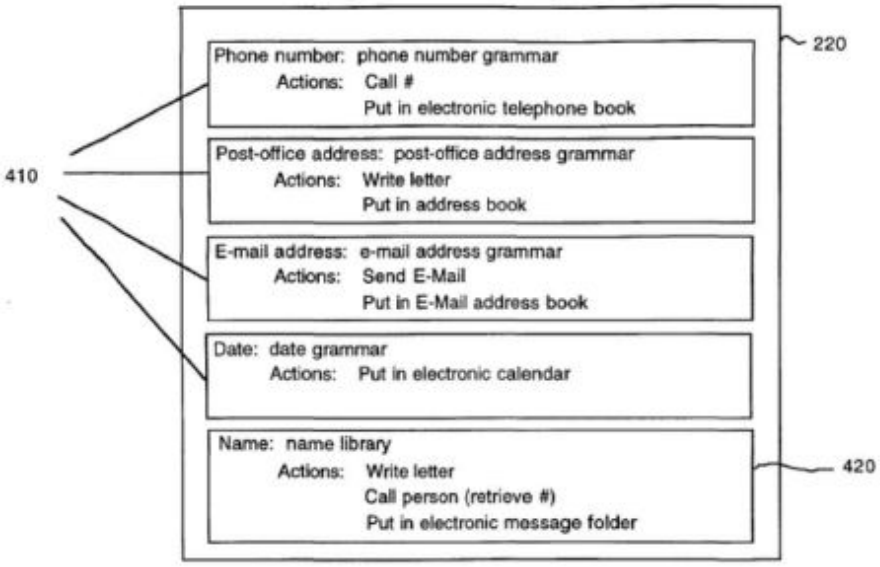


FIG. 4

"The analyzer server receives data from a document."

2:28.

Exhibit H

	"Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." 2:42-46. "Application 167 is a word-processor or e-mail program, that presents data on output device 105 to a user. The program 165 of the present invention…identif[ies] structures in the data presented by application 167, to associate actions with the structures identified in the data, to enable the user to select a structure and an action, and to automatically perform the selected action on the identified structure." 3:38-44. "Upon identification of a structure in the text, parser 310 links the actions associated with the grammar to the identified structure." 4:64-66. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." 5:6-5:18. "LiveDoc was a research project that was exploring a different way of handling the Data Detector matter where instead of having to manually select region of text and clicking a mouse button and dealing with the pop-up menus, you could hold down a key and the display would highlight to show you the regions that had been found that under normal circumstances for Data Detectors would be shown in the hierarchal pop-up menu, but to actually show them as selectable regions on the screen. And then you could click on one of them on the screen and actually get the menu of actions attached to that item." Miller Depo. at 161:16-162:4. "Q. So what are the kinds of things then that LiveDoc could find and then highlight for a user?

Exhibit H

	A. Well, the same things that were found by the shipping version of Data Detectors. In fact, both the shipping version and LiveDoc used the same underlying Data Detector capability." Miller Depo. at 174:14-20. "Am I correct then that the user could hit a button that would either show the highlighting or have the highlighting disappear if the button was released? A. Yes." Miller Depo. at 175:10-14. "Q. When you say 'pointing at a highlight and pressing the mouse button,' I'm assuming that means if the user points at a highlight and presses the mouse button, the options are presented? A. Yes. If the user places the mouse cursor over the highlight and then presses the mouse button. Q. And then so if the user then clicks the mouse button, what happens? A. If they press down on the mouse button, then they get the menu of things that can be done to that item. Q. And how do they select one of those things? A. By dragging the mouse cursor to the desired item and releasing the mouse button. Q. So there's only one click involved in that, I think as you just described it? A. It's one click down to reveal the menu and then releasing the button to select the action. Or you can move off the selected region to some other part of the screen and let go, in which case the menu would just go away." Miller Depo. at 177:7-178:5. "So the LiveDoc manager gets the text from the visible part of the screen and looks for the recognizable patterns in it. Now, it does that just by looking at this long string of texts making up all the content of the screen. So as noted here, it might say that there is a particular pattern between – or in this substring of the character. The LiveDoc manager is the thing that is going to draw those highlights." Miller Depo. at 179:12-22. "[W]hy did you make LiveDoc its own separate piece of code where it had to exchange information with the application rather than just making it a part of every application? A. I think for much the same reason – I mean, this is going back quite a ways. But I think it was the same sort of logic as for making the Data Detector engine a separate thing. That by keeping the LiveDoc manager as an external component, we would be more able to connect into potentially more applications. We wouldn't be building LiveDoc capability into all of those applications. All we would – all we would need would be for that

Exhibit H

	application to be able to tell the LiveDoc manager where on the screen are these characters." Miller Depo. at 180:11-181:3. "If they loaded a document into the LiveDoc application, there was a menu item that you had to pick to enable LiveDoc that was just sort of a convenience for us as we were developing it. But if that had been enabled, then you would press down I believe the option key on the keyboard. And any structures in the visible part of that are document that had been recognized by Data Detectors would be highlighted. And at that time, while those were highlighted, you could point the mouse cursor at one of those structures, press down and you would get the menu of actions that could be applied to the item. Q. Okay. A. You would then select one and pick it and it would run just as it had run with Data Detectors." Miller Depo. at 186:4-23. *See* Video at 0:30-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	LiveDoc Version 0.8 System discloses this element. LiveSimpleText was a word processing application that allowed users to enter text.

Exhibit H

Figure 1: Drop zone is shown in the window labeled 'Activities'. The window at the top called 'Test' is a LiveDoc window showing proper names, e-mail addresses phone number, URL, date and stock market ticker codes

Drop Zones at 60 (Figure 1).

"[T]he structure detection process is run in the background on the visible document's text, whenever that document is presented or updated." Drop Zones at p. 55.

The window named 'test' in Figure 1 belongs to a LiveDoc-enabled word processor, LiveSimpleText (see [6]), and shows a number of structures within the document in view having been recognized by the analyzers." Drop Zones at p. 60.

See also LiveDoc at 58 and Fig. 2; Drop Zones at Fig. 2.

Documents are displayed using a first computer program, such as a word processor (application 167 in Fig. 1). *See, e.g.,* Figs. 1 and 5; 5:19-22 ("FIG. 5 shows a window 510 presenting an exemplary document 210 ... "); 3:36-38 ("Application 167 is a program, such as a word processor or e-mail program, that presents data on output device 105 to user."). Output device 105 (Fig. 1) can be a CRT display device. 3:26-38.

Exhibit H

	See also Video at 0:00-0:29. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	LiveDoc Version 0.8 System discloses this element. Holding down the Option key would result in "patterns and phrase" being highlighted in color. Read Me File. The patterns and phrases that could be identified included: email addresses, URLs, FTP file specifications, America Online keywords, Internet host names, Internet newsgroups, names of people, names of companies, and strings specified in the People, Companies, and Strings filed in the System Folder:Preferences:LD Dictionary Prefs folder. Read Me File. *See, e.g.*, LiveDoc at 55 ("In LiveDoc, the structure detection process is run in the background on the visible document's text, whenever that document is presented or updated. … Pointing at a highlight and pressing the mouse button then displays the menu of actions that can be applied to the structure, as shown in Fig 2."); Drop Zones at 59 ("For example, the name 'Apple Computer, Inc.' could be associated with such actions as, 'Find the corporate headquarters on a map', 'Get Apple's corporate phone number', 'Get the current trading price of Apple stock', 'Get the people in my address book associated with Apple' and so forth."). "[M]any meaningful bits of information are computationally quite easy to recognize: recognizing an e-mail address ("fred@apple.com") or a URL ("http://www.apple.com") takes little more than a context-free grammar, if not merely a regular expression parser." LiveDoc at p. 53. "Various kinds of recognizers, including context free grammars, are used to describe the structures to be found; these structures can be made up of either a single lexical term (either a variable structure like a phone number, or a collection of static strings, like company names) or multiple terms (for instance, a meeting can be defined as a combination of date, time, and venue structures)." Drop Zones at p. 59.

Exhibit H

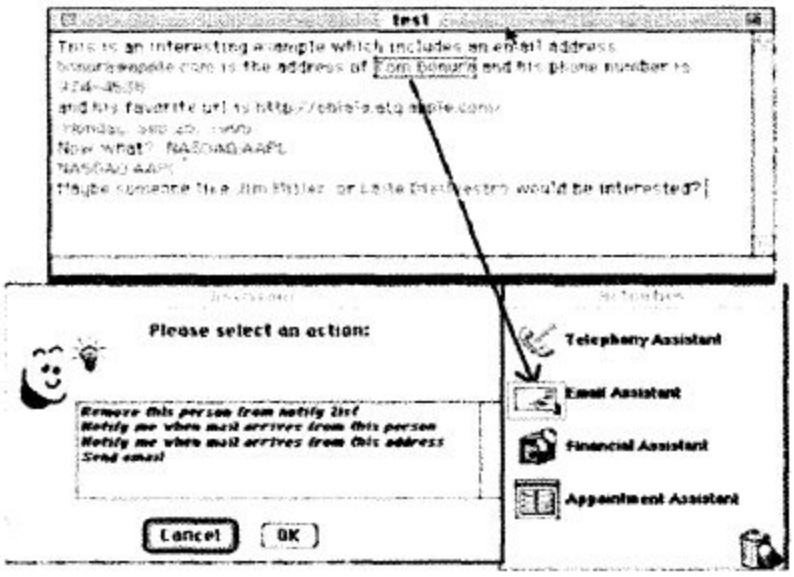
	 **Figure 2:** A user interaction with Drop Zones Drop Zones at 60 (Figure 2).

Exhibit H

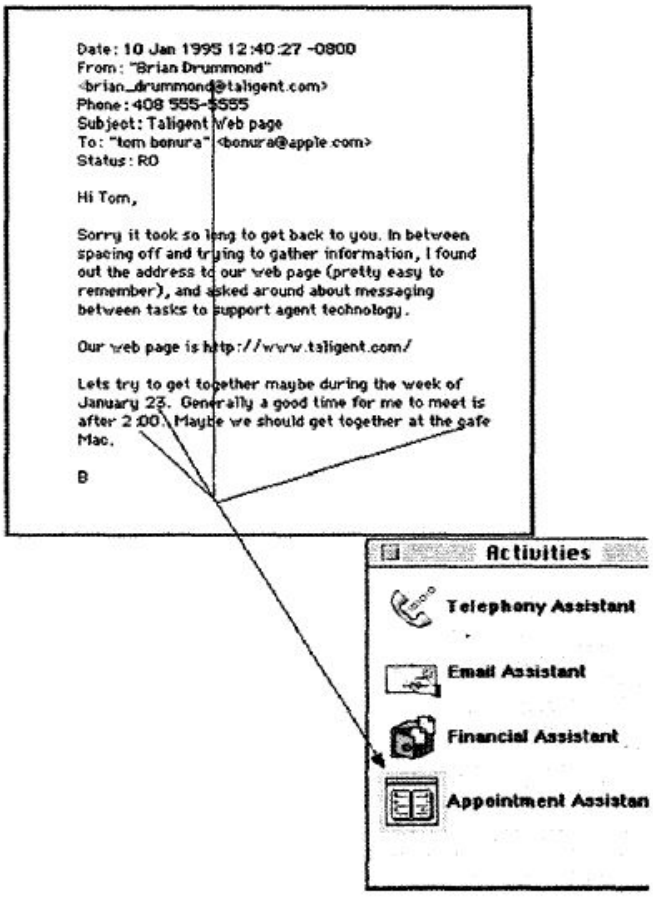


Figure 3: Selecting multiple items from a LiveDoc window for interpretation

Drop Zones at 61 (Figure 3).

"Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications."

2:42-46.

"FIGS. 8 and 9 display a flowchart illustrating preferred method 800 for recognizing patterns in documents and performing actions. This method is carried out during the run-time of application 167."

5:51-54.

Exhibit H

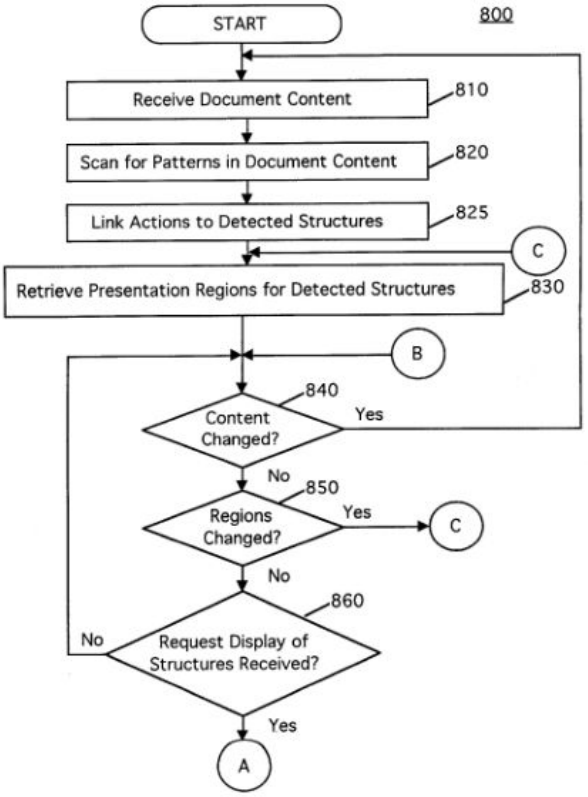


FIG. 8

See also FIGS. 5 and 6.

"If the document content being displayed on output device 105 is changed 840, for example by the user adding or modifying text, method 800 restarts."

5:64-66.

"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address."

5:6-5: 18.

Exhibit H

	See also FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 *See also, e.g.,* Figs. 5 and 6; 3:61-64 ("Analyzer server 220 ... uses patterns to parse document 210 for recognizable structures."); 5: 19-36 (" ... As illustrated in FIG. 6, analyzer server 220 identifies the phone number, post-office address, e-mail address and name ... ").  FIG. 5 Fig. 5. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	LiveDoc Version 0.8 System discloses this element. *See* previous element. "LiveDoc knows where these structures appear in the text passed to it." LiveDoc at p. 56. “Small pieces of code can then be associated with each structure to instruct applications to carry out specific user actions on the discovered structures- perhaps to tell a telephony application to Dial this phone number.” Drop Zones at p. 59. "The analyzer server receives data from a document." 2:28.

Exhibit H

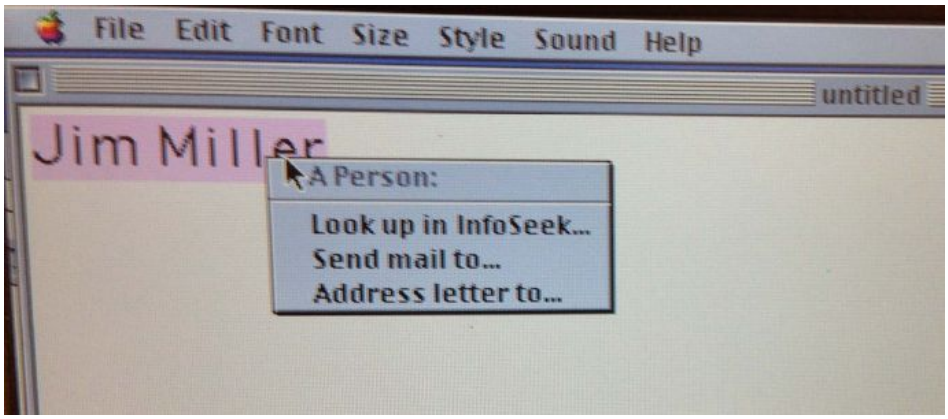
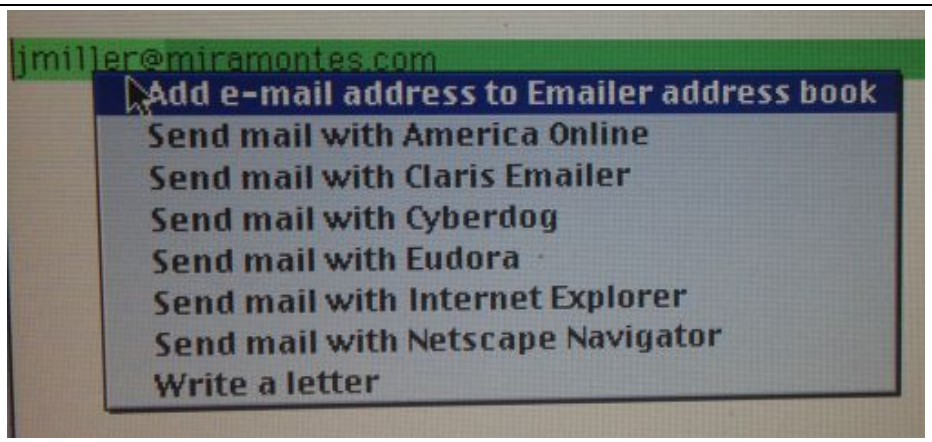
	See also, e.g., Miller at 3:8-10 ("FIG. 6 illustrates a window with the identified structures in the example document of FIG. 5 highlighted based on the analyzer server of FIG. 4.").
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	LiveDoc Version 0.8 System discloses this element. If a user types a name in the LiveSimpleText document, pressing the Option key would highlight the name, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail to" would cause the system to retrieve an email address for the selected name from the Now Contact database, and the system would then create a new email for that address. Selecting the option "Address letter to" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address. If a user types an email address in the LiveSimpleText document, pressing the Option key would highlight the email address, and right-clicking the mouse would pull up the following options:

Exhibit H



These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail with Claris Emailer" would cause the system to retrieve an email address for the selected name from the Claris Emailer address book, and the system would then create a new email for that address in Claris Emailer. The email would include a "Destination" field which would display company names for certain URL domains. Selecting the option "Write a letter" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address using Claris Works. For example:

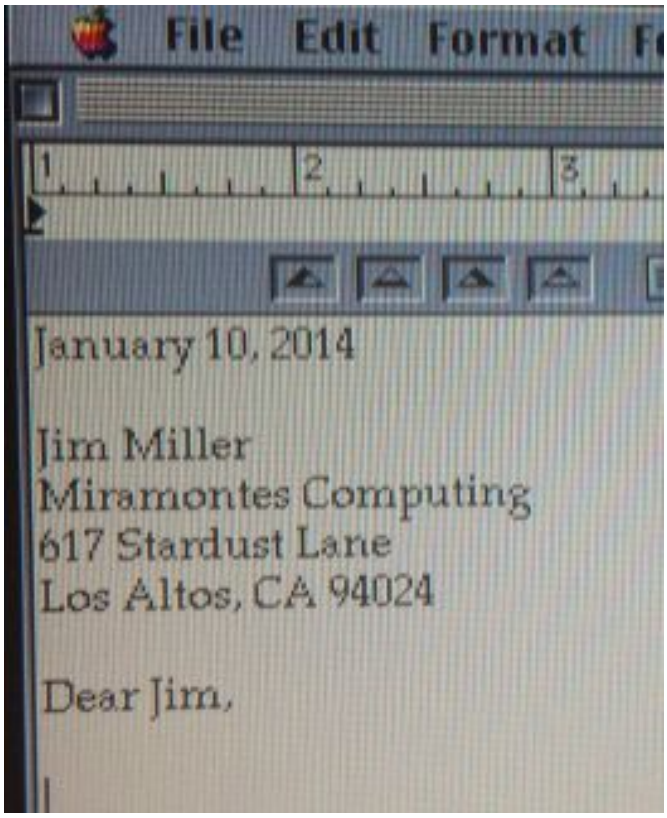


Exhibit H

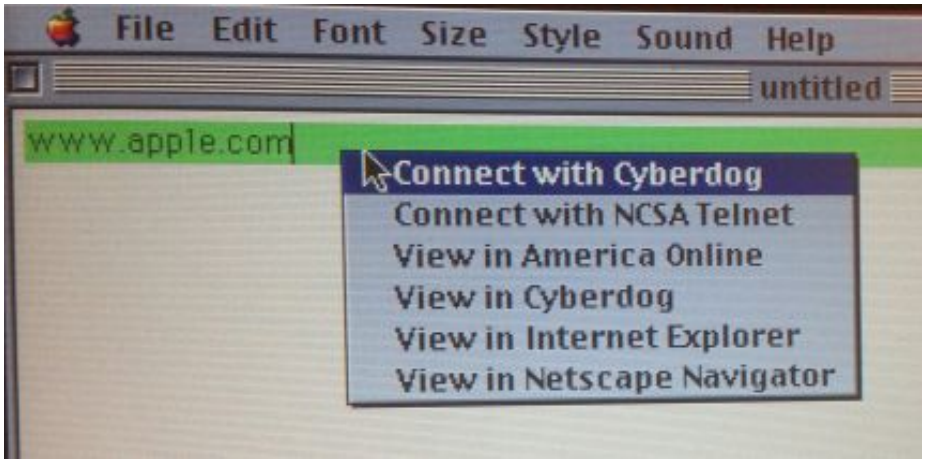
	If a user types a URL in the LiveSimpleText document, pressing the Option key would highlight the URL, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "View in Netscape Navigator" would allow the user to view the website associated with that URL. LiveDoc version 0.8 System and the prior art references and systems set forth in Exhibit U, Table 8 (e.g., Pandit), and relate to the same field of art, which is communication devices and methods. Specifically, LiveDoc version 0.8 System and certain prior art references and systems set forth in Exhibit U, Table 8 (e.g., Pandit) relate to the human-computer interaction of recognizing certain types of text as contact information in an electronic document. A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc version 0.8 System and the prior art references and systems in Exhibit F. A POSITA would have recognized advantages and benefits to have LiveDoc version 0.8 System to configure a input device in its first computer program.

Exhibit H

Figure 1: Drop zone is shown in the window labeled 'Activities'. The window at the top called 'Test' is a LiveDoc window showing proper names, e-mail addresses phone number, URL, date and stock market ticker codes

Drop Zones at 60 (Figure 1).

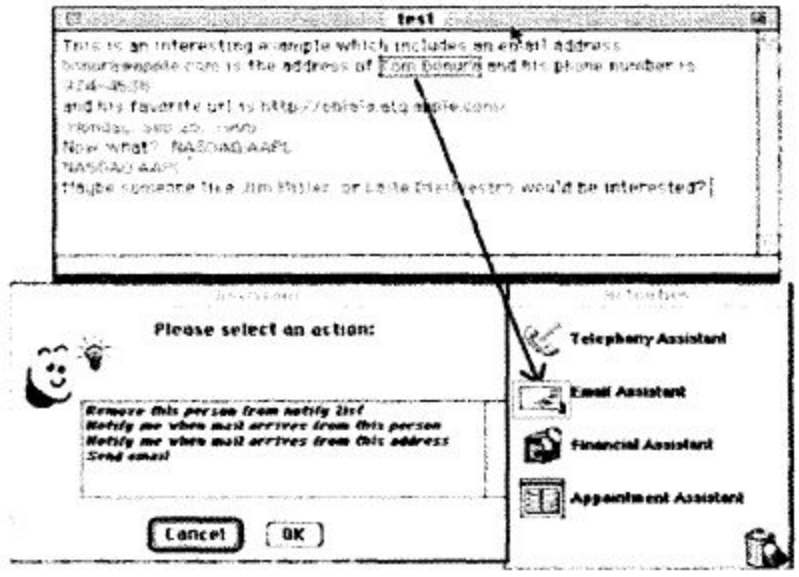


Figure 2: A user interaction with Drop Zones

Exhibit H

	Drop Zones at 60 (Figure 2). *See, e.g.*, LiveDoc at 55 ("The results of LiveDoc's analysis are then presented by visually highlighting the discovered structures with a patch of color around the structure."). *See, e.g.*, Drop Zones at 60 ("When an object is selected, it is sent to the Drop Zone control system. Each of the assistants determines if it is able to accept and act upon the set of currently selected objects."). *See, e.g.*, LiveDoc at 56 ("LiveDoc knows where these structures appear in the text passed to it … but it has no idea where in the window those characters physically appear, and, thus, where the highlights should appear: this is information held by the application, not by LiveDoc. Hence, LiveDoc must ask the application for the information about the structures it has found via a callback. Once this information is available, the highlights and their associated mouse-sensitive regions can be constructed."). *See, e.g.*, Drop Zones at 59 ("For example, the name 'Apple Computer, Inc.' could be associated with such actions as, 'Find the corporate headquarters on a map', 'Get Apple's corporate phone number', 'Get the current trading price of Apple stock', 'Get the people in my address book associated with Apple' and so forth."); Drop Zones at 60-61 ("Another call to the address book application, guided by another mapping rule, will return the e-mail address for the identified person."); LiveDoc at 58 ("Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."). *See, e.g.*, Drop Zones, Fig. 2 ("send email"); Drop Zones at 60-61; LiveDoc at 58 and Fig. 2. "These actions can then be offered to users by visually highlighting the discovered structures and attaching pop-up menus to the highlights." Drop Zones at p. 59. "Consider the situation in which a user drags a telephone number to the E-mail Assistant, presumably so that the user can send an e-mail message to the person who possesses that phone number. A literal examination of the object reveals nothing of use to the assistant: sending e-mail messages requires an e-mail address, not a phone number. However, as part of its design as an assistant for e-mail tasks, the E-mail Assistant includes an axiom that can derive e-mail addresses from phone numbers, by finding a person with the given phone number, and then obtaining the

Exhibit H

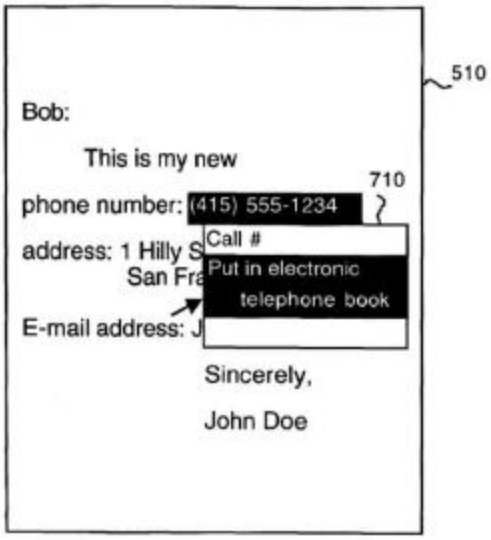
	e-mail address of that person." Drop Zones at p. 61. "Where does the E-mail Assistant find the set of people whose phone numbers and e-mail addresses can be examined? The Drop Zones representational system provides two ways through which an assistant can gain access to this information. Mappings can be built between the objects inside the Drop Zones representational system (e.g., there is an object called PERSON, which has such attributes as PhoneNumber and EmailAddress) and databases or other applications. Such a mapping, in combination with a scripting language or some other programmatic way of manipulating applications, enables the E-mail Assistant to look inside an address book application for a person with the stated phone number." Drop Zones at p. 61. "An input device 110, such as a keyboard and mouse, and an output device 105, such as a CRT or voice module, are coupled to CPU 120." Miller at 3:26-28. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book. Upon selection of the action ... user interface 240 transmits the corresponding telephone number and selected action to action processor 250." Miller at 5:38-47.  FIG. 7

Exhibit H

	"FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the 'e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5:18. *See also, e.g.*, Miller at FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. The "detect structures" button 520 in Fig. 5 is an input device that allows the user to enter a command to initiate the parsing operation. *See, e.g.*, Miller at 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "). *See also, e.g.*, Miller at Figs. 4 and 9; 6:9-33 ("[I]f an action is selected 940, the action is executed 950 on the structure selected in block 920."). *See also* Video at 1:27-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	LiveDoc Version 0.8 System discloses this element. *See* previous element. *See, e.g.*, Drop Zones at 61 ("When objects are selected, they are inspected by the assistants in the Drop Zone. These assistants are built around a collection of facts and axioms that determine whether and how they can operate in some meaningful way on various kinds of objects."). "Consider the situation in which a user drags a telephone number to the E-mail Assistant, presumably so that the user can send an e-mail message to the person who possesses that phone number. A literal examination of the object reveals nothing of use to the assistant: sending e-mail

Exhibit H

	messages requires an e-mail address, not a phone number. However, as part of its design as an assistant for e-mail tasks, the E-mail Assistant includes an axiom that can derive e-mail addresses from phone numbers, by finding a person with the given phone number, and then obtaining the e-mail address of that person." Drop Zones at p. 61. "Where does the E-mail Assistant find the set of people whose phone numbers and e-mail addresses can be examined? The Drop Zones representational system provides two ways through which an assistant can gain access to this information. Mappings can be built between the objects inside the Drop Zones representational system (e.g., there is an object called PERSON, which has such attributes as PhoneNumber and EmailAddress) and databases or other applications. Such a mapping, in combination with a scripting language or some other programmatic way of manipulating applications, enables the E-mail Assistant to look inside an address book application for a person with the stated phone number." Drop Zones at p. 61. "Since the program may be executed during the run-time of another program, i.e. the application which presents the document, such as Microsoft Word, an application program interface provides mechanisms for interprogram communications." 2:42-46. "Window 510 includes a button 520 for initiating program 165 . . . . Upon initiation of program 165, system 100 transmits the contents of document 210 to analyzer server 220." 5:22-26. *See also* FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18 *See also, e.g.,* 5:22-37 ("Window 510 includes a button 520 for initiating program 165 ... "); 5:6-37; 3:52-4:10 ("After identifying structures and linking actions, application program interface 230 [of program 165] communicates with application 167 to obtain information on the identified structure so that user interface 240 can successfully present and enable

Exhibit H

	selection of the actions"). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	LiveDoc Version 0.8 System discloses this element. *See* previous element. "Consider the situation in which a user drags a telephone number to the E-mail Assistant, presumably so that the user can send an e-mail message to the person who possesses that phone number. A literal examination of the object reveals nothing of use to the assistant: sending e-mail messages requires an e-mail address, not a phone number. However, as part of its design as an assistant for e-mail tasks, the E-mail Assistant includes an axiom that can derive e-mail addresses from phone numbers, by finding a person with the given phone number, and then obtaining the e-mail address of that person." Drop Zones at p. 61. "Where does the E-mail Assistant find the set of people whose phone numbers and e-mail addresses can be examined? The Drop Zones representational system provides two ways through which an assistant can gain access to this information. Mappings can be built between the objects inside the Drop Zones representational system (e.g., there is an object called PERSON, which has such attributes as PhoneNumber and EmailAddress) and databases or other applications. Such a mapping, in combination with a scripting language or some other programmatic way of manipulating applications, enables the E-mail Assistant to look inside an address book application for a person with the stated phone number." Drop Zones at p. 61. "A representation of semantics can be useful in other ways. Consider the problem of trying to invoke an action on a collection of differing terms in a document. For instance, the sender of the e-mail message in Figure 3, based on a human's reading of the message, is clearly interested in getting together for lunch. LiveDoc has identified a number of structures in the e-mail message, which, as usual, are shown with colored highlights. However, the bits of information that make up the details of this proposed meeting are spread throughout the message, and, as a result, they fail to match the rigid syntactic 'Meeting' grammar built into LiveDoc. Drop Zones goes beyond LiveDoc in allowing the user to select some subset of those terms and drag them as a group to the meeting assistant for interpretation. Because the Meeting Assistant contains an axiom

Exhibit H

	specifying that a meeting is indicated by a date, a time, a venue, and a person's name, this collection of objects is recognized as a meeting by the Meeting Assistant. This assistant will then highlight itself when the objects are dragged over it, and an action like [Add this meeting to your calendar' can be offered to the user via the Assistant window." Drop Zones at p. 62. "FIG. 4 illustrates an example of an analyzer server 220, which includes grammars 410 and a string library 420 such as a dictionary, each with associated actions. One of the grammars 410 is a telephone number grammar with associated actions for dialing a number identified by the telephone number grammar or placing the number in an electronic telephone book. Analyzer server 220 also includes grammars for post-office addresses, e-mail addresses and dates, and a string library 420 containing important names. When analyzer server 220 identifies an address using the e-mail address' grammar, actions for sending e-mail to the identified address and putting the identified address in an e-mail address book are linked to the address." Miller at 5:6-5: 18. *See also* Miller at FIG. 4 at 420 ("write letter" and "retrieve #"); FIG. 4 at 410 ("write letter" and "[p]ut in electronic calendar"); 4:58-6:18. *See also* Video at 1:27-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. If a user selects the option "Add e-mail address to Emailer address book," the system would add the email address to the Claris Emailer address book. *See, e.g.*, Drop Zones at 61 ("After the user drops the name on the E-mail Assistant, a set of actions that make sense for people are presented in the Assistant window."). *See, e.g.*, Drop Zones at 60 ("[T]hinking about the name of a person and a phone number … from the perspective of an address book easily leads to the interpretation, 'Add this person to my address book'.").

Exhibit H

	See, e.g., Drop Zones at 61 (“Alternatively, these facts can be held directly within the Drop Zones representational system, as relations of terms, such as: (PHONE-NUMBER ‘Tom Bonura’ 974-4538) These terms can be provided by developers or, through an appropriate human interface, end-users.”). When a user selects a detected structure, the system prompts the user to select a candidate action by displaying a pop-up menu of actions. *See, e.g.,* 4:27-31 ("Upon selection of this structure, user interface 240 presents and enables selection of the linked candidate actions using any selection mechanism, such as a conventional pull-down or pop-up menu."). As shown in Fig. 4, several of these actions update an information source to include first information (e.g., put in address book). *See, e.g.,* Fig. 4; 5:6-18. "As shown in FIG. 7, upon recognition of a mouse-down operation over a structure, user interface 240 presents a pop-up menu 710. In this example, pop-up menu 710 displays the candidate actions linked to the selected telephone number grammar 410, including dialing the number and putting the number into an electronic telephone book." 5:38-43. *See also* Video at 1:15-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 5. For example, LiveDoc Version 0.8 System and the prior art references and systems set forth in Exhibit U, Table 5 (e.g., Miller, Pandit) relate to managing information in an address book/contact database. A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc Version 0.8 System and the prior art references and systems in Exhibit U. A POSITA would have recognized advantages and benefits to have LiveDoc Version 0.8 System provide a prompt for updating the information source to include the first information.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. *See* claim 1.

Exhibit H

	For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. *See* Claim 1. As shown in Fig. 2 of Drop Zones, when a user drags and drops a person onto the Email Assistant they are provided a list of email related actions to select from, including "send email." *See also* Drop Zones at 61. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7 (*e.g.*, Luciw 735, Siitonen, Domini, and Schabes). A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc Version 0.8 System and the prior art references and systems in Exhibit U, Table 7. For example, a POSITA would have recognized advantages and benefits to have LiveDoc Version 0.8 System display the plurality of distinct instances of second information from search results to enable user selection of one of them for use in performing the action.
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. *See* Claim 1. Now Contact, Claris Works and Claris Emailer were available on the same computer as LiveDoc Version 0.8. *See, e.g.*, Drop Zones at 61 ("Another call to the address book application, guided by another mapping rule, will return the email address for the identified person."). *See also* Video at 1:27-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	

Exhibit H

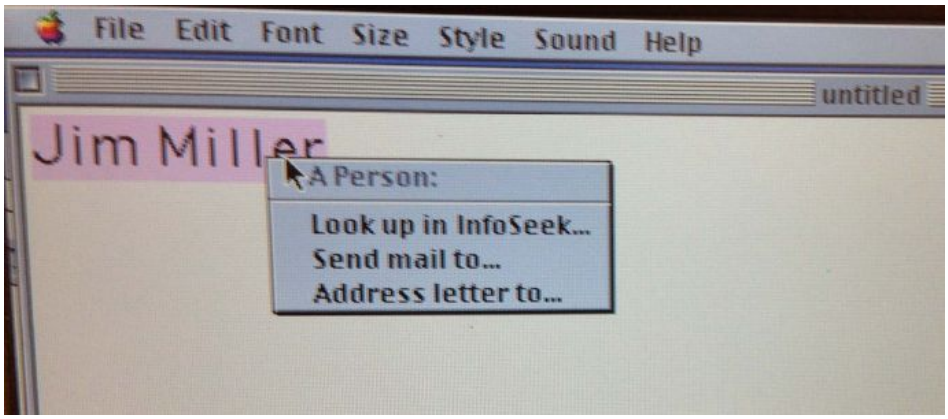
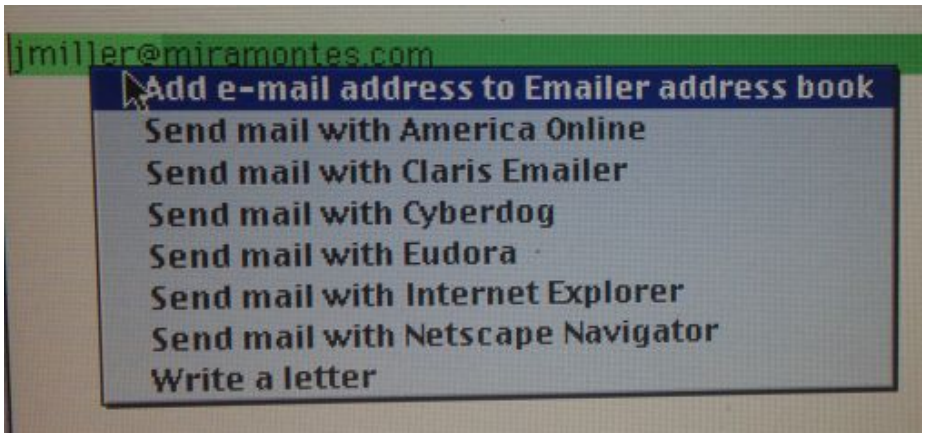
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. If a user types a name in the LiveSimpleText document, pressing the Option key would highlight the name, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail to" would cause the system to retrieve an email address for the selected name from the Now Contact database, and the system would then create a new email for that address. Selecting the option "Address letter to" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address. If a user types an email address in the LiveSimpleText document, pressing the Option key would highlight the email address, and right-clicking the mouse would pull up the following options: 

Exhibit H

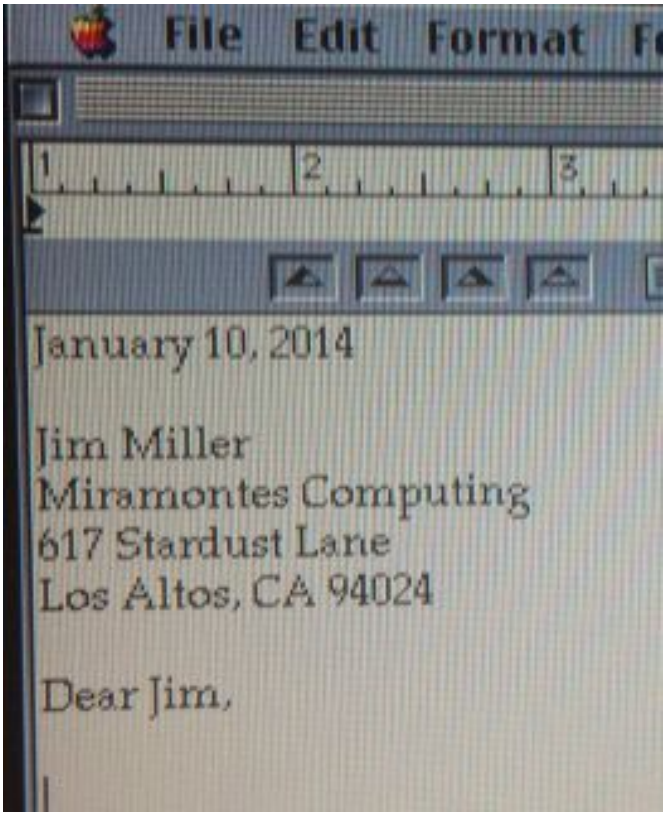
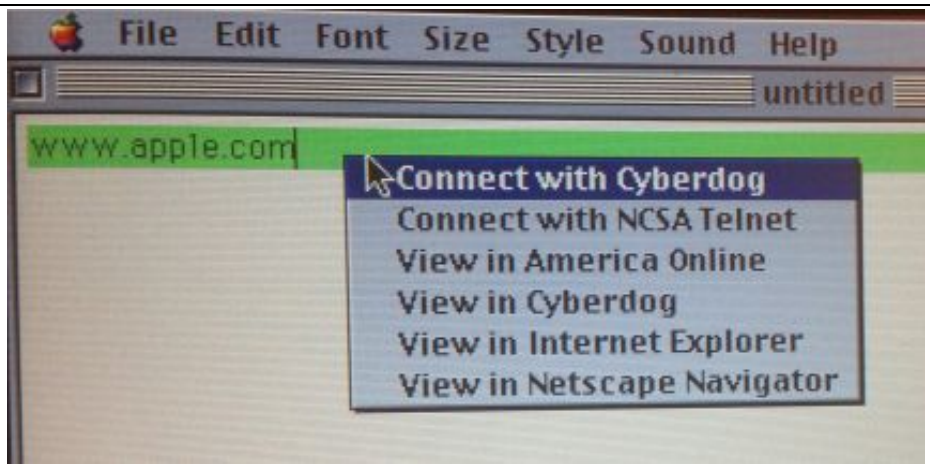
	These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail with Claris Emailer" would cause the system to retrieve an email address for the selected name from the Claris Emailer address book, and the system would then create a new email for that address in Claris Emailer. The email would include a "Destination" field which would display company names for certain URL domains. Selecting the option "Write a letter" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address using Claris Works. For example:  If a user types a URL in the LiveSimpleText document, pressing the Option key would highlight the URL, and right-clicking the mouse would pull up the following options:

Exhibit H



These options, when selected, would cause various actions to occur. For example, selecting the option "View in Netscape Navigator" would allow the user to view the website associated with that URL.

See, e.g., LiveDoc at 57 (referring to processors); Fig. 2 (illustrating a screen from an Apple computer)); LiveDoc at 53 ("There is a real opportunity to advance the computing field here, by bringing these two worlds together: by enabling an ordinary document, built with any application, to automatically offer users access to some of the meaningful bits of its content, and by helping users carry out appropriate actions on these objects."); at 58 ("Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."); Drop Zones at 61 ("Another call to the address book application, guided by another mapping rule, will return the email address for the identified person.").

See, e.g., LiveDoc at 57 ("However, other styles of interaction exist: Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application.").

"Consider the situation in which a user drags a telephone number to the E-mail Assistant, presumably so that the user can send an e-mail message to the person who possesses that phone number. A literal examination of the object reveals nothing of use to the assistant: sending e-mail messages requires an e-mail address, not a phone number. However, as part of its design as an assistant for e-mail tasks, the E-mail Assistant includes an axiom that can derive e-mail addresses from phone numbers, by finding a person with the given phone number, and then obtaining the

Exhibit H

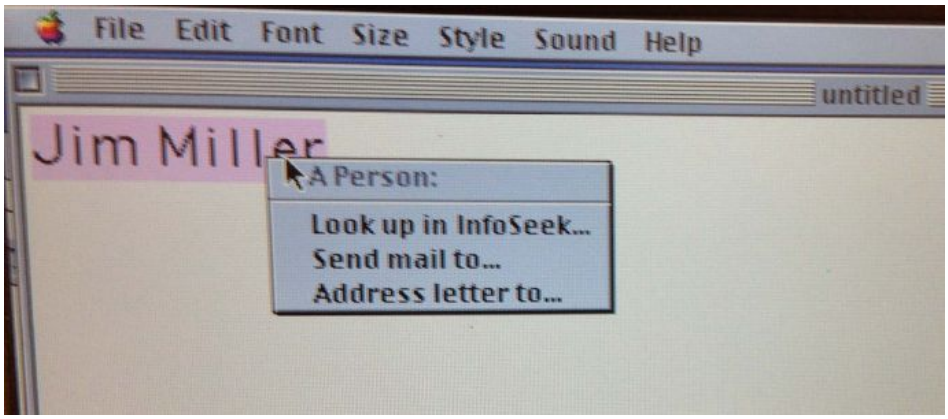
	e-mail address of that person." Drop Zones at p. 61. *See also* Video at 1:27-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc Version 0.8 System and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have LiveDoc Version 0.8 System cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	LiveDoc Version 0.8 System discloses claim 1. *See* claim 1. LiveDoc Version 0.8 System further discloses this element. If a user types a name in the LiveSimpleText document, pressing the Option key would highlight the name, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail to" would cause the system to retrieve an email address for the selected name from the Now Contact database, and the system would then create a new email for that address. Selecting the option "Address letter to" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address.

Exhibit H

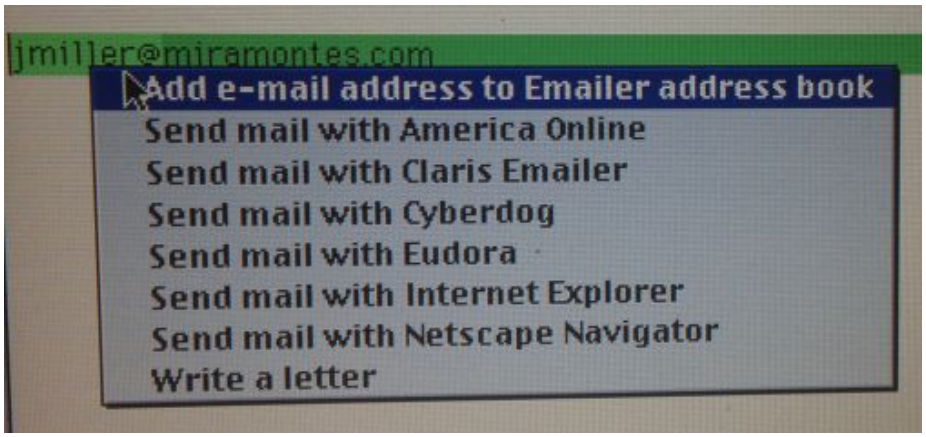
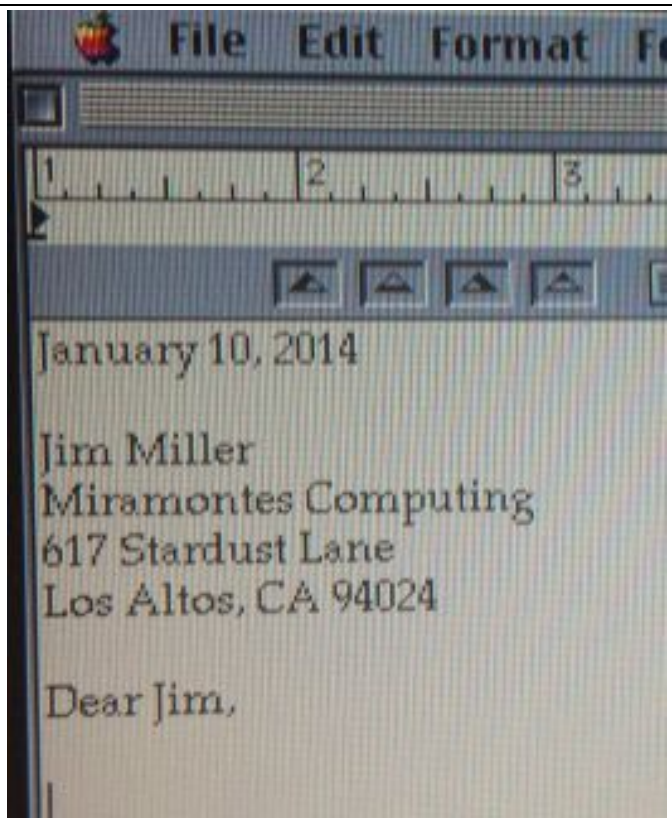
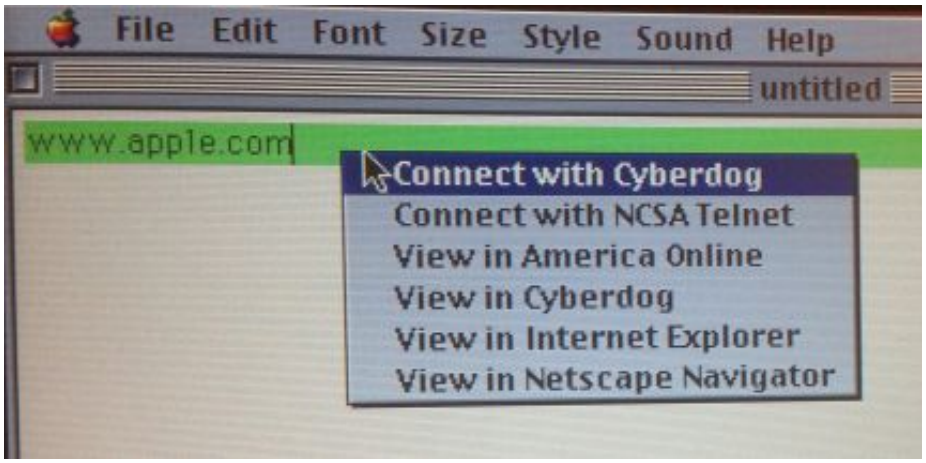
	If a user types an email address in the LiveSimpleText document, pressing the Option key would highlight the email address, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail with Claris Emailer" would cause the system to retrieve an email address for the selected name from the Claris Emailer address book, and the system would then create a new email for that address in Claris Emailer. The email would include a "Destination" field which would display company names for certain URL domains. Selecting the option "Write a letter" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address using Claris Works. For example:

Exhibit H



If a user types a URL in the LiveSimpleText document, pressing the Option key would highlight the URL, and right-clicking the mouse would pull up the following options:



These options, when selected, would cause various actions to occur. For example, selecting the option "View in Netscape Navigator" would allow the user to view the website associated with that URL.

See, e.g., LiveDoc at 57 ("However, other styles of interaction exist:

Exhibit H

	Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."). *See, e.g.*, LiveDoc at 57 (referring to processors); Fig. 2 (illustrating a screen from an Apple computer)); LiveDoc at 53 ("There is a real opportunity to advance the computing field here, by bringing these two worlds together: by enabling an ordinary document, built with any application, to automatically offer users access to some of the meaningful bits of its content, and by helping users carry out appropriate actions on these objects."); at 58 ("Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."); Drop Zones at 61 ("Another call to the address book application, guided by another mapping rule, will return the email address for the identified person."). For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc Version 0.8 System and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have LiveDoc Version 0.8 System cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	To the extent the preamble is limiting, LiveDoc Version 0.8 System discloses the preamble. A LiveDoc Version 0.8 System included a computer containing a program (on its hard disk drive and in system memory) for performing the recited method steps. LiveDoc Version 0.8 System included LiveSimpleText. "LiveSimpleText is a version of SimpleText that has been modified to use the LiveDoc API and, thereby, the Structure Detectors background application. When 'LiveDoc' mode is activated (via the Edit menu), the text shown in the window is analyzed by Structure Detectors to find meaningful patterns and phrases." Read Me File. *See, e.g.*, LiveDoc at 57 (referring to processors); Fig. 2 (illustrating a

Exhibit H

	screen from an Apple computer)); LiveDoc at 53 ("There is a real opportunity to advance the computing field here, by bringing these two worlds together: by enabling an ordinary document, built with any application, to automatically offer users access to some of the meaningful bits of its content, and by helping users carry out appropriate actions on these objects."); at 58 ("Imagine a detector that finds the formula of an organic molecule in a document, and an action that presents a three-dimensional rendering of that molecule within the context of the document itself, rather than in a separate application."); Drop Zones at 61 ("Another call to the address book application, guided by another mapping rule, will return the email address for the identified person."). *See, e.g.*, Miller at Fig. 1, at 170. *See also* Video at 0:30-1:40.
displaying the document electronically using the first computer program;	LiveDoc Version 0.8 System discloses this element. *See* claim 1. LiveSimpleText was a word processing application that allowed users to enter text. *See also* Video at 0:00-0:29.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	LiveDoc Version 0.8 System discloses this element. *See* claim 1. Holding down the Option key would result in "patterns and phrase" being highlighted in color. Read Me File. The patterns and phrases that could be identified included: email addresses, URLs, FTP file specifications, America Online keywords, Internet host names, Internet newsgroups, names of people, names of companies, and strings specified in the People, Companies, and Strings filed in the System Folder:Preferences:LD Dictionary Prefs folder. Read Me File.
retrieving the first information;	LiveDoc Version 0.8 System discloses this element. *See* previous row. *See* claim 1.
providing an input device, configured by the first computer program, that allows a user to enter	LiveDoc Version 0.8 System discloses this element. *See* claim 1.

Exhibit H

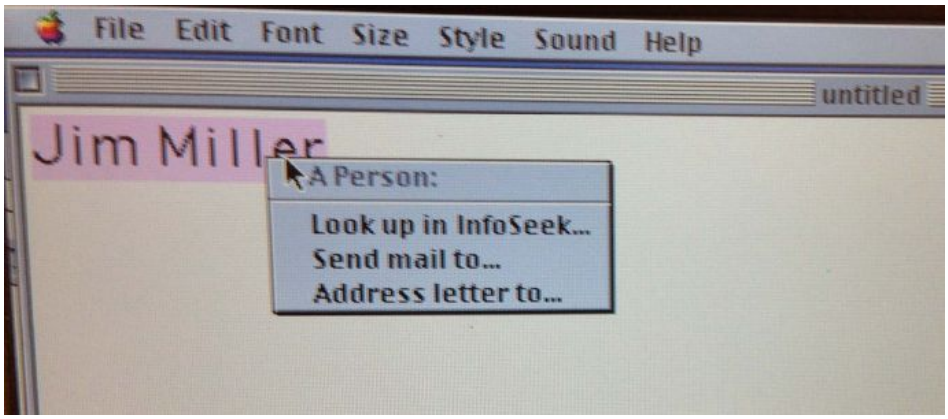
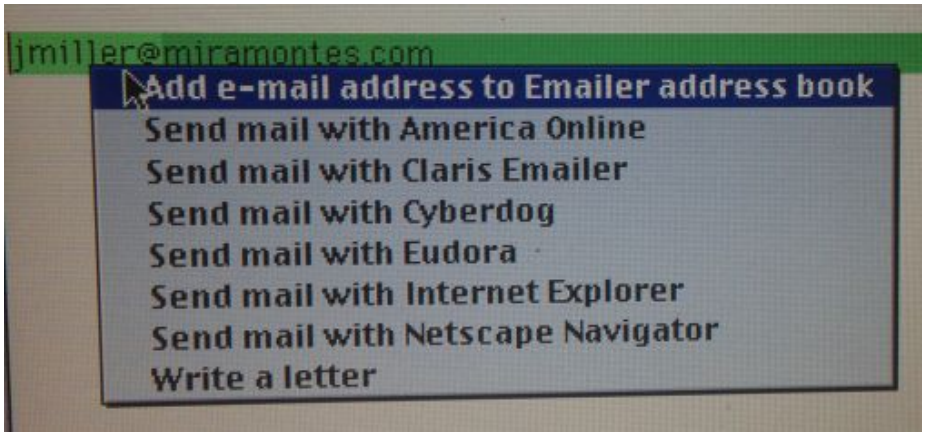
a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	If a user types a name in the LiveSimpleText document, pressing the Option key would highlight the name, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail to" would cause the system to retrieve an email address for the selected name from the Now Contact database, and the system would then create a new email for that address. Selecting the option "Address letter to" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address. If a user types an email address in the LiveSimpleText document, pressing the Option key would highlight the email address, and right-clicking the mouse would pull up the following options:  These options, when selected, would cause various actions to occur. For example, selecting the option "Send mail with Claris Emailer" would cause the system to retrieve an email address for the selected name from

Exhibit H

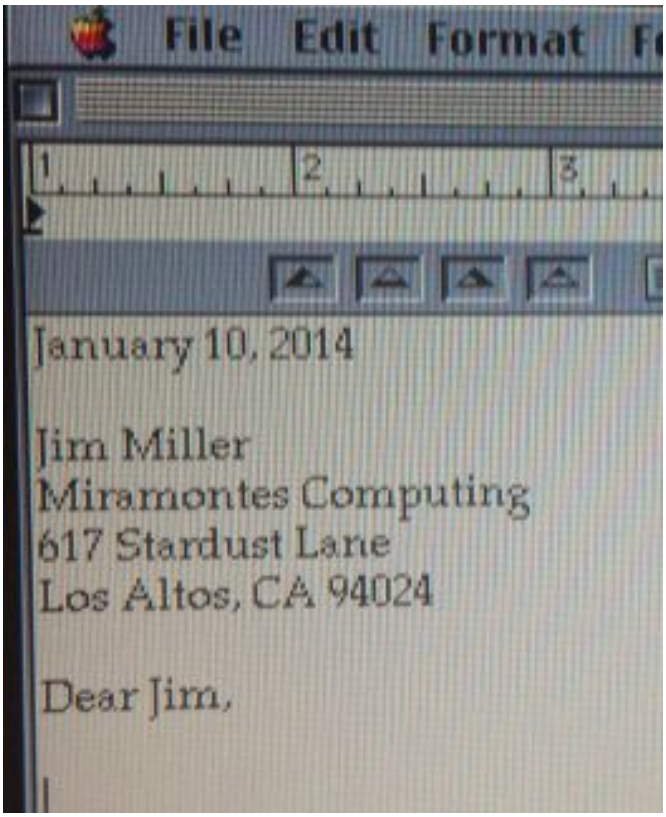
	the Claris Emailer address book, and the system would then create a new email for that address in Claris Emailer. The email would include a "Destination" field which would display company names for certain URL domains. Selecting the option "Write a letter" would cause the system to retrieve a mailing address for the selected name from the Now Contact database, and the system would then create a letter containing that address using Claris Works. For example:  If a user types a URL in the LiveSimpleText document, pressing the Option key would highlight the URL, and right-clicking the mouse would pull up the following options:

Exhibit H

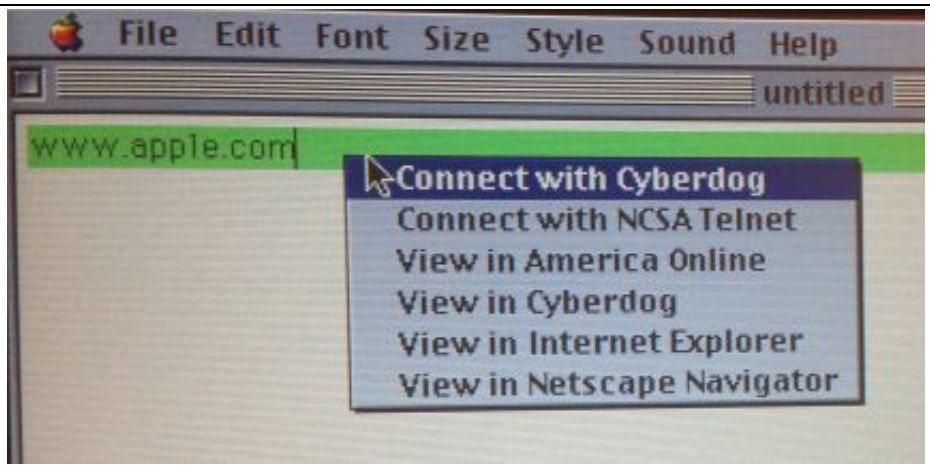
	 These options, when selected, would cause various actions to occur. For example, selecting the option "View in Netscape Navigator" would allow the user to view the website associated with that URL. *See also* Video at 1:27-1:40. LiveDoc version 0.8 System and the prior art references and systems set forth in Exhibit U, Table 8 (e.g., Pandit), relate to the same field of art, which is communication devices and methods. Specifically, LiveDoc version 0.8 System and certain prior art references and systems set forth in Exhibit U, Table 8 (e.g., Pandit) relate to the human-computer interaction of recognizing certain types of text as contact information in an electronic document. A POSITA would have reasonable expectation of success to achieve predictable results in combining LiveDoc version 0.8 System and the prior art references and systems in Exhibit F. A POSITA would have recognized advantages and benefits to have LiveDoc version 0.8 System to configure a input device in its first computer program. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	LiveDoc Version 0.8 System discloses this element. *See* previous row. *See* claim 1.

Exhibit H

if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	LiveDoc Version 0.8 System discloses this element. *See* previous row. *See also* Video at 1:27-1:40. *See* claim 1.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising:	To the extent the preamble is limiting, LiveDoc Version 0.8 System discloses the preamble. *See* claim 23.
providing a prompt for updating the information source to include the first information.	LiveDoc Version 0.8 System discloses this element. *See* Claim 8. If a user selects the option "Add e-mail address to Emailer address book," the system would add the email address to the Claris Emailer address book. *See also* Video at 1:15-1:40. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.

Exhibit I

Claim Chart Applying Newton MessagePad 2000 System Against the '843 Patent

The Apple Newton MessagePad 2000 handheld device ("Newton") was offered for sale, sold, and/or publicly used in the United States by at least March 1997.[1] It therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(b). As shown below, Newton anticipates and/or renders obvious claims 1, 8, 13, 15, 17, 18, 19, 23 and 30 of the '843 patent.

"Obviousness Statement" - To the extent that the Judge or Jury finds that Newton does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

Evidence of the availability of Newton include the following:

- "Apple Ships $1,000 MessagePad 2000," Washingtonpost Newsweek Interactive (March 24, 1997).
- "Apple Unveils Super Newton New Messagepad More Powerful, Multifunctional," New Orleans Times Picayune (October 29, 1996).
- "Apple says initial MessagePad sales brisk," Reuters (April 20, 1997).
- Joe Hutsko, "Treading Lightly in a Sea of Hand-Held Computers," New York Times on the Web: Technology | Cybertimes (Aug. 21, 1997).

Evidence of the design and operation of Newton include the following:

- Apple Newton, including source code for versions 2.0, 2.1, and 2.3.
- "MessagePad 2000 User's Manual," Apple Computer, Inc. (1997) ("Newton Manual").
- "Newton Programmer's Guide: For Newton 2.0," Apple Computer, Inc. (1996) ("Newton Guide").
- "Newton Programmer's Guide: 2.1 OS Addendum," Apple Computer, Inc. (1997) ("Newton 2.1 Addendum").

[1] Defendants reserve the right to rely on earlier versions of the Apple Newton that have similar functionality, including but not limited to the MessagePad 100, MessagePad 110, MessagePad 120, and MessagePad 130.

Exhibit I

- "Newton 2.0 User Interface Guidelines," Apple Computer, Inc. ISBN 0-201-48838-8 (first printing May 1996) ("Newton Guidelines").
- MessagePad 2000 with Newton 2.1 Operating System Datasheet, Apple Computer, Inc. (1997) ("Newton Datasheet").
- Newton Programmer's Reference, Apple Computer, Inc. (1996).
- Apple MessagePad Handbook, Apple Computer, Inc. (1995).
- Newton MessagePad 2000 handheld device, available for inspection at DLA Piper LLP (US).
- Photographs of MessagePad 2000, contained below.

Further evidence of the availability, design, and operation of Newton is contained in the deposition of Giulia Pagallo ("Pagallo Depo.").

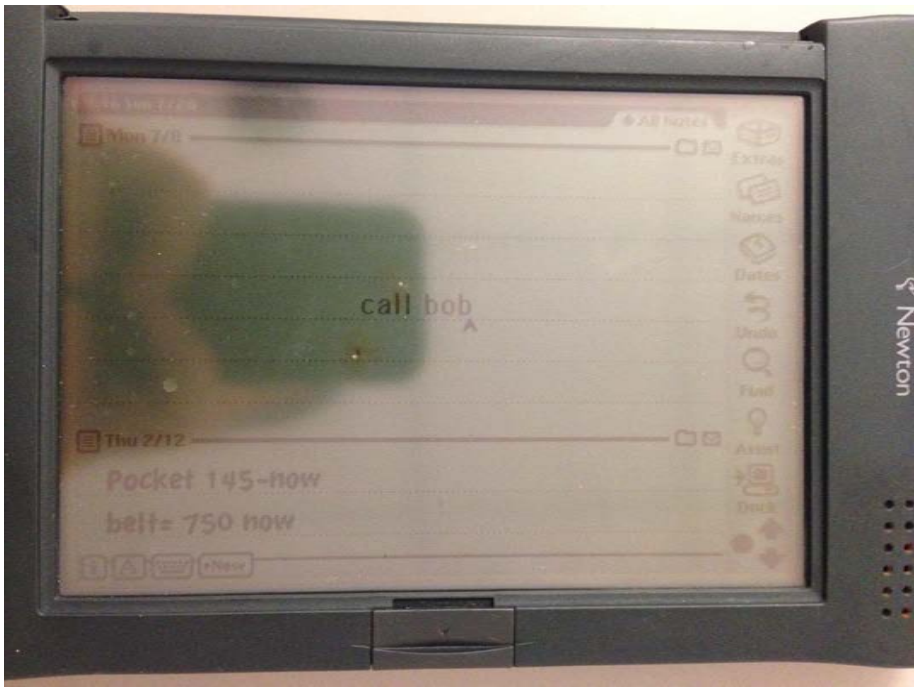
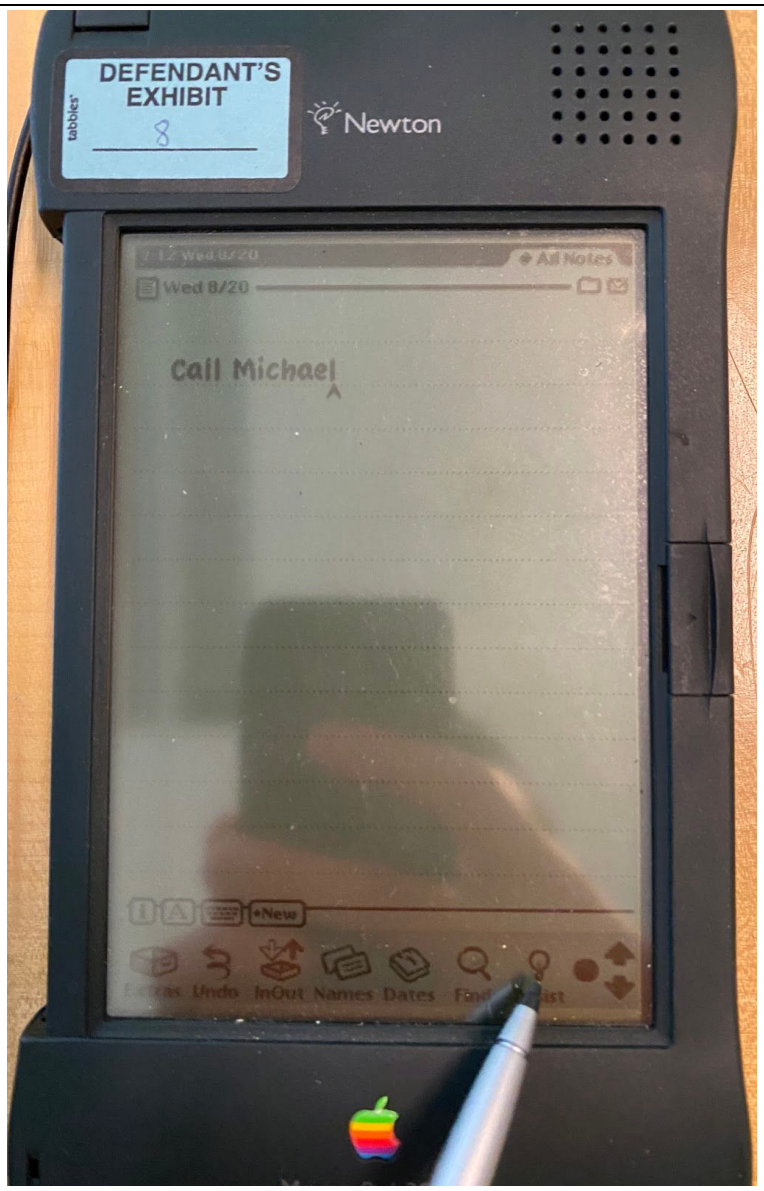
'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, Newton discloses the preamble. For example, the Newton implements a method for information handling within a document created by the Notes application.  *See also* screenshots taken during my Inspection.

Exhibit I



Newton Manual states:

"The Notepad is like a long roll of paper. It always contains at least one item. You can use it to write and draw notes and other items.

The Notepad is initially set as the default application, or backdrop, that you see when you close all other applications. When the Notepad is the backdrop, it is always open, though it may be hidden underneath other things. To see the Notepad, you must close everything else by tapping all the X's you see."

Newton Manual, p. 35.

Exhibit I

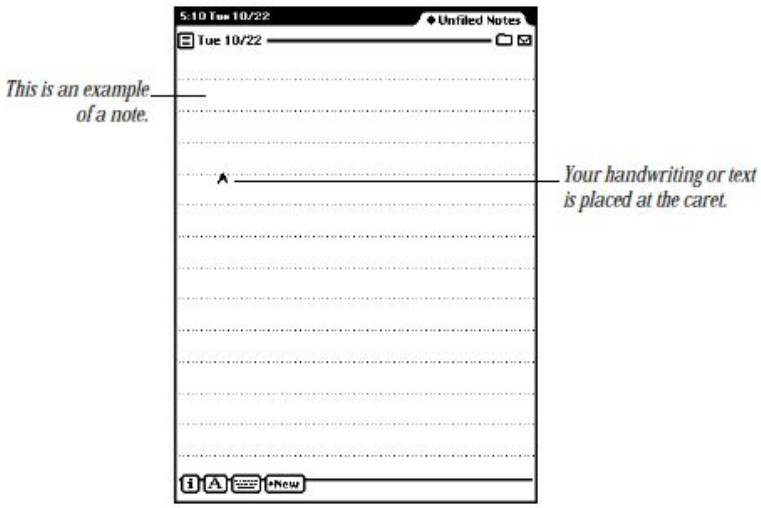
	"Creating a note Tap the Recognition button [A] and from the list that appears, tap your choice—Text, Ink Text, Shapes, or Sketches. If you are writing, tap the pen where you want your words to appear on the screen. A caret appears. You can now write anywhere on the screen and the information is placed at the caret. If you are drawing, your drawings appear wherever you draw them on the screen." Newton Manual, p. 37.  Newton Manual, p. 37. "Dialing telephone numbers You can place a telephone call while you are in the Name File. You can also tap Assist or use the Calls application in the Extras Drawer to make calls." Newton Manual 65. "You can organize the information in your MessagePad by filing things in folders that you can create and name. You can also use your MessagePad to find any text information that is stored in the internal memory of your MessagePad or on a storage card." Newton Manual, p. 159. "Finding information

Exhibit I

	The MessagePad makes it easy to find specific pieces of information no matter where you have filed them." Newton Manual, p. 164. "You can have the MessagePad automatically perform certain tasks for you using Assist. The tasks that can be performed include sending faxes or electronic mail, dialing telephone numbers, scheduling, setting the time and date, finding text, printing, and entering items in your To Do list. You can also tap Assist to access on-screen help." Newton Manual, p. 193. Newton Guide states[2]: "This chapter describes how your application can support finding text, dates, or your own data types in its data. If you want users to be able to use the system's Find service to locate data in your application, you should be familiar with the material discussed in this chapter. * * * The Find service searches for occurrences of data items the user specifics on a Find slip. The Find slip may be supplied by the system or by the developer. Figure 16-1 illustrates the system-supplied Find slip." Newton Guide, p. 16-1. "The Intelligent Assistant is a system service that attempts to complete actions specified by the user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can use the built-in applications to complete predefined tasks such as calling, faxing, printing, scheduling, and so on. You can also program the Assistant to execute any task that your application performs. In addition, you can display your application's online help from the Assistant. This chapter describes how to make application behaviors and online help available from the Assistant. If you want to provide a textual interface to your application or its online help, you should become familiar with the Assistant and the concepts discussed in this chapter.

[2] The Newton MessagePad 2000 handheld device ran version 2.1 of the Newton operating system. *See, e.g.* Newton Manual at p. 253. Version 2.1 was substantially similar to version 2.0 of the Newton operating system in the relevant respects. For example, the Notes, Names and Intelligent Assistant feature operated in substantially the same manner in version 2.1 as in version 2.0. *See, e.g.* Newton 2.1 Addendum at p. xix ("This book describes the additions to the Newton operating system for version 2.1); *see id. generally.*

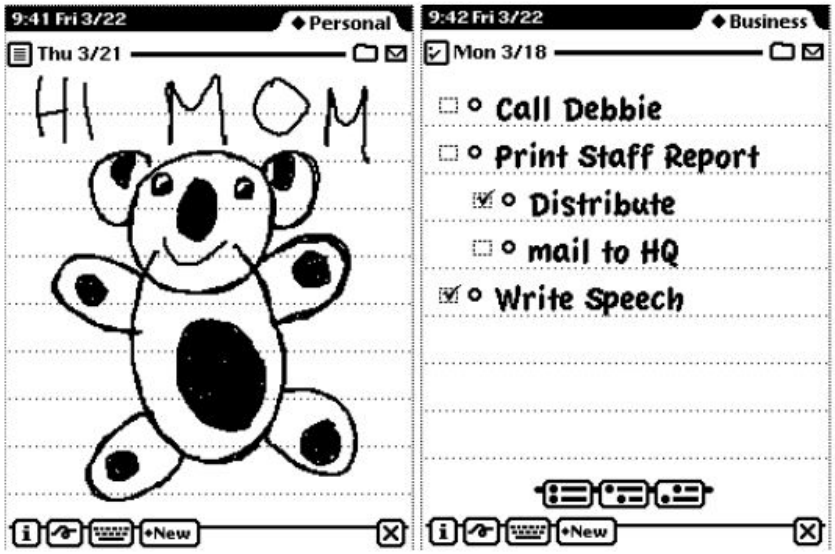
Exhibit I

	* * * When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input. The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities. The templates may be supplied by the system or by your application. Some system-supplied templates make use of lexical dictionaries, which are also supplied by the system. For more information about lexical dictionaries, see Chapter 9, 'Recognition.' Depending on the amount of information that parsing the input string provides, the Assistant either attempts to complete a task or prompts the user to supply additional information."" Newton Guide, pp. 18-1-18-2. "Names This section describes the application program interface (API) to the Names application. The Names application manages a database of people and places. It presents information either as a business card, or as a list of all the available information. These two views are shown in Figure 19-1." Newton Guide, p. 19-2. "About the Notes Application Notes is a simple application based on NewtApp that allows the user to create new stationery, scroll up and down, route and file notes, and scan an overview." Newton Guide, p. 19-31. "Notes This section describes the Notes API. The Notes application uses three types of stationery: regular notes, checklists, and outlines. Figure 19-6 shows a note and a checklist; an outline (not shown) is like the checklist without the checkboxes." Newton Guide, p. 19-30.

Exhibit I

Built-in Applications and System Data

Figure 19-6 Notes note and Checklist views



Newton Guide, p. 19-31.

"A key part of the Newton information architecture is the Intelligent Assistant. The Intelligent Assistant is a system service that attempts to complete actions for the user according to deductions it makes about the task that the user is currently performing. The Assistant is always instantly available to the user through the Assist button, yet remains nonintrusive.

The Assistant knows how to complete several built-in tasks; they are Scheduling (adding meetings), Finding, Reminding (adding To Do items), Mailing, Faxing, Printing, Calling, and getting time information from the Time Zones map. Each of these tasks has several synonyms; for example, the user can write 'call,' 'phone,' 'ring,' or 'dial' to make a phone call."

Newton Guide, p. 1-8.

See also Newton Guidelines at pp. 8-22-8-28: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs.

Exhibit I

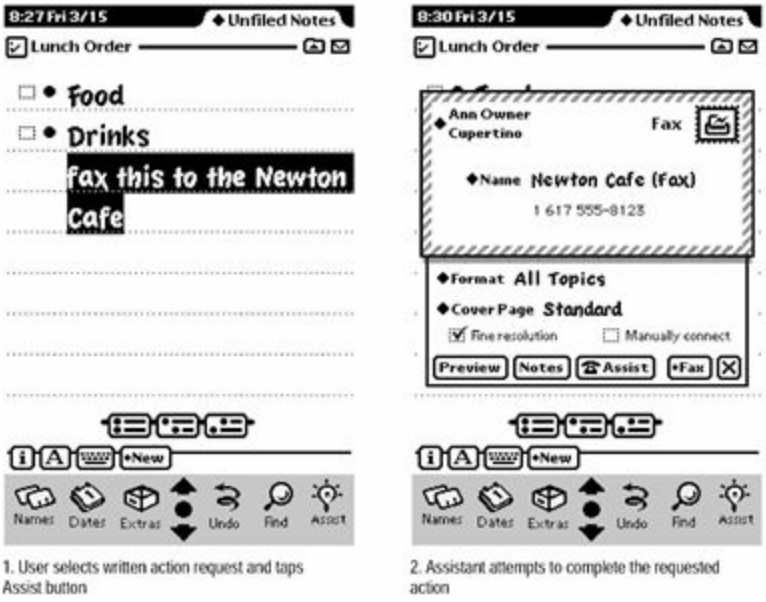
	Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly. Invoking the Assistant A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation. Figure 8-22 The Assist button makes the Assistant try a written action request  1. User selects written action request and taps Assist button 2. Assistant attempts to complete the requested action Interpreting the Request Phrase The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same

Exhibit I

	task if given the word ‘phone,’ ‘call,’ or ‘dial.’ Applications can add words to the Assistant’s lexicon, but users cannot. The Assistant matches words regardless of their capitalization. For example, it considers the word ‘phone’ to be the same as the word ‘Phone.’ The order in which the user writes words is not significant. For example, the phrase ‘Royce fax’ produces the same result as the phrase ‘fax Royce.’ This syntax-independent architecture allows easier localization of applications for international audiences. The Assistant ignores words it does not know, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as ‘Make a phone call to Bob at work’ and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words. Assist Slip If the Assistant can’t interpret a user’s written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip’s Please picker and can write on the slip’s input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote ‘Bob,’ the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words. [Figure 8-23] An Assist slip’s Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker. [Figure 8-24] The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your applications registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the

Exhibit I

	top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top portion of the Please picker. For example, the word 'call' appears but the synonyms 'ring' and 'phone' do not. Recently used synonyms may appear in the bottom half of the picker, however. If a user writes a verb the Assistant doesn't know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase 'buzz 555-1234' the Assistant does not match the word 'buzz' to an action, but it can identify '555-1234' as having the format of a telephone number. Based on that information, the Assistant deduces the user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker. In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see 'Help' on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action. The primary function of an Assist slip is to specify an action. IF the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip. Task Slips. Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user writes the request 'fax Bob,' the Assistant can ge Bob's fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page. The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date

Exhibit I

	Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86). Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way." "Q. What is the Intelligent Assist feature? A. Um, the Intelligent Assist feature is in the Newton, is a system service that was designed to facilitate interactions and user actions. Like, for instance, placing a call, sending an email, creating a reminder, so that the users could do that from, say, the Notes application without necessarily having to go into the calendar or the date, the Names application. Q. Or even going – A. The call. Q. – into the Call application. A. For instance, yes." Pagallo Depo. at 68:5-17. "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "Q. So if a user is within the Notepad application and the user has entered in text that says fax and then a phone number – A. Uh-huh. Q. – how can the user then invoke Intelligent Assist after entering in that text? What are the steps that need to be taken? A. Once the text is entered and is recognized, then the user can tap Assist." Pagallo Depo. at 73:19-74:2.

Exhibit I

	"What feature is recognizing the text on the – in the Notepad application for the Intelligent Assist feature to work? A. The handwriting recognition feature that we talked about earlier today." Pagallo Depo. at 74:10-14. "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "And if a contact with that name does not already exist, what would the user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right.

Exhibit I

	Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number? A. Yes." Pagallo Depo at 92:6-20. "Q. And then after the text is entered in, then the user presses the Intelligent Assist button, the Intelligent Assist feature is what is recognizing or the event – A. What is understanding the command. Q. Okay. So within Intelligent Assist, are there specific commands – only specific commands or actions that the Intelligent Assist feature can recognize? A. Yes." Pagallo Depo. at 75:1-9. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if a user taps the word Please, then it – um, is the user presented with different options for actions that could occur? A. The different option for action that could occur with a person's name." Pagallo Depo. at 85:5-9. "So underneath the action word, call, fax, find, mail, print, remember, schedule in time, the other items that appear under the Please Picker are the recent action –

Exhibit I

	A. I believe so." Pagallo Depo at 85:21-25. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh. Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * * A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work. Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4 "But if you do the selection, you – the Intelligent Assistant will act on the selected text only." Pagallo Depo. at 87:15-17. "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5.

Exhibit I

	"But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number? A. Yes." Pagallo Depo at 92:6-20. "And then is the user able to then save information to the Name File after making the call? A. Um, yes. * * * Yes. Sorry, my – I got sidetracked, but yes. So the same UI that we, um, happen before when we try the Jennifer Lopez case, happens in this case, after I entered the number, because, um, I didn't have a phone number So I had to enter the phone number. And once the phone number and I place the call, the UI to add it to the Names File is presented to the user. Q. Okay. And then so the user is presented with an option to add this new contact to the new file? Okay. A. Yes." Pagallo Depo at 92:21-93:11. "And then how does the user then add the – this new contact to the Name File? Is there an icon or button that's add or – A. There is a UI that has the name card with the name, the phone number, and it says, 'Would you like to add a new name with the information,' and then you can say – select yes or no. And if I select yes, then that new card is added to the Name File." Pagallo Depo at 93:12-19. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number

Exhibit I

	and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "So the screen that shows for the user that you were just describing – A. Uh-huh. Q. – where you can see the specifications that are being used for the phone call, does the user have the opportunity to change any of the information that's presented in the screen, so if the phone number – if the user realizes that this phone number for Bob is incorrect and wants to change the phone number that's being used, does the user have the opportunity to do that? A. Yes. The user has the opportunity to tap a name and select a different name or select a different entry." Q. Okay. And that's presented to the user as options before a call – A. Correct. Q. – is executed?" Pagallo Depo at 97:5-22. "And then if, in the instance that the information that's presented to the user for Bob is correct, and the user decides to place the phone call, what does the user need to do next after it's presented with the – the screen? A. Tap on the button that says Call." Pagallo Depo at 97:23-98:3. "So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up. Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12. "The – the system – the Intelligent Assistant was a framework that could be invoked through the application, but it was a – as a system service." Pagallo Depo at 100:16-18. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Newton discloses this element. For example, Newton includes a Notes application which displays a document including text entered by a user, as illustrated in the photograph above.

Exhibit I

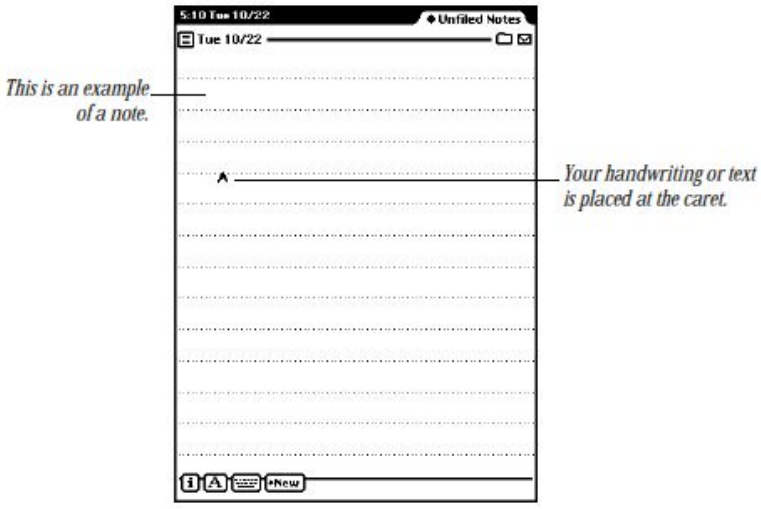
	Newton Manual states: "The Notepad is like a long roll of paper. It always contains at least one item. You can use it to write and draw notes and other items. The Notepad is initially set as the default application, or backdrop, that you see when you close all other applications. When the Notepad is the backdrop, it is always open, though it may be hidden underneath other things. To see the Notepad, you must close everything else by tapping all the X's you see." Newton Manual, p. 35. "Creating a note Tap the Recognition button [A] and from the list that appears, tap your choice—Text, Ink Text, Shapes, or Sketches. If you are writing, tap the pen where you want your words to appear on the screen. A caret appears. You can now write anywhere on the screen and the information is placed at the caret. If you are drawing, your drawings appear wherever you draw them on the screen." Newton Manual, p. 37.  Newton Manual, p. 37. Newton Guide states: "About the Notes Application

Exhibit I

Notes is a simple application based on NewtApp that allows the user to create new stationery, scroll up and down, route and file notes, and scan an overview."

Newton Guide, p. 19-31.

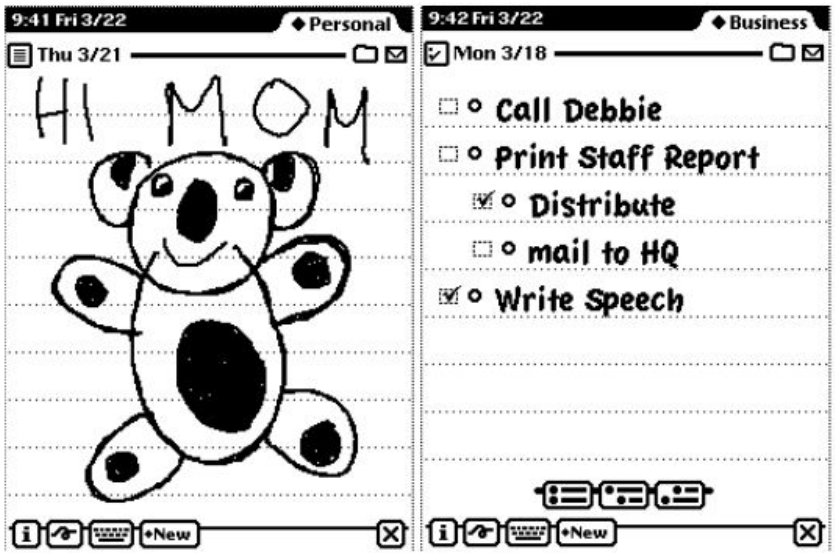
"Notes

This section describes the Notes API. The Notes application uses three types of stationery: regular notes, checklists, and outlines. Figure 19-6 shows a note and a checklist; an outline (not shown) is like the checklist without the checkboxes."

Newton Guide, p. 19-30.

Built-in Applications and System Data

Figure 19-6 Notes note and Checklist views



Newton Guide, p. 19-31.

"When a user writes a line of text on the Newton screen, the Newton system software performs a series of operations, as follows:

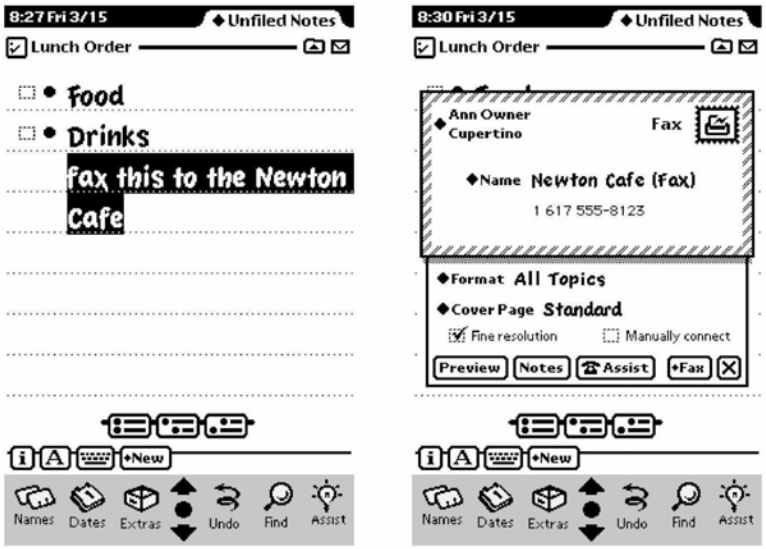
- The raw data for the input is captured as ink, which is also known as **sketch ink** or **raw ink**.
- Raw ink is stored as a sequence of **strokes** or stroke data.
- If the view in which the ink was drawn is configured for **ink text,** the recognition system groups the stroke data into a series of **ink words,** based on the timing and spacing of the user's

Exhibit I

	handwriting. A user can insert, delete, and move ink words in the same way as recognized text. Ink words can be scaled to various sizes for display and printing. They can also be recognized at a later time, by a process known as **deferred recognition.** ● If the view in which the ink was drawn supports or is configured for text recognition, the ink words are processed by the recognition system into recognized text and displayed in a typeface." Newton Guide, p. 8-2. See also Newton Guidelines at pp. 8-22-8-23: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly. Invoking the Assistant A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Exhibit I

Figure 8-22 The Assist button makes the Assistant try a written action request



1. User selects written action request and taps Assist button

2. Assistant attempts to complete the requested action

Interpreting the Request Phrase

The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same task if given the word ‘phone,’ ‘call,’ or ‘dial.’ Applications can add words to the Assistant’s lexicon, but users cannot.

The Assistant matches words regardless of their capitalization. For example, it considers the word ‘phone’ to be the same as the word ‘Phone.’

The order in which the user writes words is not significant. For example, the phrase ‘Royce fax’ produces the same result as the phrase ‘fax Royce.’ This syntax-independent architecture allows easier localization of applications for international audiences.

The Assistant ignores words it does not know, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as ‘Make a phone call to Bob at work’ and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words.

Exhibit I

	Assist Slip If the Assistant can't interpret a user's written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip's Please picker and can write on the slip's input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote 'Bob,' the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words. [Figure 8-23] An Assist slip's Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker. [Figure 8-24] The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your applications registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top portion of the Please picker. For example, the word 'call' appears but the synonyms 'ring' and 'phone' do not. Recently used synonyms may appear in the bottom half of the picker, however. If a user writes a verb the Assistant doesn't know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase 'buzz 555-1234' the Assistant does not match the word 'buzz' to an action, but it can identify '555-1234' as having the format of a telephone number. Based on that information, the Assistant deduces the user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker. In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see 'Help' on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action.

Exhibit I

	The primary function of an Assist slip is to specify an action. IF the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip. Task Slips. Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user writes the request 'fax Bob,' the Assistant can ge Bob's fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page. The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86). Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way." "Q. So if a user is within the Notepad application and the user has entered in text that says fax and then a phone number – A. Uh-huh. Q. – how can the user then invoke Intelligent Assist after entering in that text? What are the steps that need to be taken? A. Once the text is entered and is recognized, then the user can tap Assist." Pagallo Depo. at 73:19-74:2.

Exhibit I

	"The – the system – the Intelligent Assistant was a framework that could be invoked through the application, but it was a – as a system service." Pagallo Depo at 100:16-18. For example, during my Inspection of the Newton Device, I observed that the Notes application can contain a number of different "Notes" documents. In the screenshot below, the Notes application displays a document that includes the text "Call Michael" entered by a user.

Exhibit I

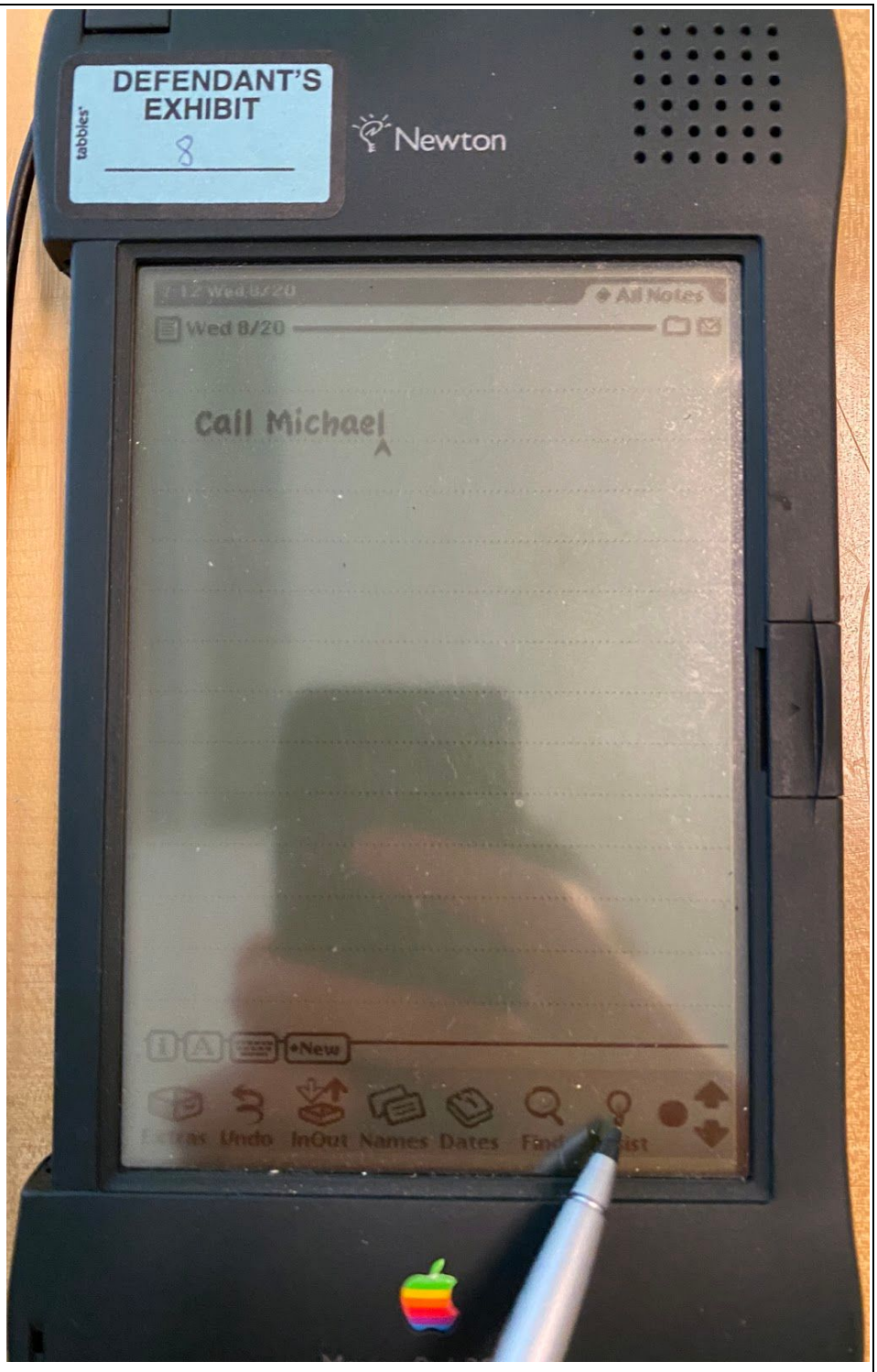
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.

Exhibit I

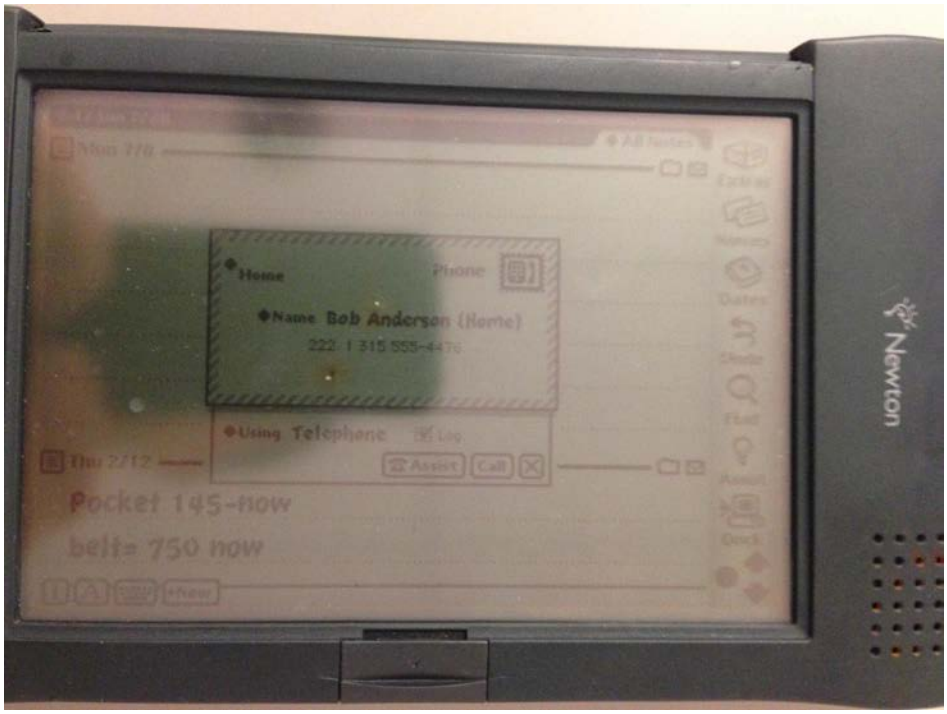
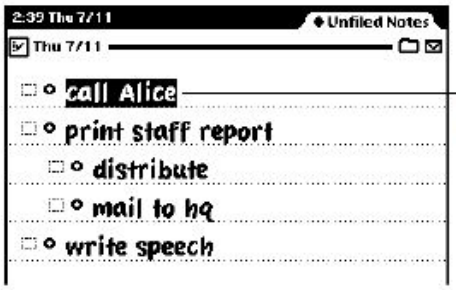
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Newton discloses this element. For example, tapping the "Assist" button launches the Intelligent Assist feature with text from the document. The photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application (as in the first photograph in the set above) and then tapped the "Assist" button. As illustrated in the photograph below, the full name and phone number associated with the name "bob" in the Names application is displayed.  Newton Manual states:

Exhibit I

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

3 Tap Assist.

The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.

4 Enter other necessary information.

5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call.

Newton Manual, p. 196.

"Using the correct request words

The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.)

Exhibit I

	● Call to dial a telephone number. Synonyms: phone, ring, dial ***Call Bob at home*** looks in the Name File to find Bob's home phone number, thenputs it in the call slip. ● Fax to fax the item on the screen. Synonyms: none ***Fax Anderson*** opens a fax slip with the name Anderson and Anderson's fax number filled in. …" Newton Manual, pp. 196-97. Newton Guide states: "A key part of the Newton information architecture is the Intelligent Assistant. The Intelligent Assistant is a system service that attempts to complete actions for the user according to deductions it makes about the task that the user is currently performing. The Assistant is always instantly available to the user through the Assist button, yet remains nonintrusive. The Assistant knows how to complete several built-in tasks; they are Scheduling (adding meetings), Finding, Reminding (adding To Do items), Mailing, Faxing, Printing, Calling, and getting time information from the Time Zones map. Each of these tasks has several synonyms; for example, the user can write 'call,' 'phone,' 'ring,' or 'dial' to make a phone call." Newton Guide, p. 1-8. "RAM-based frames are not persistent until they are saved in a data structure called a **soup**, which is an opaque object that provides a persistent, dynamic repository for data." Newton Guide, p. 11-3. "When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input. **The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities**." Newton Guide, p. 18-1.

Exhibit I

	"When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup. The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose." Newton Guide, p. 18-4. "To permit more natural interaction, the Assistant ignores words that do not appear in any registered template's lexicon. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as "Make a call to Bob at work" and ignores the others. For example, the words "call", "Bob" and "work" are meaningful to the Assistant because they appear in the lexicons of registered templates. (In this case, the templates are supplied and registered by the system.) On the other hand, because the words "a", "to", and "at" do not appear in the lexicons of any registered templates, they are not matched and are therefore ignored." Newton Guide, p. 18-9. See also Newton Guidelines at pp. 8-23-8-28: "Interpreting the Request Phrase The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same task if given the word 'phone,' 'call,' or 'dial.' Applications can add words to the Assistant's lexicon, but users cannot. The Assistant matches words regardless of their capitalization. For example, it considers the word 'phone' to be the same as the word 'Phone.' The order in which the user writes words is not significant. For example, the phrase 'Royce fax' produces the same result as the phrase 'fax Royce.' This syntax-independent architecture allows easier localization of applications for international audiences.

Exhibit I

	The Assistant ignores words it does not know, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as 'Make a phone call to Bob at work' and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words. Assist Slip If the Assistant can't interpret a user's written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip's Please picker and can write on the slip's input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote 'Bob,' the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words. [Figure 8-23] An Assist slip's Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker. [Figure 8-24] The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your applications registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top portion of the Please picker. For example, the word 'call' appears but the synonyms 'ring' and 'phone' do not. Recently used synonyms may appear in the bottom half of the picker, however. If a user writes a verb the Assistant doesn't know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase 'buzz 555-1234' the Assistant does not match the word 'buzz' to an action, but it can identify '555-1234' as having the format of a telephone number. Based on that information, the Assistant deduces the

Exhibit I

	user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker. In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see 'Help' on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action. The primary function of an Assist slip is to specify an action. IF the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip. Task Slips. Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user writes the request 'fax Bob,' the Assistant can ge Bob's fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page. The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86). Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way."

Exhibit I

	"So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "Q. So if a user is within the Notepad application and the user has entered in text that says fax and then a phone number – A. Uh-huh. Q. – how can the user then invoke Intelligent Assist after entering in that text? What are the steps that need to be taken? A. Once the text is entered and is recognized, then the user can tap Assist." Pagallo Depo. at 73:19-74:2. "What feature is recognizing the text on the – in the Notepad application for the Intelligent Assist feature to work? A. The handwriting recognition feature that we talked about earlier today." Pagallo Depo. at 74:10-14. "Q. And then after the text is entered in, then the user presses the Intelligent Assist button, the Intelligent Assist feature is what is recognizing or the event – A. What is understanding the command. Q. Okay. So within Intelligent Assist, are there specific commands – only specific commands or actions that the Intelligent Assist feature can recognize? A. Yes." Pagallo Depo. at 75:1-9. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number?

Exhibit I

	So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. Tapping the "Assist" button launches the Intelligent Assistant and causes a "Phone" window to appear. The "Phone" window includes the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael").

Exhibit I

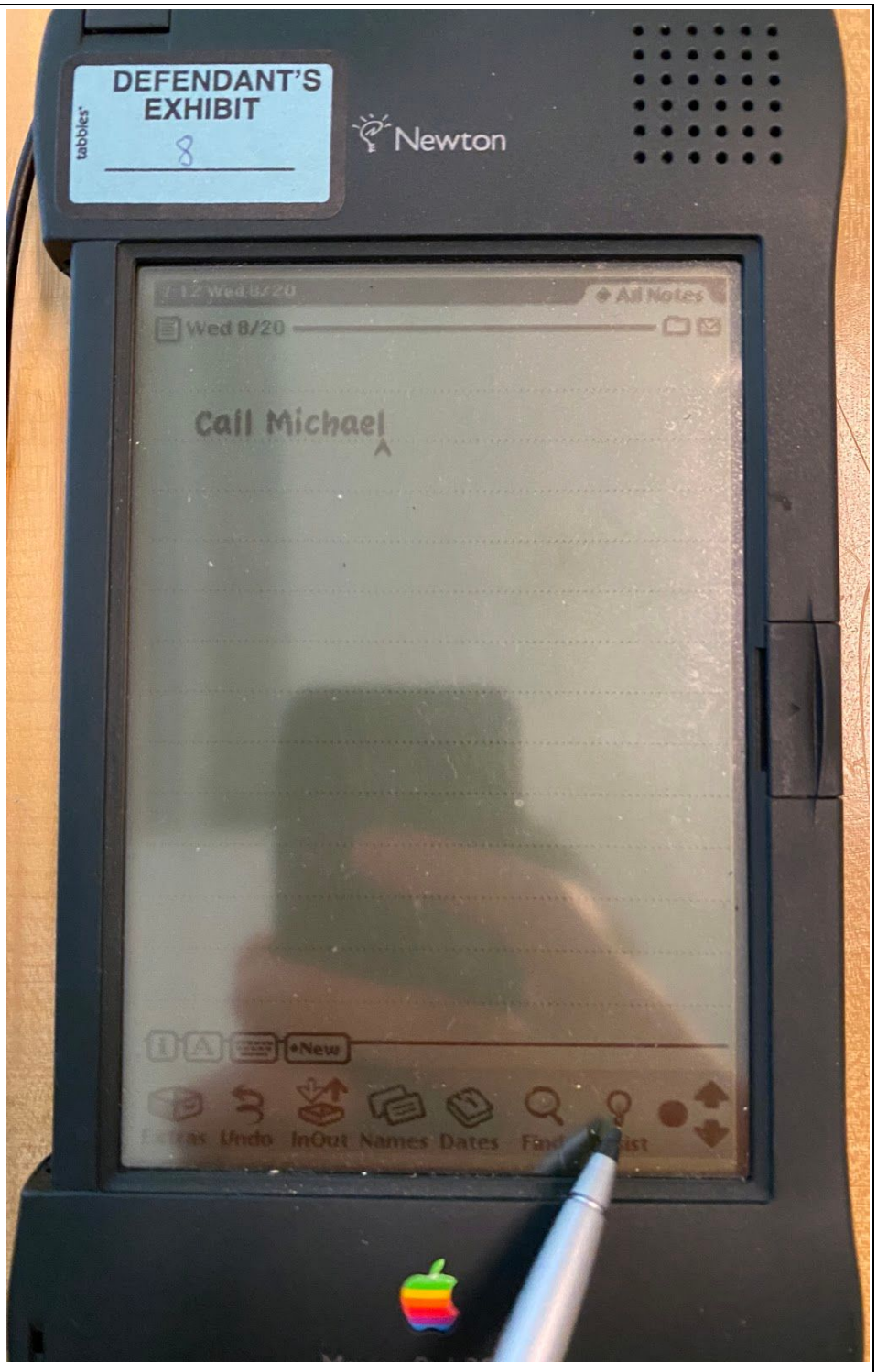
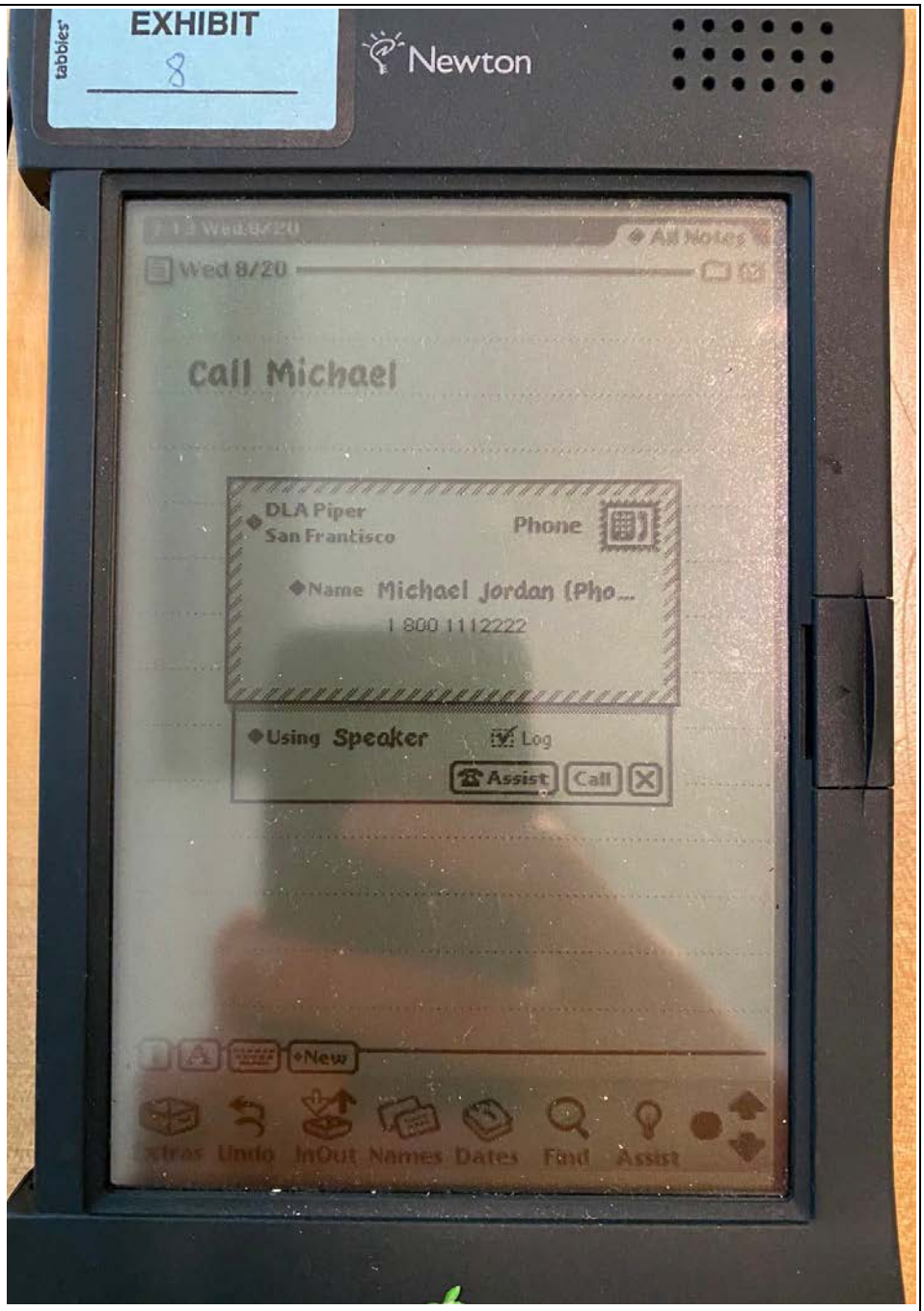

DEFENDANT'S
EXHIBIT
8
Newton
Wed 8/20
All Notes
Call Michael
New
Undo
InOut
Names
Dates

Exhibit I



Prior to the aforementioned sequence of events observed during my Inspection, a new contact card was added to include the first name "Michael," last name "Jordan," and phone number "800 1112222" to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card). The last name ("Jordan") and phone number ("1 800 1112222") was associated with the first name ("Michael") in this contact card. Indeed, prior to adding this contact card

Exhibit I

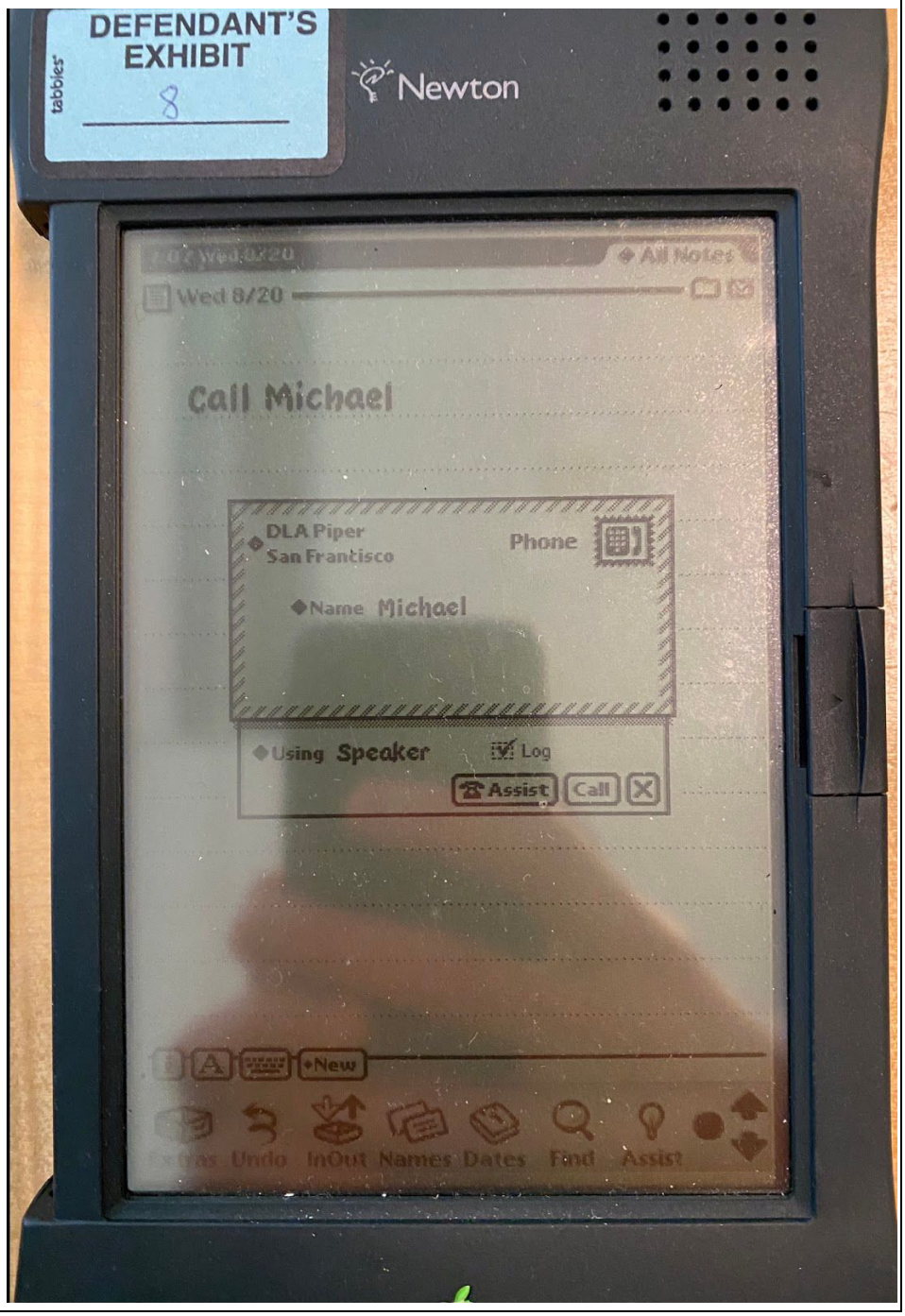
	to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card), the same aforementioned sequence of events (i.e. "Call Michael") did not result in the display of any other information associated with the name "Michael." 

Exhibit I

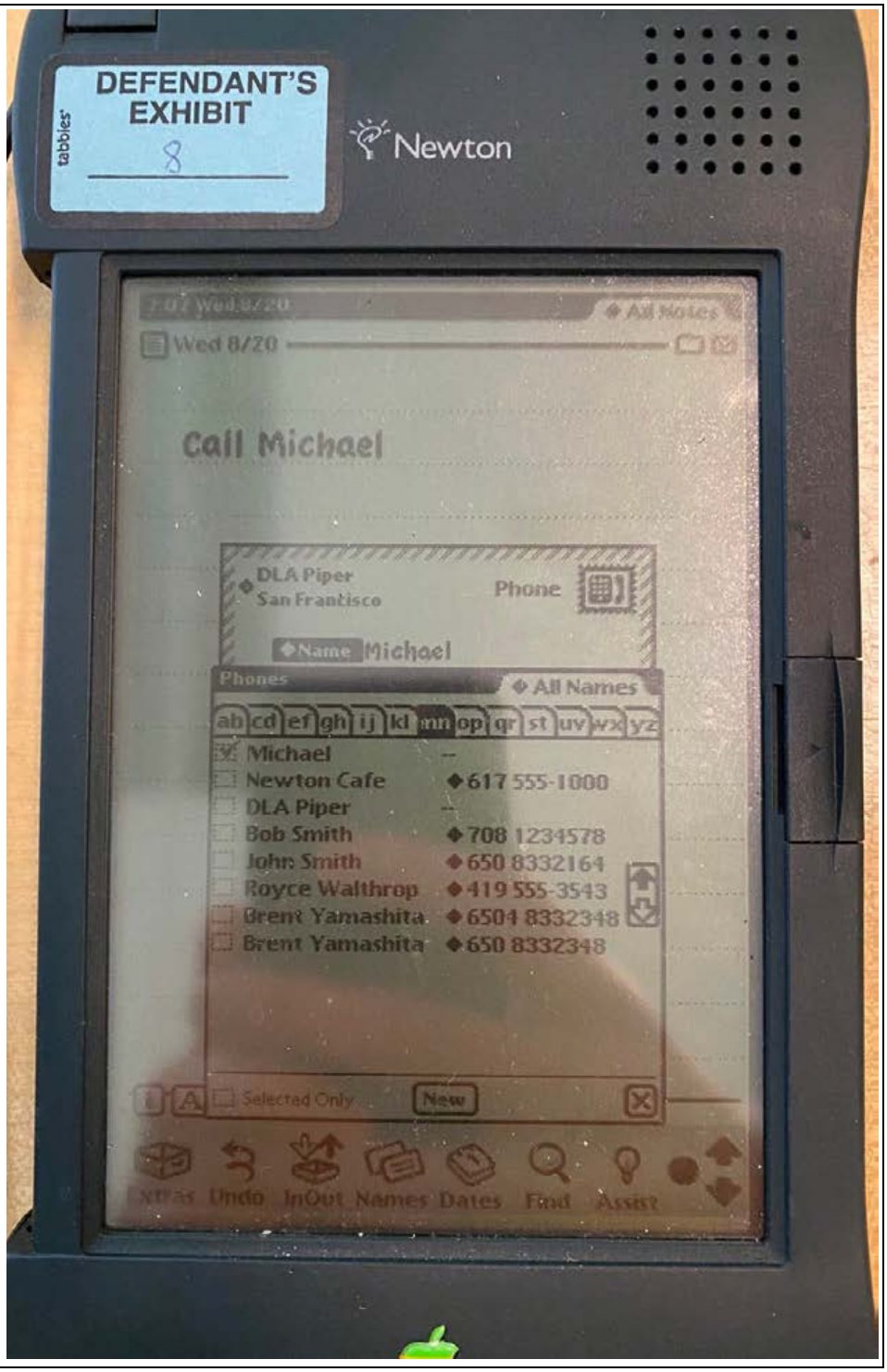

DEFENDANT'S
EXHIBIT
8
Newton
Call Michael
DLA Piper
San Francisco
Phone
Name Michael
Phones
All Names
Michael
Newton Cafe
617 555-1000
DLA Piper
Bob Smith
708 1234578
John Smith
650 8332164
Royce Walthrop
419 555-3543
Brent Yamashita
6504 8332348
Brent Yamashita
650 8332348
Selected Only
New
Undo
InOut
Names
Dates
Find
Assist

Exhibit I

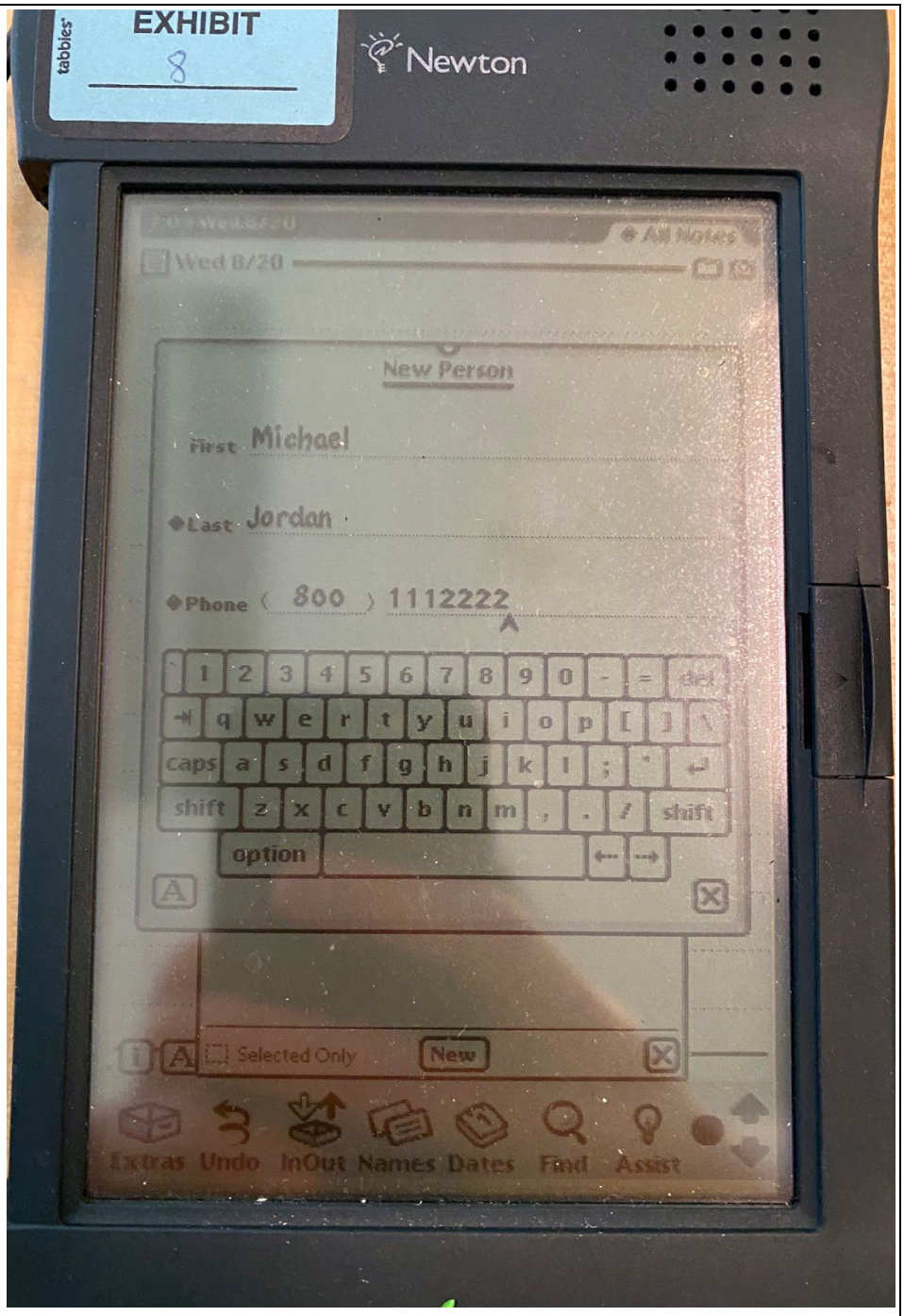
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	Newton discloses this element. For example, the photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application and then tapped the "Assist" button. As illustrated in the

Exhibit I

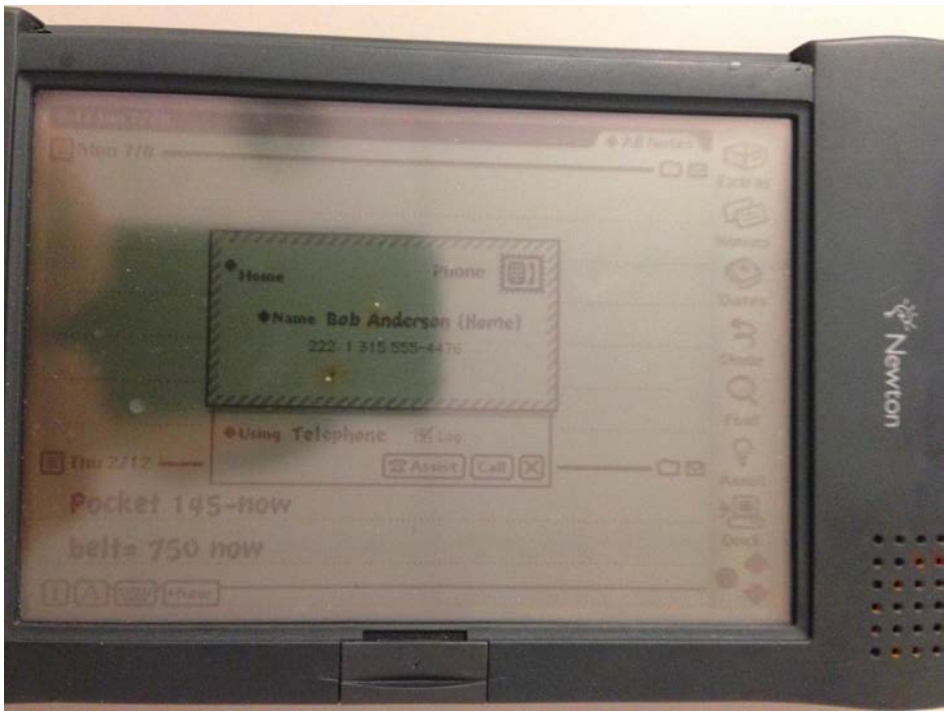
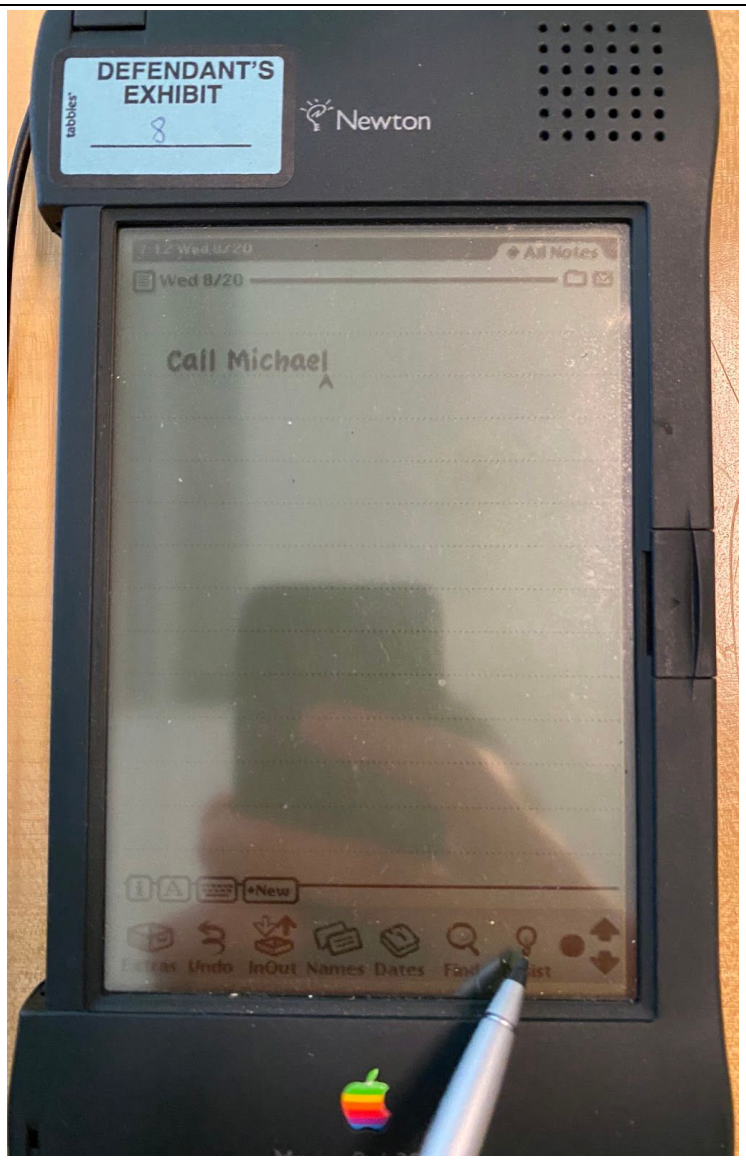
	photograph, the name "bob" was retrieved from the document and displayed in the dialog box, along with associated information (in this case a full name and phone number).  *See also* screenshots taken during my Inspection.

Exhibit I



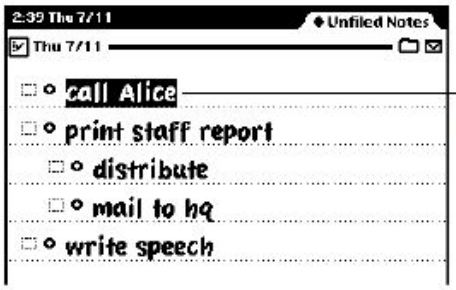
Newton Manual states:

Exhibit I

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

3 Tap Assist.

The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.

4 Enter other necessary information.

5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call.

Newton Manual, p. 196.

"Using the correct request words

The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.)

Exhibit I

- Call to dial a telephone number.

Synonyms: phone, ring, dial
Call Bob at home looks in the Name File to find Bob's home phone number, thenputs it in the call slip.

- Fax to fax the item on the screen.

Synonyms: none
Fax Anderson opens a fax slip with the name Anderson and Anderson's fax number filled in.
…"
Newton Manual, pp. 196-97.

Newton Guide states:

"The system-supplied Find slip contains an input line that specifies a search string, several buttons indicate the scope of the search, and a Look For picker (pop-up menu) that specifies the kind of search to perform."

Newton Guide, p. 16-2.

See also Newton Guide at Figure 16-1.

Figure 16-1 The system-supplied Find slip

"When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input.

The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities."

Newton Guide, p. 18-1.

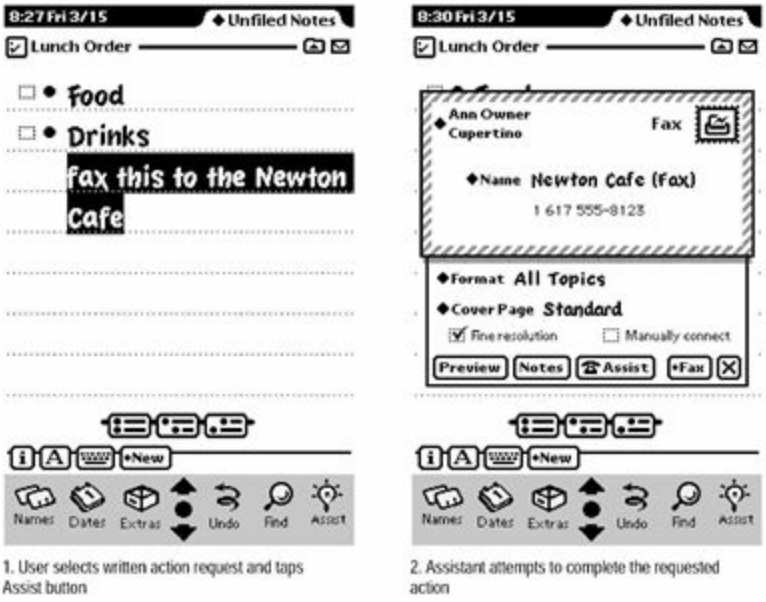
"When an action is specified but required information is still missing, the

Exhibit I

	Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup. The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose." Newton Guide, p. 18-4. See also Newton Guidelines at pp. 8-22-8-23: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly. Invoking the Assistant A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Exhibit I

Figure 8-22 The Assist button makes the Assistant try a written action request



1. User selects written action request and taps Assist button

2. Assistant attempts to complete the requested action

”

“And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax.
A. Correct.
Q. And then what does the Intelligent Assist feature to before it’s able to perform the command, such as in this instance, faxing a particular phone number?
So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that’s entered in by the user?
A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type.
And so then, um, upon being – um, invoking the Assistant in this – let’s continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote.” Pagallo Depo. at 78:19-79:15.

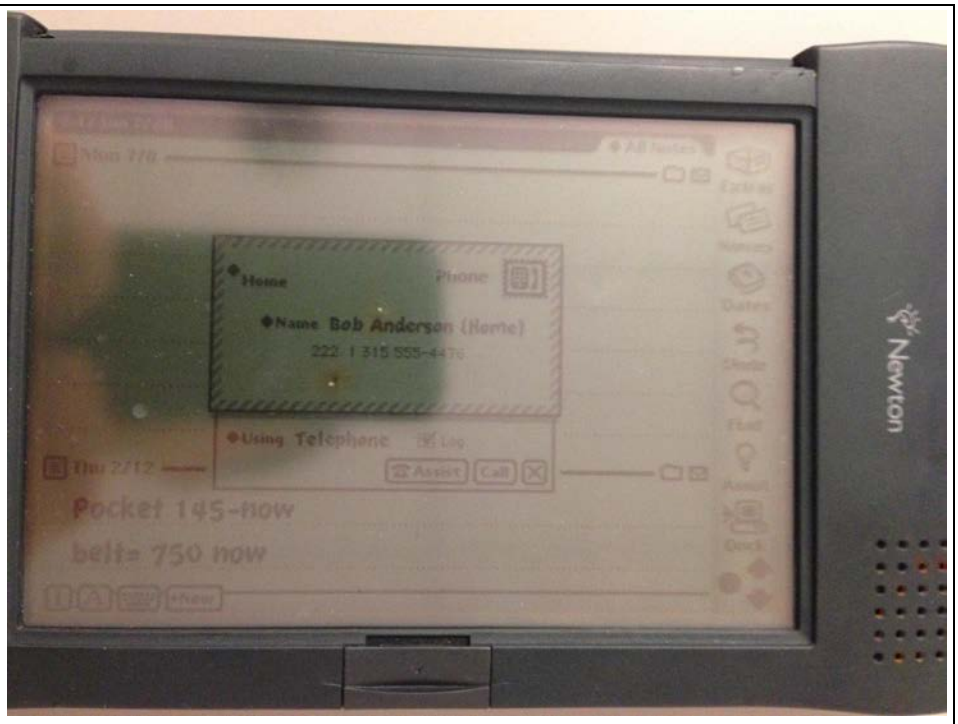
“But if you do the selection, you – the Intelligent Assistant will act on the selected text only.” Pagallo Depo. at 87:15-17.

“So it recognized the command words, I think, that we had previously

Exhibit I

	discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. Tapping the "Assist" button launches the Intelligent Assistant and causes a "Phone" window to appear. The "Phone" window includes the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). The name "Michael" contained in the text string "Call Michael" was retrieved from the Notes application document and displayed in the "Phone" window, along with the associated last name and phone number of "Michael." See also screenshots from the "while the document is being displayed" element above.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Newton discloses this element. For example, as illustrated in the photographs above, Newton included a touchscreen with which a user could input data using a stylus. Also illustrated in the photographs above, Newton further included user interface elements with which the user could interact to perform the recited actions. Tapping the "Assist" button launches the Intelligent Assist feature with text from the document. The photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application and then tapped the "Assist" button. As illustrated in the photograph below, Newton searches for second information (in this case a full name and phone number) associated with the name "bob" in the Names application, and displays this information in a dialog box.

Exhibit I

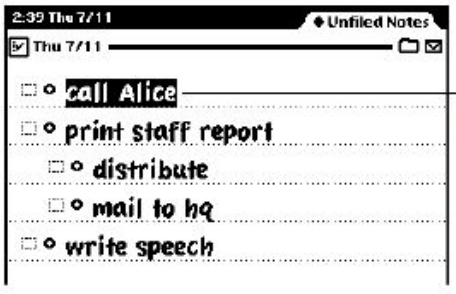


Newton Manual states:

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

Exhibit I

	3 Tap Assist. The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.  4 Enter other necessary information. 5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call. Newton Manual, p. 196. "Using the correct request words The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.) ● Call to dial a telephone number. Synonyms: phone, ring, dial ***Call Bob at home*** looks in the Name File to find Bob's home phone number, thenputs it in the call slip. ● Fax to fax the item on the screen. Synonyms: none ***Fax Anderson*** opens a fax slip with the name Anderson and Anderson's fax number filled in. …" n Manual, pp. 196-97. Newton Guide states: "When a user writes a line of text on the Newton screen, the Newton system software performs a series of operations, as follows: ● The raw data for the input is captured as ink, which is also known as **sketch ink** or **raw ink**. ● Raw ink is stored as a sequence of **strokes** or stroke data.

Exhibit I

- If the view in which the ink was drawn is configured for **ink text,** the recognition system groups the stroke data into a series of **ink words,** based on the timing and spacing of the user's handwriting. A user can insert, delete, and move ink words in the same way as recognized text. Ink words can be scaled to various sizes for display and printing. They can also be recognized at a later time, by a process known as **deferred recognition.**
- If the view in which the ink was drawn supports or is configured for text recognition, the ink words are processed by the recognition system into recognized text and displayed in a typeface."

Newton Guide, p. 8-2.

"The system-supplied Find slip contains an input line that specifies a search string, several buttons indicate the scope of the search, and a Look For picker (pop-up menu) that specifies the kind of search to perform."

Newton Guide, p. 16-2.

See also Newton Guide at Figure 16-1.

Figure 16-1 The system-supplied Find slip

"The everywhere and Selected buttons specify that the system perform searches in applications other than the currently active one. Applications must register with the Find service to participate in such searches.

Tapping the Everywhere button tells the system to conduct searches in all currently available applications registered with the Find service. This kind of search is called a Global find. Applications need not be open to participate in a Global find.

A Global find is similar to a series of Local find operations initiated by

Exhibit I

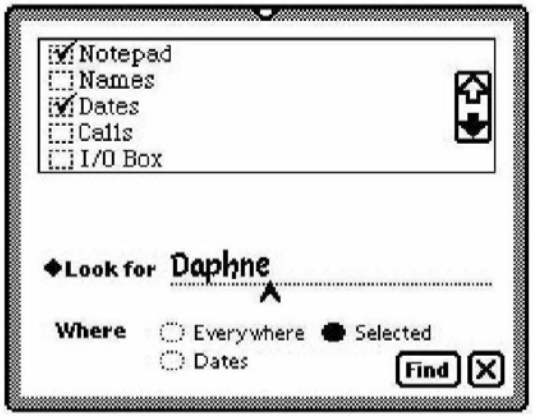
	the system. When the user requests a Global find, the system executes a Local find in each application registered with the Find service. Tapping the Selected button causes a checklist to appear at the top of the Find slip. The list includes all currently available applications registered with the Find service. Tapping items in the list places a check mark next to applications in which the system should conduct a Local find. This kind of search is called a Selected find. The slip in Figure 16-4 depicts a search for the string 'Daphne' in the Notes and Dates application." Newton Guide at 16-3. See also Newton Guide at Figure 16-4. **Figure 16-4** Searching selected applications  About the Find Service "After setting the parameters of the search with the Find slip, the user initiates the search by tapping the Find button." Newton Guide, p. 16-4. "When the search is completed, the Find service displays an overview list of items that match the search criteria. Figure 16-6 shows the Find overview as it might appear after searching all applications for the string 'Daphne'." Newton Guide, p. 16-4. See also Newton Guide at Figure 16-6.

Exhibit I

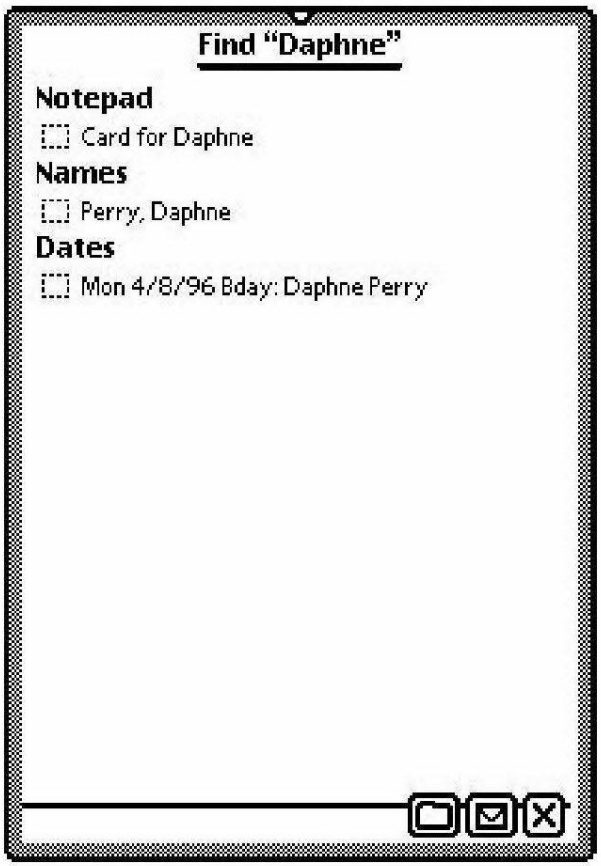
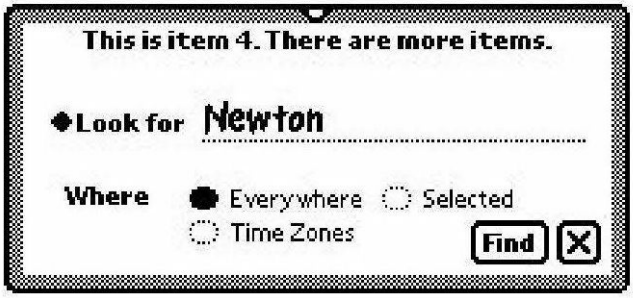
	Figure 16-6 The Find overview  "The user can tap items in the Find overview to display them. As items are displayed, a status message at the top of the Find slip indicates which item is displayed and whether there are more results to display. Figure 16-7 depicts this status message Newton Guide, p. 16-5 See also Newton Guide at Figure 16-7.

Exhibit I

Figure 16-7 Find status message



"When more than one item is found, the status message indicates that there are more items to display.

Between uses, the Find service stores the setting of the Look For picker. The next time this service is used, it reopens in the most recently set find mode. Note that in order to conserve memory, the list of found items is not saved between finds."

Newton Guide, p. 16-5.

"When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input.

The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities."

Newton Guide, p. 18-1.

"When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup.

The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose."

Exhibit I

	Newton Guide, p. 18-4. "You define the PostParse method in your task template. Your PostParse method must perform any actions required to complete the primary task. The PostParse method usually acts as a dispatching method that executes the subtasks comprising the primary action. In doing so, the PostParse method may validate data and retrieve information that the Assistant was unable to find on its own." Newton Guide, p. 18-8. "Words representing elements of the signature array do not necessarily need to appear in the input string in order to be matched to a template; for example, your PostParse method might supply Bob's fax number by finding it in the Names soup." Newton Guide, p. 18-10. "Continuing with the example based on the previous code fragment, when the Assistant parses the input phrase "fax Bob", it matches the word "fax" to the fax_action template in the first element of the signature array and creates an action frame from this template. The Assistant places this action frame in an action slot (named for the symbol in the first element of the preConditions array) that it creates in the task frame. Similarly, the Assistant creates a target frame when the word "Bob" is matched to the who_obj template in the third element of the signature array. The Assistant places this target frame in a recipient slot (named for the symbol in the third element of the preConditions array) that it creates in the task frame. Words representing elements of the signature array do not necessarily need to appear in the input string in order to be matched to a template; for example, your PostParse method might supply Bob's fax number by finding it in the Names soup." Newton Guide, p. 18-10. See also Newton Guidelines at pp. 8-22-8-28: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of

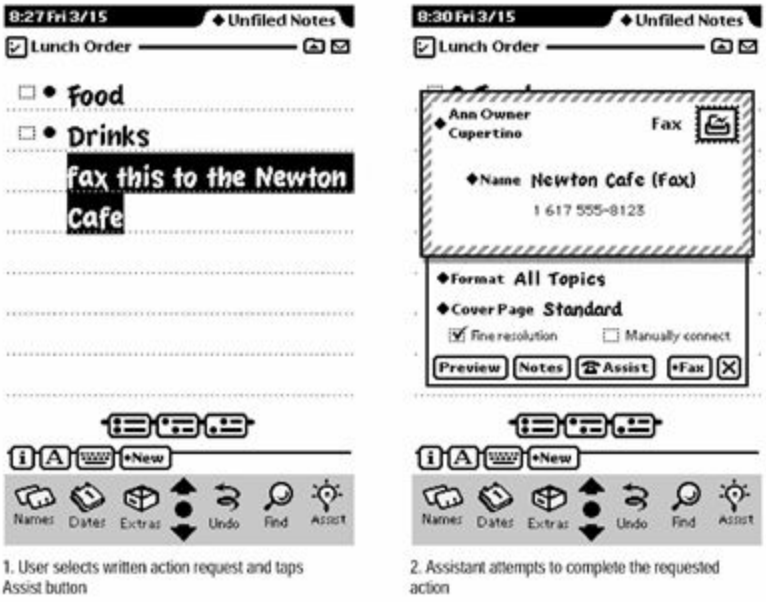
Exhibit I

built-in applications. If your application uses Assistant services, it should behave similarly.

Invoking the Assistant

A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Figure 8-22 The Assist button makes the Assistant try a written action request



Interpreting the Request Phrase

The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same task if given the word ‘phone,’ ‘call,’ or ‘dial.’ Applications can add words to the Assistant’s lexicon, but users cannot.

Exhibit I

	The Assistant matches words regardless of their capitalization. For example, it considers the word 'phone' to be the same as the word 'Phone.' The order in which the user writes words is not significant. For example, the phrase 'Royce fax' produces the same result as the phrase 'fax Royce.' This syntax-independent architecture allows easier localization of applications for international audiences. The Assistant ignores words it does not know, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as 'Make a phone call to Bob at work' and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words. Assist Slip If the Assistant can't interpret a user's written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip's Please picker and can write on the slip's input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote 'Bob,' the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words. [Figure 8-23] An Assist slip's Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker. [Figure 8-24] The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your applications registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top

Exhibit I

	portion of the Please picker. For example, the word ‘call’ appears but the synonyms ‘ring’ and ‘phone’ do not. Recently used synonyms may appear in the bottom half of the picker, however. If a user writes a verb the Assistant doesn’t know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase ‘buzz 555-1234’ the Assistant does not match the word ‘buzz’ to an action, but it can identify ‘555-1234’ as having the format of a telephone number. Based on that information, the Assistant deduces the user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker. In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see ‘Help’ on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action. The primary function of an Assist slip is to specify an action. IF the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip. Task Slips. Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user writes the request ‘fax Bob,’ the Assistant can ge Bob’s fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page. The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see ‘Routing Slips’ on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see ‘Find’ on page 86).

Exhibit I

	Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way." "Q. What is the Intelligent Assist feature? A. Um, the Intelligent Assist feature is in the Newton, is a system service that was designed to facilitate interactions and user actions. Like, for instance, placing a call, sending an email, creating a reminder, so that the users could do that from, say, the Notes application without necessarily having to go into the calendar or the date, the Names application. Q. Or even going – A. The call. Q. – into the Call application. A. For instance, yes." Pagallo Depo. at 68:5-17. "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "Q. And then after the text is entered in, then the user presses the Intelligent Assist button, the Intelligent Assist feature is what is recognizing or the event – A. What is understanding the command. Q. Okay. So within Intelligent Assist, are there specific commands – only specific commands or actions that the Intelligent Assist feature can recognize? A. Yes." Pagallo Depo. at 75:1-9. "And so after the user enters in the text, fax, and a phone number, and

Exhibit I

	then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if a user taps the word Please, then it – um, is the user presented with different options for actions that could occur? A. The different option for action that could occur with a person's name." Pagallo Depo. at 85:5-9. "So underneath the action word, call, fax, find, mail, print, remember, schedule in time, the other items that appear under the Please Picker are the recent action – A. I believe so." Pagallo Depo at 85:21-25. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh. Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * * A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work.

Exhibit I

	Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4 "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "And if a contact with that name does not already exist, what would the user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and

Exhibit I

	then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number. A. Yes." Pagallo Depo at 92:6-20. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "And then if, in the instance that the information that's presented to the user for Bob is correct, and the user decides to place the phone call, what does the user need to do next after it's presented with the – the screen? A. Tap on the button that says Call." Pagallo Depo at 97:23-98:3. "So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up. Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant and caused a "Phone" window to appear. The "Phone" window includes the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). The name "Michael" contained in the text

Exhibit I

	string "Call Michael" was retrieved from the Notes application document and displayed in the "Phone" window, along with the associated last name and phone number of "Michael." See also screenshots from the "while the document is being displayed" element above. As can be seen in the screenshots, I then tapped the "Call" button in the "Phone" window with a stylus, which caused Newton to generate tones to dial the phone number ("1 800 1112222") associated with the name "Michael." *See also* screenshots taken during my Inspection.

Exhibit I

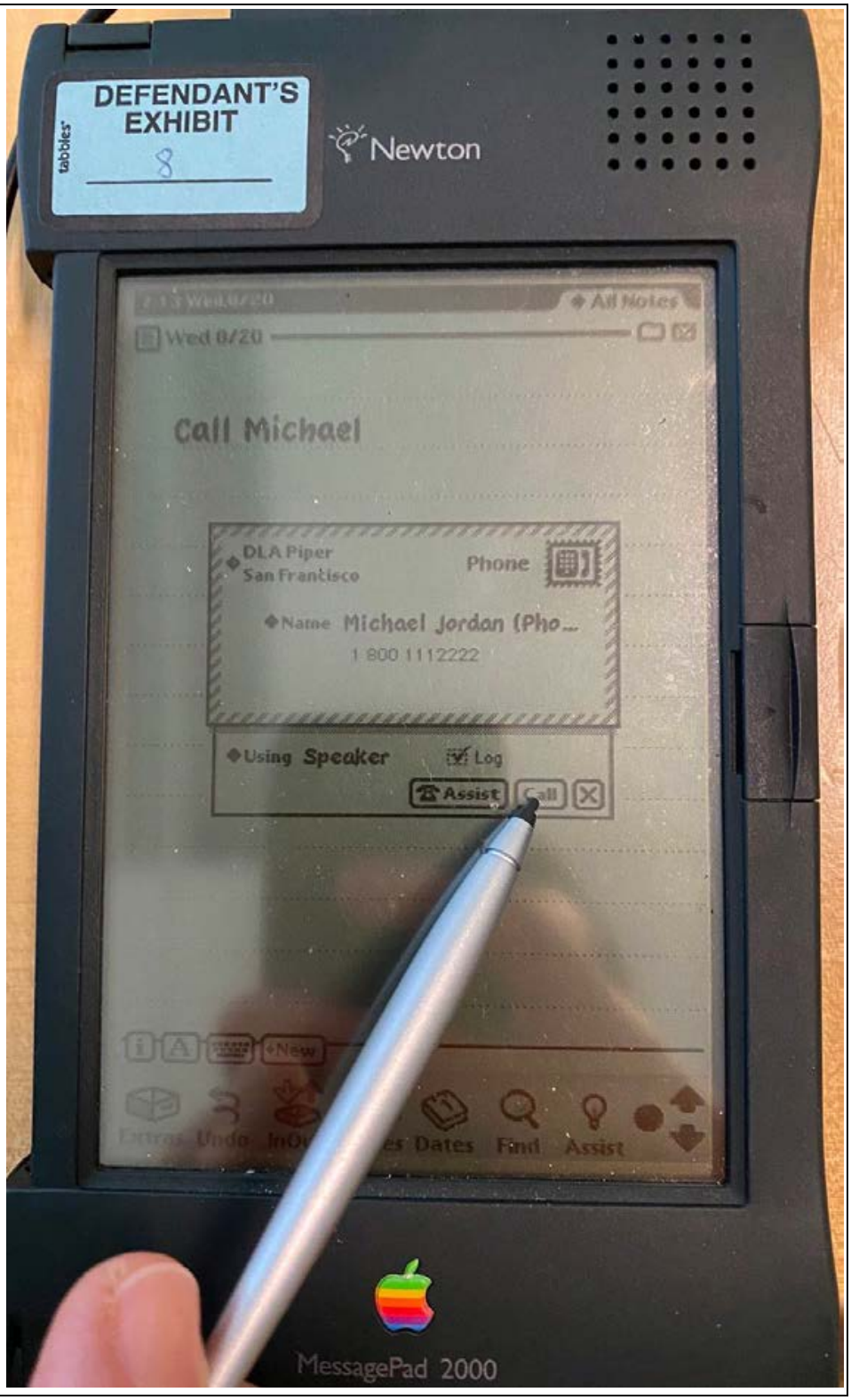
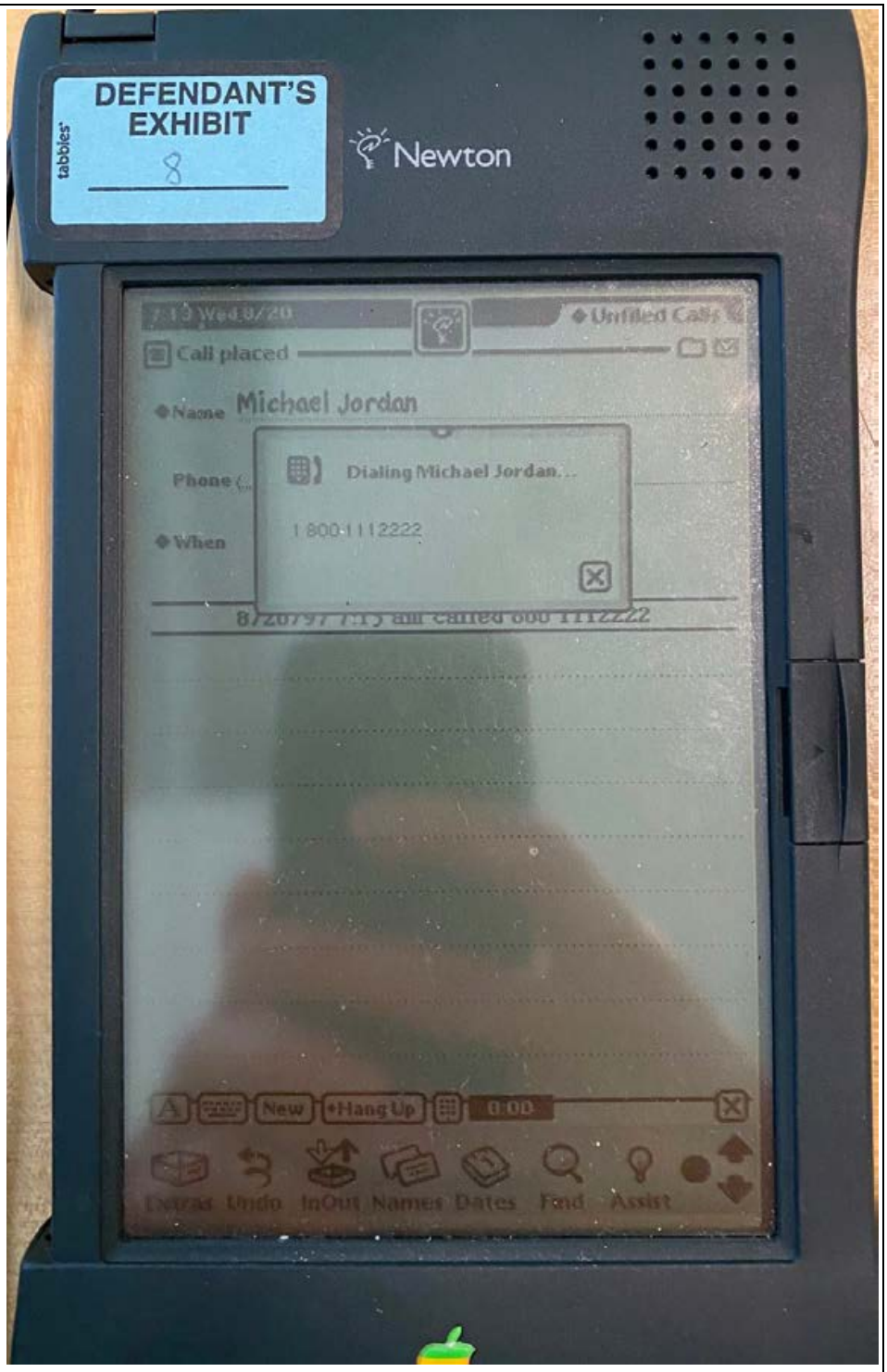

DEFENDANT'S EXHIBIT
8
tabbies
Newton
All Notes
Wed 8/20
Call Michael
DLA Piper
San Francisco
Phone
Name Michael Jordan (Pho...
1 800 1112222
Using Speaker
Log
Assist
Call
New
Undo
Dates
Find
Assist
MessagePad 2000

Exhibit I



Prior to the aforementioned sequence of events observed during my Inspection, a new contact card was added to include the first name "Michael," last name "Jordan," and phone number "800 1112222" to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card). The last name ("Jordan") and

Exhibit I

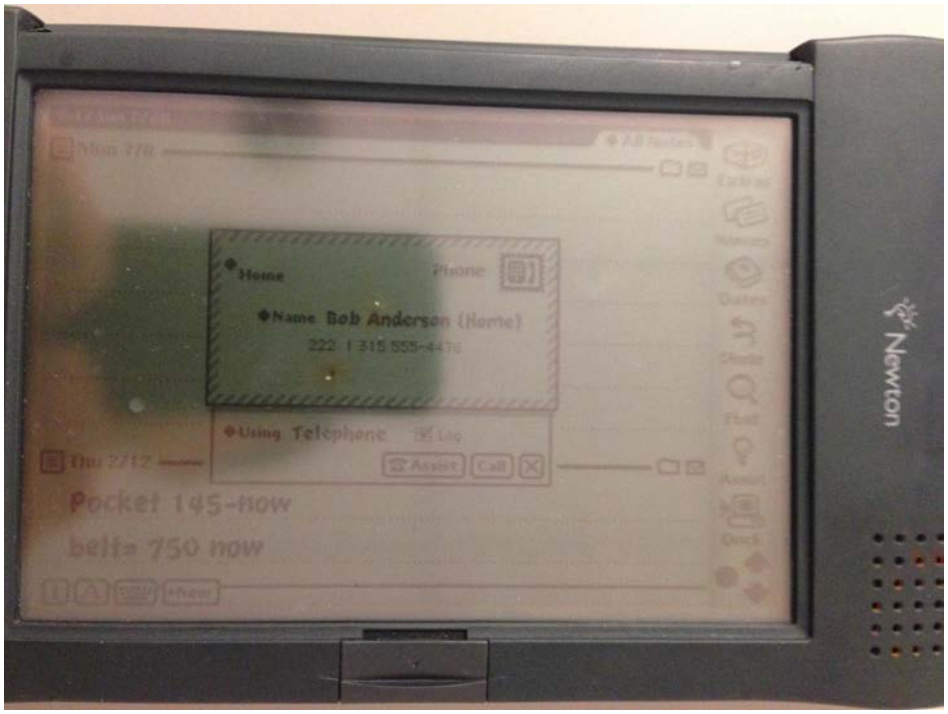
	phone number ("1 800 1112222") was associated with the first name ("Michael") in this contact card. Indeed, prior to adding this contact card to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card), the same aforementioned sequence of events (i.e. "Call Michael") did not result in the display of any other information associated with the name "Michael." See also screenshots from the "while the document is being displayed" element above. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Newton discloses this element. For example, tapping the "Assist" button launches the Intelligent Assist feature with text from the document. The photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application and then tapped the "Assist" button. As illustrated in the photograph below, Newton searches for second information (in this case a full name and phone number) associated with the name "bob" in the Names application, and displays this information in a dialog box. 

Exhibit I

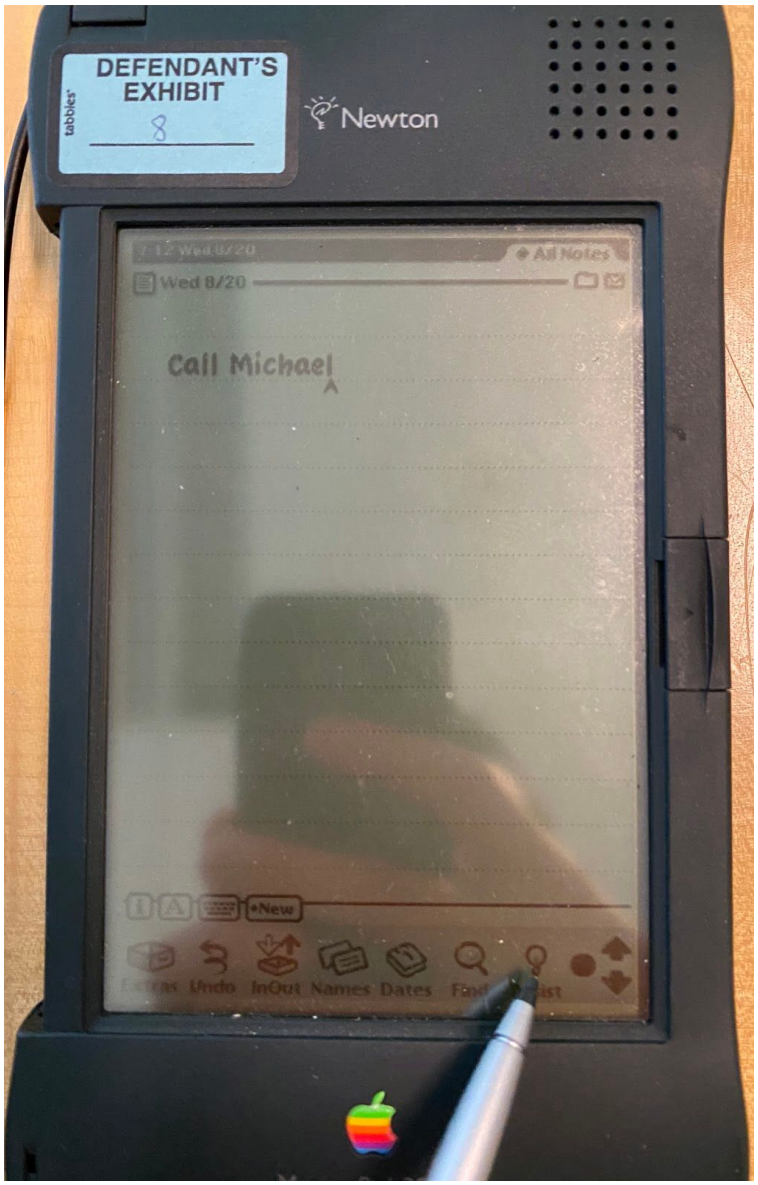
	See also screenshots taken during my Inspection. 

Exhibit I

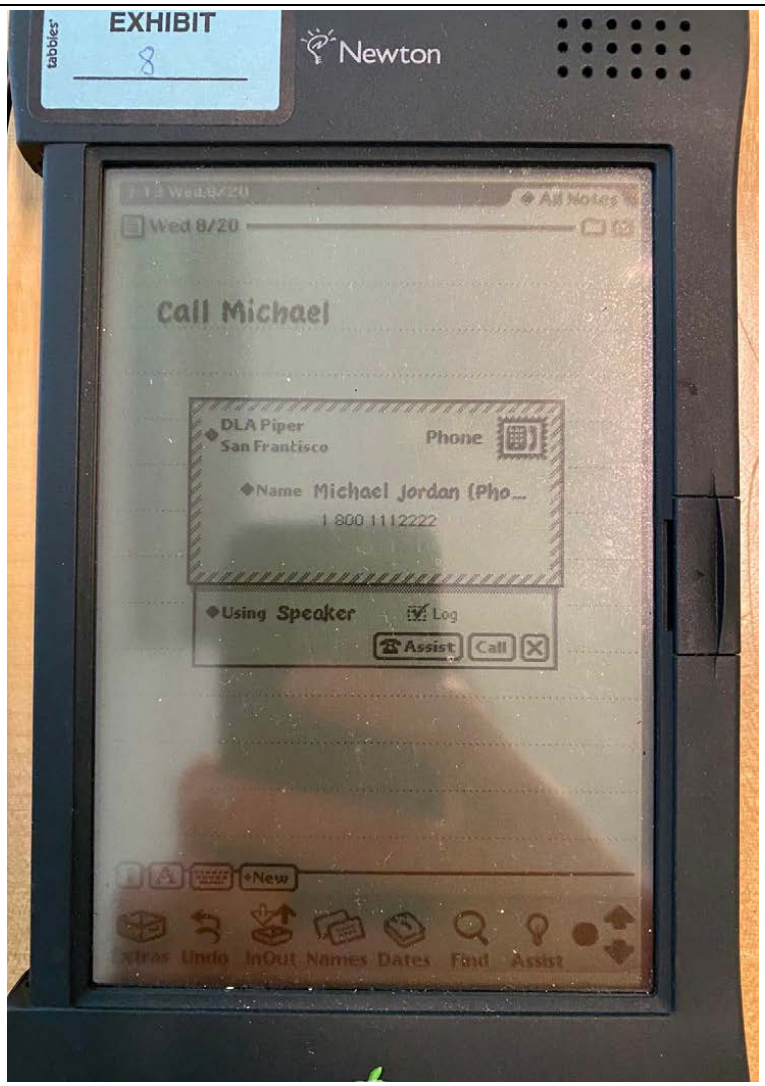
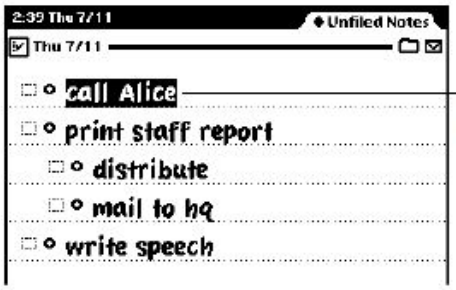
	 Newton Manual states:

Exhibit I

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

3 Tap Assist.

The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.

4 Enter other necessary information.

5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call.

Newton Manual, p. 196.

"Using the correct request words

The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.)

Exhibit I

- Call to dial a telephone number.

Synonyms: phone, ring, dial

Call Bob at home looks in the Name File to find Bob's home phone number, thenputs it in the call slip.

- Fax to fax the item on the screen.

Synonyms: none

Fax Anderson opens a fax slip with the name Anderson and Anderson's fax number filled in.

…"

Newton Manual, pp. 196-97.

Using the Name File

You can use the Name File as an address book to store information about people, companies, and groups. The Name File contains name cards that you create. Each card has information such as name, address, telephone numbers, electronic mail addresses, and notes. You can also create your own field labels for special information.

Tap Names to go to the Name File. Tap it again to put away the Name File. You can also tap in the lower-right corner of the Name File to put it away.

Newton Manual, p. 55; *see also* pp. 50-51 (creating new cards), 52-54 (adding information to a card), 54 (changing information in an existing card), 56-57 (viewing) cards.

Newton Guide states:

"The everywhere and Selected buttons specify that the system perform searches in applications other than the currently active one. Applications must register with the Find service to participate in such searches.

Tapping the Everywhere button tells the system to conduct searches in

Exhibit I

all currently available applications registered with the Find service. This kind of search is called a Global find. Applications need not be open to participate in a Global find.

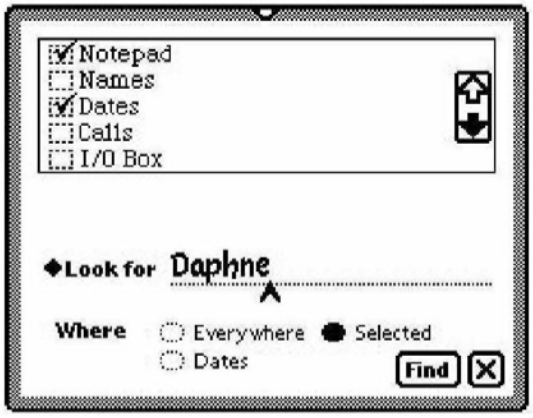
A Global find is similar to a series of Local find operations initiated by the system. When the user requests a Global find, the system executes a Local find in each application registered with the Find service.

Tapping the Selected button causes a checklist to appear at the top of the Find slip. The list includes all currently available applications registered with the Find service. Tapping items in the list places a check mark next to applications in which the system should conduct a Local find. This kind of search is called a Selected find. The slip in Figure 16-4 depicts a search for the string 'Daphne' in the Notes and Dates application."

Newton Guide at 16-3.

See also Newton Guide at Figure 16-4.

Figure 16-4 Searching selected applications



About the Find Service

"After setting the parameters of the search with the Find slip, the user initiates the search by tapping the Find button."

Newton Guide, p. 16-4.

"When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input.

Exhibit I

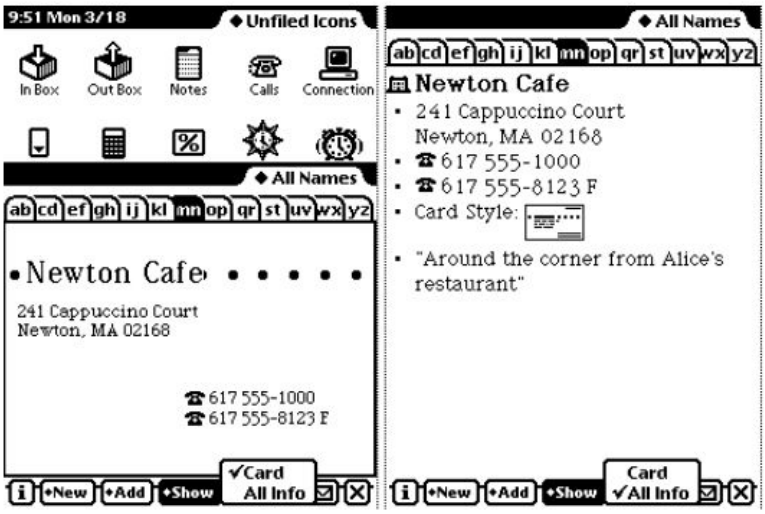
	The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities." Newton Guide, p. 18-1. "When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup. The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose." Newton Guide, p. 18-4. "You define the PostParse method in your task template. Your PostParse method must perform any actions required to complete the primary task. The PostParse method usually acts as a dispatching method that executes the subtasks comprising the primary action. In doing so, the PostParse method may validate data and retrieve information that the Assistant was unable to find on its own." Newton Guide, p. 18-8. "Words representing elements of the signature array do not necessarily need to appear in the input string in order to be matched to a template; for example, your PostParse method might supply Bob's fax number by finding it in the Names soup." Newton Guide, p. 18-10. "Continuing with the example based on the previous code fragment, when the Assistant parses the input phrase "fax Bob", it matches the word "fax" to the fax_action template in the first element of the signature array and creates an action frame from this template. The Assistant places this action frame in an action slot (named for the symbol in the first element of the preConditions array) that it creates in the task frame. Similarly, the Assistant creates a target frame when the word "Bob" is matched to the who_obj template in the third element of the signature array. The Assistant places this target frame in a recipient slot (named for the symbol in the third element of the preConditions array) that it creates

Exhibit I

	in the task frame. Words representing elements of the signature array do not necessarily need to appear in the input string in order to be matched to a template; for example, your PostParse method might supply Bob's fax number by finding it in the Names soup." Newton Guide, p. 18-10. "Your PostParse method implements the behavior your application provides through the Assistant. This method resides in the PostParse slot of your task template. It is called after all the templates in the signature slot have been matched. Your PostParse method must provide any behavior required to complete the specified task, such as obtaining additional information from the ParseUtter result frame as necessary, and handling error conditions." Newton Guide, p. 18-17. "Names This section describes the application program interface (API) to the Names application. The Names application manages a database of people and places. It presents information either as a business card, or as a list of all available information. These two views are shown in Figure 19-1." Newton Guide, p. 19-2. "The Names application stores its data in the ROM_CardFileSoupName ("Names") soup. Entries in this soup are frames for either a person, an owner, a group, a company, or a worksite card." Newton Guide, p. 19-7.

Exhibit I

Figure 19-1 Names application Card and All Info views



Newton Guide, p. 19-3.

See also Newton Guidelines at pp. 8-22: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly."

See also Newton Guidelines at 8-27-8-28: "Task Slips.
Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip.

For example, if a user writes the request 'fax Bob,' the Assistant can get Bob's fax number from the Names File application. But what if Bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page.

The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing

Exhibit I

	actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86). Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way." "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "Q. And then after the text is entered in, then the user presses the Intelligent Assist button, the Intelligent Assist feature is what is recognizing or the event – A. What is understanding the command. Q. Okay. So within Intelligent Assist, are there specific commands – only specific commands or actions that the Intelligent Assist feature can recognize? A. Yes." Pagallo Depo. at 75:1-9. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number?

Exhibit I

	So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if a user taps the word Please, then it – um, is the user presented with different options for actions that could occur? A. The different option for action that could occur with a person's name." Pagallo Depo. at 85:5-9. "So underneath the action word, call, fax, find, mail, print, remember, schedule in time, the other items that appear under the Please Picker are the recent action – A. I believe so." Pagallo Depo at 85:21-25. "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "And if a contact with that name does not already exist, what would the user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez

Exhibit I

	is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number. A. Yes." Pagallo Depo at 92:6-20. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up. Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12.

Exhibit I

	For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant. Newton then searched the Names application contact cards (or, more specifically, the "Names soup," which contains the data for each contact card) for information associated with the name "Michael." A "Phone" window then appeared to include the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). See also screenshots from the "while the document is being displayed" element above. Prior to the aforementioned sequence of events observed during my Inspection, a new contact card was added to include the first name "Michael," last name "Jordan," and phone number "800 1112222" to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card). The last name ("Jordan") and phone number ("1 800 1112222") was associated with the first name ("Michael") in this contact card. Indeed, prior to adding this contact card to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card), the same aforementioned sequence of events (i.e. "Call Michael") did not result in the display of any other information associated with the name "Michael." See also screenshots from the "while the document is being displayed" element above. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Newton discloses this element. For example, tapping the "Assist" button launches the Intelligent Assist feature with text from the document. The photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application and then tapped the "Assist" button. As illustrated in the photograph below, Newton searches for second information (in this case a full name and phone number) associated with the name "bob" in the Names application, and displays this information in a dialog box.

Exhibit I

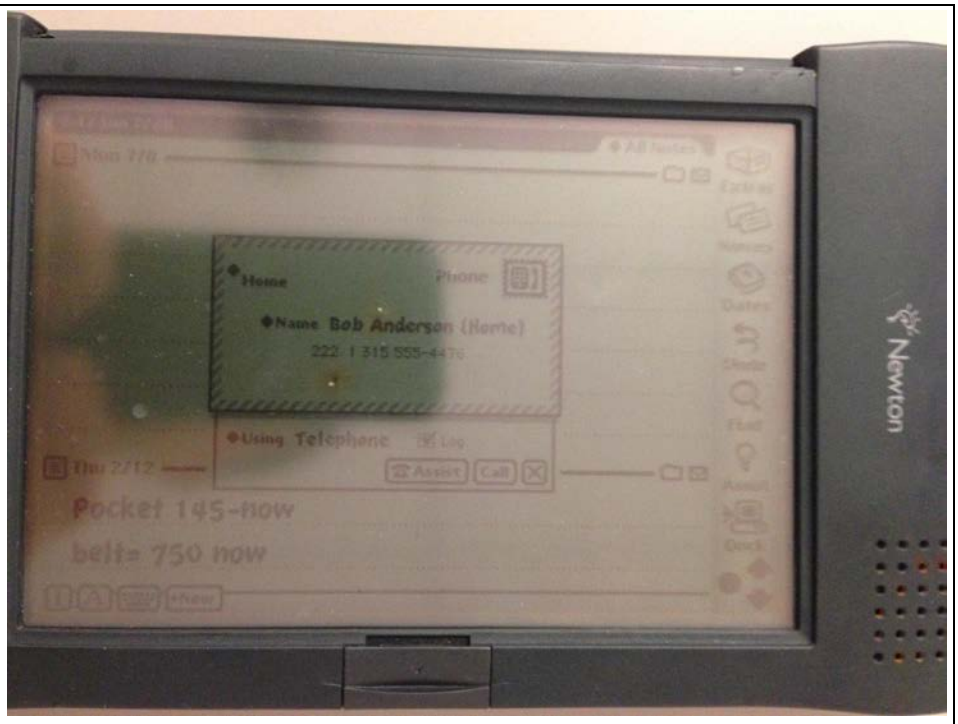
	 Tapping the "call" button leads to the MessagePad generating tones to place a call to the phone number associated with the name "bob" in the Names application. The photograph below illustrates the display of the Newton device after the user has entered the text "email bob" in the Notes application and then tapped the "Assist" button. Tapping the "mail" button allows a user to send an email to the email address associated with the name "bob" in the Names application.

Exhibit I

	 *See also* other screenshots from inspection at DLA Piper.

Exhibit I

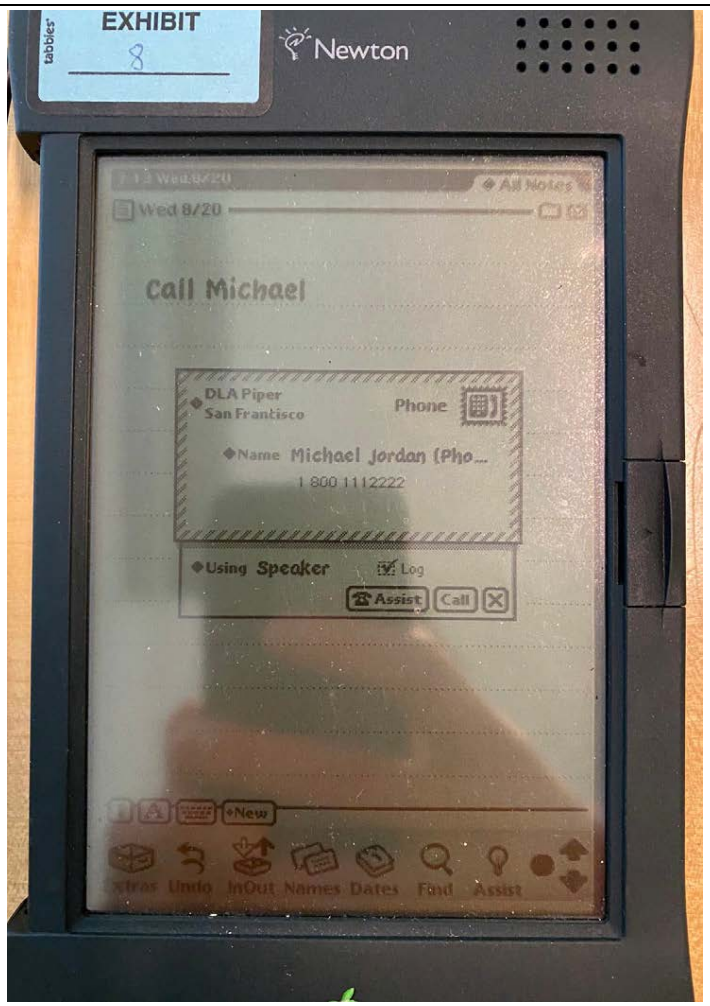


EXHIBIT
8
Newton
Wed 8/20
All Notes
Call Michael
DLA Piper
San Francisco
Phone
Name Michael Jordan (Pho...
1 800 1112222
Using Speaker
Log
Assist
Call
New
Xtras
Undo
In/Out
Names
Dates
Find
Assist

Exhibit I

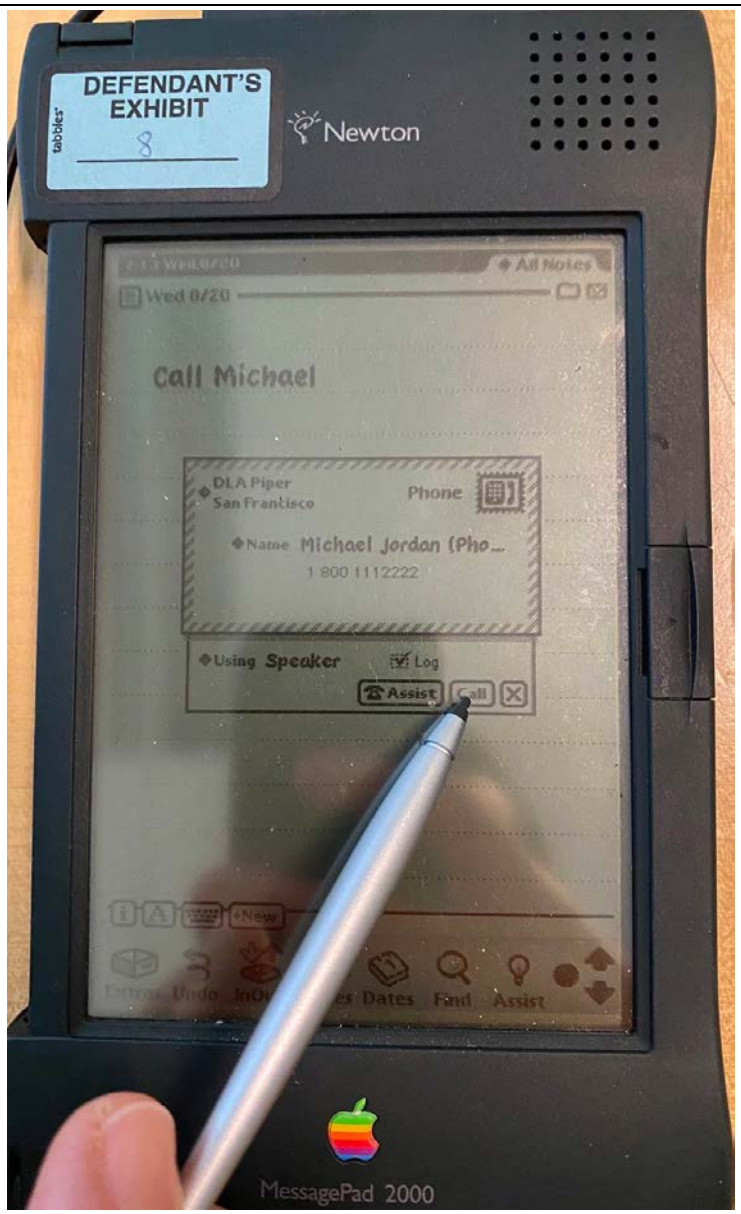
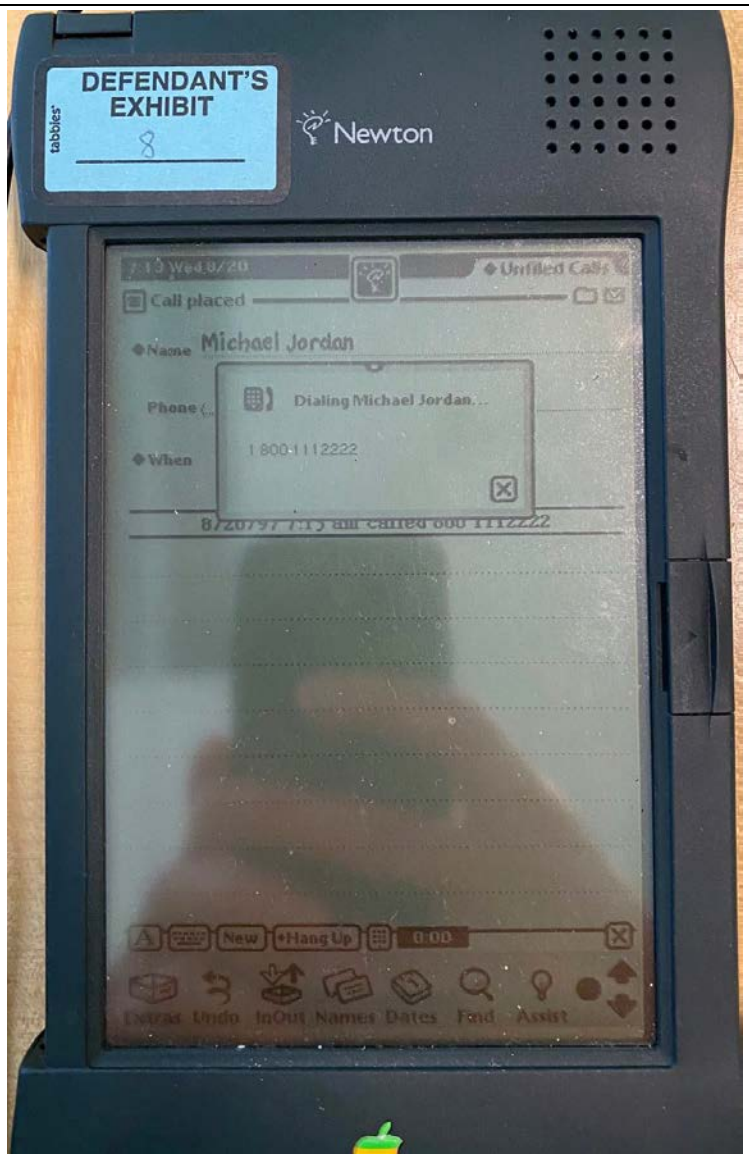

DEFENDANT'S EXHIBIT
8
Newton
Call Michael
DLA Piper San Francisco
Phone
Name Michael Jordan (Pho...
1 800 1112222
Using Speaker
Log
Assist
Call
Dates
Find
Assist
MessagePad 2000

Exhibit I



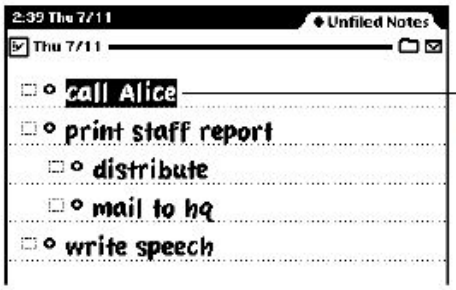
Newton Manual states:

Exhibit I

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

3 Tap Assist.

The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.

4 Enter other necessary information.

5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call.

Newton Manual, p. 196.

"Using the correct request words

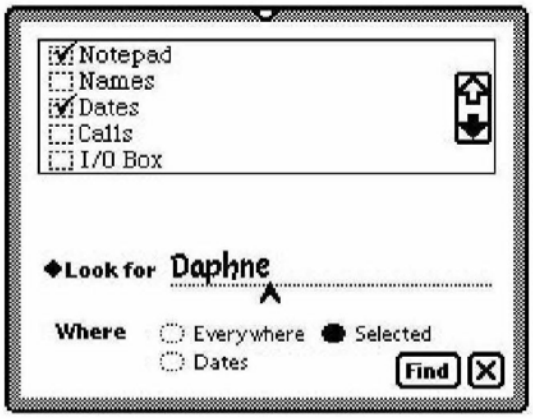
The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.)

Exhibit I

	● Call to dial a telephone number. Synonyms: phone, ring, dial ***Call Bob at home*** looks in the Name File to find Bob's home phone number, thenputs it in the call slip. ● Fax to fax the item on the screen. Synonyms: none ***Fax Anderson*** opens a fax slip with the name Anderson and Anderson's fax number filled in. …" Newton Manual, pp. 196-97. Newton Guide states: "The everywhere and Selected buttons specify that the system perform searches in applications other than the currently active one. Applications must register with the Find service to participate in such searches. Tapping the Everywhere button tells the system to conduct searches in all currently available applications registered with the Find service. This kind of search is called a Global find. Applications need not be open to participate in a Global find. A Global find is similar to a series of Local find operations initiated by the system. When the user requests a Global find, the system executes a Local find in each application registered with the Find service. Tapping the Selected button causes a checklist to appear at the top of the Find slip. The list includes all currently available applications registered with the Find service. Tapping items in the list places a check mark next to applications in which the system should conduct a Local find. This kind of search is called a Selected find. The slip in Figure 16-4 depicts a search for the string 'Daphne' in the Notes and Dates application." Newton Guide at 16-3. See also Newton Guide at Figure 16-4.

Exhibit I

Figure 16-4 Searching selected applications



About the Find Service

"After setting the parameters of the search with the Find slip, the user initiates the search by tapping the Find button."

Newton Guide, p. 16-4.

"When the search is completed, the Find service displays an overview list of items that match the search criteria. Figure 16-6 shows the Find overview as it might appear after searching all applications for the string 'Daphne'."

Newton Guide, p. 16-4.

"When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup.

The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose."

Newton Guide, p. 18-4.

Exhibit I

Figure 18-3 Calling task slip

Newton Guide, p. 18-4.

"The built-in tasks that the Assistant can complete include calling, faxing, finding, mailing, printing, remembering (adding To Do items), scheduling (adding meetings), getting time information from the built-in Time Zones application and displaying online helNewton Guide, p. Note that the top portion of this menu displays only one word for each action the Assistant can perform. For example, the word "call" appears here but the synonyms "ring" and "phone" do not. Recently used synonyms may appear in the bottom half of the slip, however."

Newton Guide, p. 18-3.

"When routing an item from the Assistant—for example, when filing, faxing, printing, or mailing the item—the Assistant sends a GetTargetInfo message to your application"

Newton Guide, p. 18-20.

"// Action template for sending electronic mail
mail_act := {
lexicon: ["mail", "send", "email"],
…}"

Newton Guide, p. 18-24.

"Newton can serve as a phone dialer by dialing phone numbers through the speaker. The dialing tones are built into the system ROM, along with several other sounds that can be used in applications."

Exhibit I

	Newton Guide, p. 1-9. See also Newton Guidelines at fig. 7-24. See also Newton Guidelines at pp. 8-22-8-28: "The Intelligent Assistant is a system service that attempts to complete actions specified by a user's written input. You can think of the Assistant as an alternate interface to Newton applications and services. The Assistant can complete a number of tasks using the built-in applications and services, and your application can extend the Assistant to carry out tasks that the application performs. Users can also display an application's online help from the Assistant. This section describes the Assistant's user interface in the context of built-in applications. If your application uses Assistant services, it should behave similarly. Invoking the Assistant A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Exhibit I

Figure 8-22 The Assist button makes the Assistant try a written action request

1. User selects written action request and taps Assist button

2. Assistant attempts to complete the requested action

Interpreting the Request Phrase

The Assistant can attempt to complete an action only if it can construe one from the phrase the user writes or selects before tapping the Assist button. The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications. It can associate multiple verbs with a single action. For example, the Assistant performs the same task if given the word 'phone,' 'call,' or 'dial.' Applications can add words to the Assistant's lexicon, but users cannot.

The Assistant matches words regardless of their capitalization. For example, it considers the word 'phone' to be the same as the word 'Phone.'

The order in which the user writes words is not significant. For example, the phrase 'Royce fax' produces the same result as the phrase 'fax Royce.' This syntax-independent architecture allows easier localization of applications for international audiences.

The Assistant ignores words it does not known, giving users the freedom to write naturally. Rather than limiting the user to terse commands, the Assistant extracts meaningful words from phrases such as 'Make a phone call to Bob at work' and ignores the others. There is a limit to this freedom, however. The Assistant does not attempt to interpret a phrase containing more than 15 words.

Exhibit I

	Assist Slip If the Assistant can't interpret a user's written request, the Assistant displays an Assist slip where the user can provide more information. The user can choose an action from the Assist slip's Please picker and can write on the slip's input line. If a user wrote some words before tapping the Assist button but did not write enough to clearly specify an action, the Assist slip displays those words and includes a message prompting the user to provide additional information. For example, if a user just wrote 'Bob,' the Assistant could perform a number of actions: it could find Bob, fax Bob, call Bob, schedule a meeting with Bob, and so on. Figure 8-23 shows examples of Assist slips with too few words and with no words. [Figure 8-23] An Assist slip's Please picker lists the actions that applications have currently registered with it, as well as eight phrases the Assistant has tried to interpret recently. Figure 8-24 shows a sample Please picker. [Figure 8-24] The built-in tasks that the Assistant lists in the Please picker include calling, faxing, finding, mailing, printing, remembering To Do items, scheduling meetings, and getting time information from the Time Zones application. If your applications registers additional actions with the Assistant, they automatically appear in the Please picker. Note that the top portion of this picker displays only one word for each action. If the Assistant knows synonyms for an action, they do not appear in the top portion of the Please picker. For example, the word 'call' appears but the synonyms 'ring' and 'phone' do not. Recently used synonyms may appear in the bottom half of the picker, however. If a user writes a verb the Assistant doesn't know, it may be able to deduce a likely action from other words. For example, if a user writes the phrase 'buzz 555-1234' the Assistant does not match the word 'buzz' to an action, but it can identify '555-1234' as having the format of a telephone number. Based on that information, the Assistant deduces the user wants to call, fax, or find the phone number, and it lists only those actions in the Please picker. In addition to the Please picker and an input line, an Assist slip has a How Do I? button in the lower left corner for accessing the Newton online help service (see 'Help' on page 8-28). In the lower right corner of an Assist slip are a Do button for initiating the action specified in the slip and a large Close box for canceling the action.

Exhibit I

	The primary function of an Assist slip is to specify an action. IF the Assistant can determine an action to take based on words a user writes or selects before tapping the Assist button, then the Assistant does not display an Assist slip. Once the Assistant knows what action to take, it can resolve other missing or ambiguous information with a task slip. Task Slips. Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user write the request 'fax Bob,' the Assistant can ge Bob's fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page. The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86). Besides providing a means of correcting missing or ambiguous information, a task slip also gives a user one last chance to confirm or cancel execution of the task before the Assistant actually takes action. It's especially important to provide this opportunity to confirm, modify, or cancel the task if executing it will change the user's current context (open other applications), modify the user's data, or inconvenience the user in some way." "Q. What is the Intelligent Assist feature? A. Um, the Intelligent Assist feature is in the Newton, is a system service that was designed to facilitate interactions and user actions. Like, for instance, placing a call, sending an email, creating a reminder, so that the users could do that from, say, the Notes application without necessarily having to go into the calendar or the date, the Names application. Q. Or even going –

Exhibit I

	A. The call. Q. – into the Call application. A. For instance, yes." Pagallo Depo. at 68:5-17. "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So underneath the action word, call, fax, find, mail, print, remember, schedule in time, the other items that appear under the Please Picker are the recent action – A. I believe so." Pagallo Depo at 85:21-25. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh.

Exhibit I

	Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * * A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work. Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4 "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "And if a contact with that name does not already exist, what would the user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16.

Exhibit I

	"Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number. A. Yes." Pagallo Depo at 92:6-20. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "And then if, in the instance that the information that's presented to the user for Bob is correct, and the user decides to place the phone call, what does the user need to do next after it's presented with the – the screen? A. Tap on the button that says Call." Pagallo Depo at 97:23-98:3. "So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up.

Exhibit I

	Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant. Newton then searched the Names application contact cards (or, more specifically, the "Names soup," which contains the data for each contact card) for information associated with the name "Michael." A "Phone" window then appeared to include the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). See also screenshots from the "while the document is being displayed" element above. As can be seen in the screenshots, I then tapped the "Call" button in the "Phone" window with a stylus, which caused Newton to generate tones to dial the phone number ("1 800 1112222") associated with the name "Michael." See also screenshots from the "providing an input device . . ." element above. Prior to the aforementioned sequence of events observed during my Inspection, a new contact card was added to include the first name "Michael," last name "Jordan," and phone number "800 1112222" to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card). The last name ("Jordan") and phone number ("1 800 1112222") was associated with the first name ("Michael") in this contact card. Indeed, prior to adding this contact card to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card), the same aforementioned sequence of events (i.e. "Call Michael") did not result in the display of any other information associated with the name "Michael." See also screenshots from the "while the document is being displayed" element above. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.

Claim 8	
A method according to claim 1,	Newton discloses claim 1. *See* claim 1 above.

Exhibit I

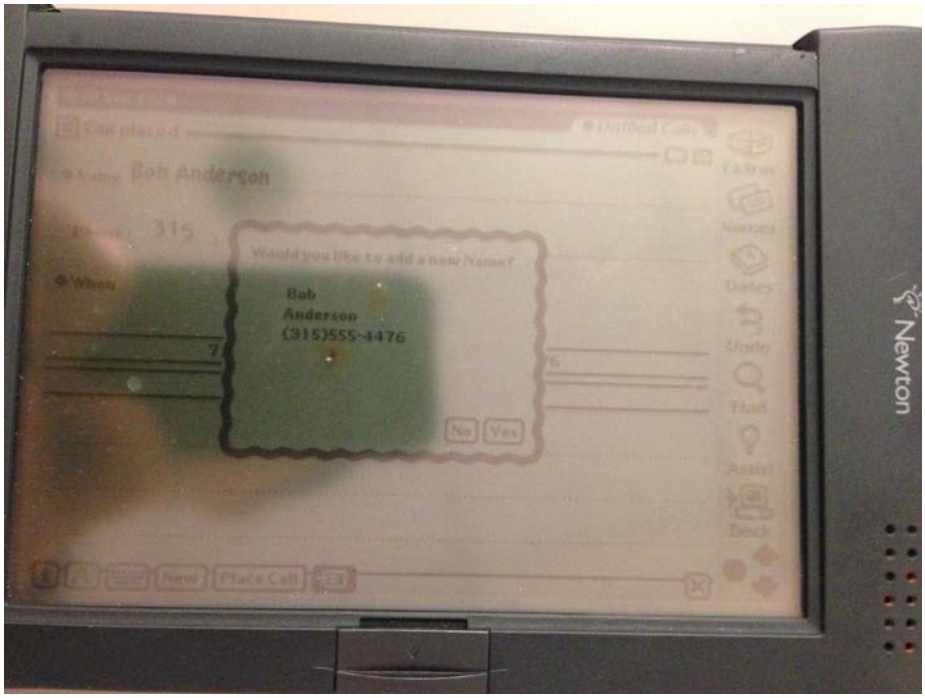
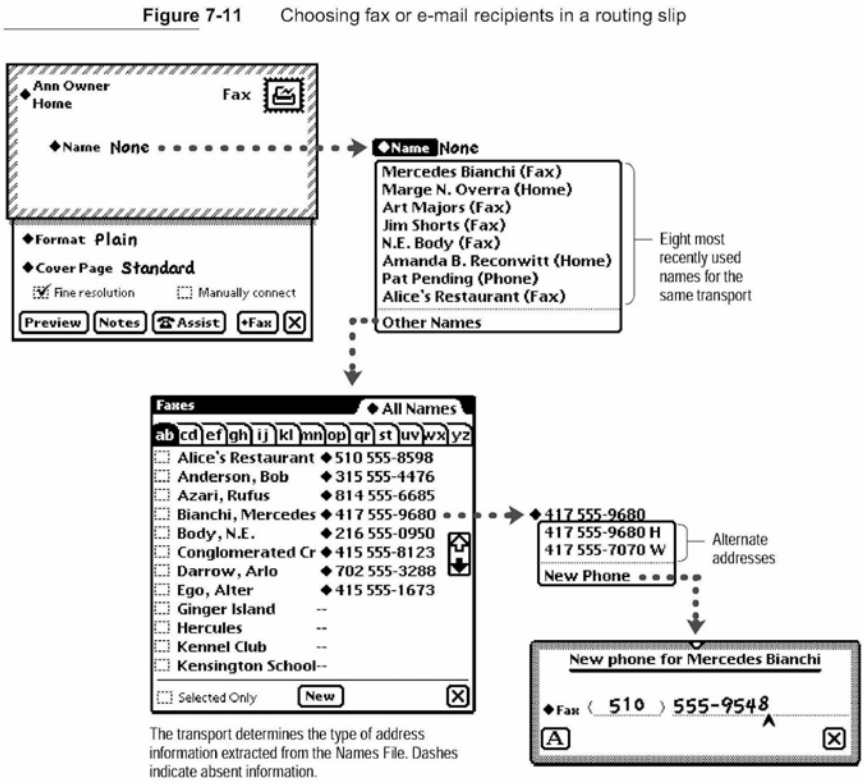
further comprising, providing a prompt for updating the information source to include the first information.	Newton further discloses this element. For example, once the call has been completed, tapping on the far-right button on the row of buttons along the bottom of the screen (highlighted in black in the photograph below), results in a prompt for the user to store the name and phone number as a new Name in the Names database.  See also Newton Guidelines at fig. 7-11:

Exhibit I



See also Newton Guidelines at p. 8-27: "Task Slips.

Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip.

For example, if a user write the request 'fax Bob,' the Assistant can get Bob's fax number from the Names File application. But what if bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page.

The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86)."

Exhibit I

	"And if a contact with that name does not already exist, what would the user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16. "And then is the user able to then save information to the Name File after making the call? A. Um, yes. * * * Yes. Sorry, my – I got sidetracked, but yes. So the same UI that we, um, happen before when we try the Jennifer Lopez case, happens in this case, after I entered the number, because, um, I didn't have a phone number So I had to enter the phone number. And once the phone number and I place the call, the UI to add it to the Names File is presented to the user. Q. Okay. And then so the user is presented with an option to add this new contact to the new file? Okay. A. Yes." Pagallo Depo at 92:21-93:11. "And then how does the user then add the – this new contact to the Name File? Is there an icon or button that's add or – A. There is a UI that has the name card with the name, the phone number, and it says, 'Would you like to add a new name with the information,' and then you can say – select yes or no. And if I select yes, then that new card is added to the Name File." Pagallo Depo at 93:12-19. "So the screen that shows for the user that you were just describing – A. Uh-huh. Q. – where you can see the specifications that are being used for the phone call, does the user have the opportunity to change any of the information that's presented in the screen, so if the phone number – if the user realizes that this phone number for Bob is incorrect and wants to change the phone number that's being used, does the user have the

Exhibit I

	opportunity to do that? A. Yes. The user has the opportunity to tap a name and select a different name or select a different entry." Q. Okay. And that's presented to the user as options before a call – A. Correct. Q. – is executed?" Pagallo Depo at 97:5-22. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant. However, the name "Michael" did not result in the display of any other information associated with the name "Michael." I was then able to tap the "Name" field in the "Phone" window with the stylus. A new "Phones" window then appeared with a "New" button, which I tapped with the stylus. A "New Person" window appeared that allowed me to enter first name, last name, and phone number for the new contact card. The last name ("Jordan") and phone number ("1 800 1112222") were associated with the first name ("Michael") in this contact card and added to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card).

Exhibit I

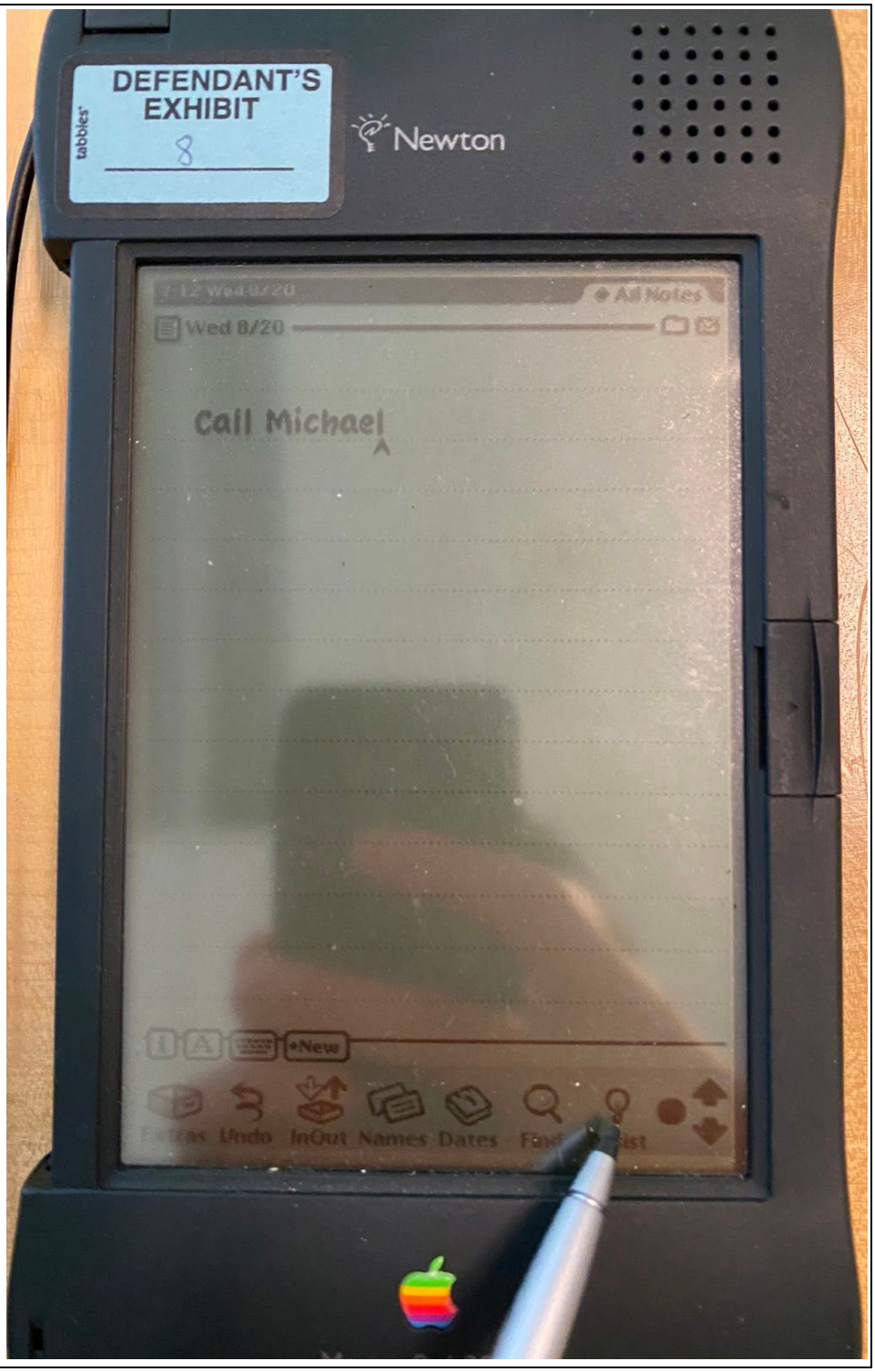

DEFENDANT'S
EXHIBIT
8
Newton
Wed 8/20
All Notes
Call Michael
New
Undo
InOut
Names
Dates

Exhibit I

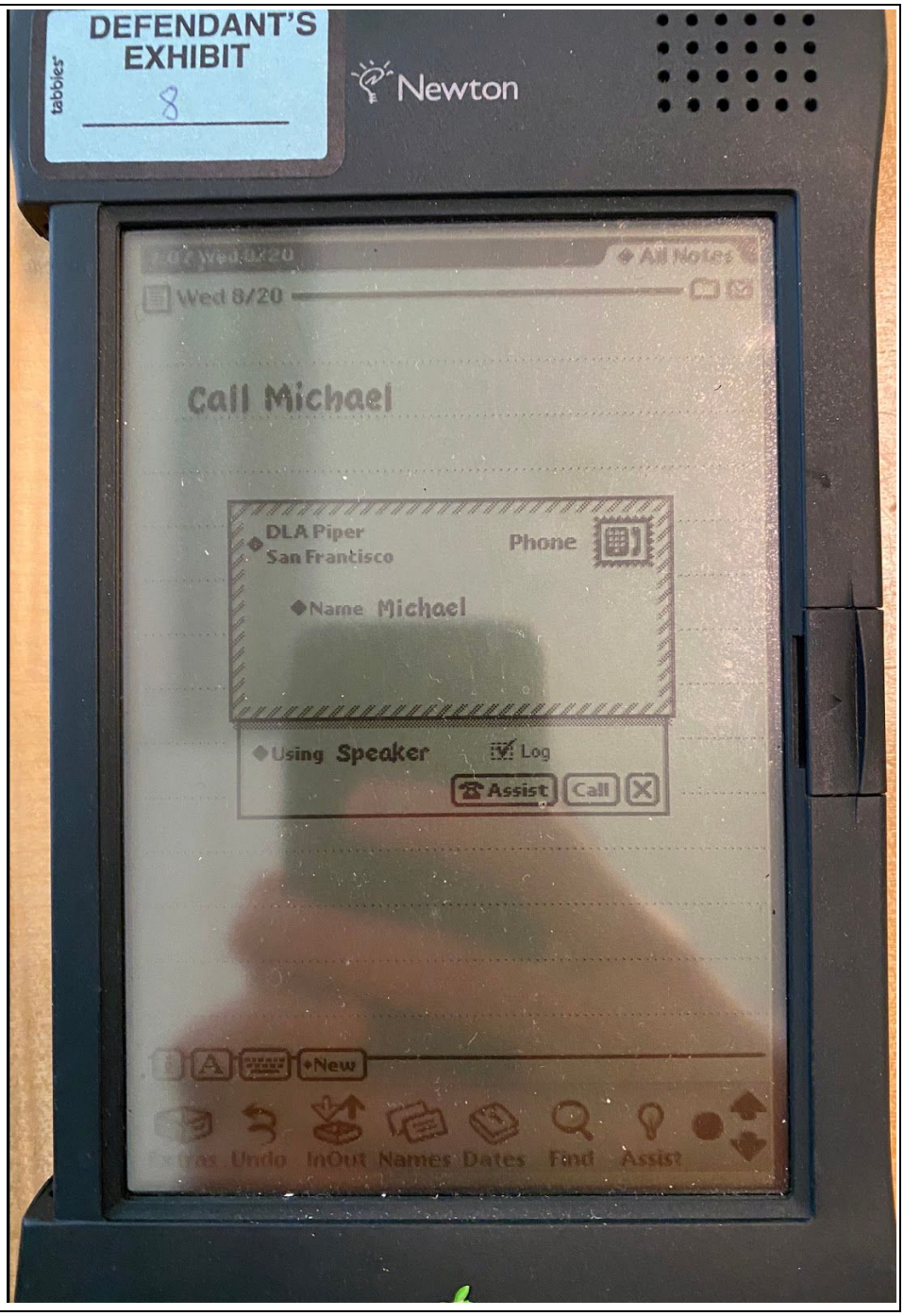

DEFENDANT'S
EXHIBIT
8
tabbies
Newton
Wed 8/20
All Notes
Call Michael
DLA Piper
San Frantisco
Phone
Name Michael
Using Speaker
Log
Assist
Call
New
Undo
InOut
Names
Dates
Find
Assist

Exhibit I

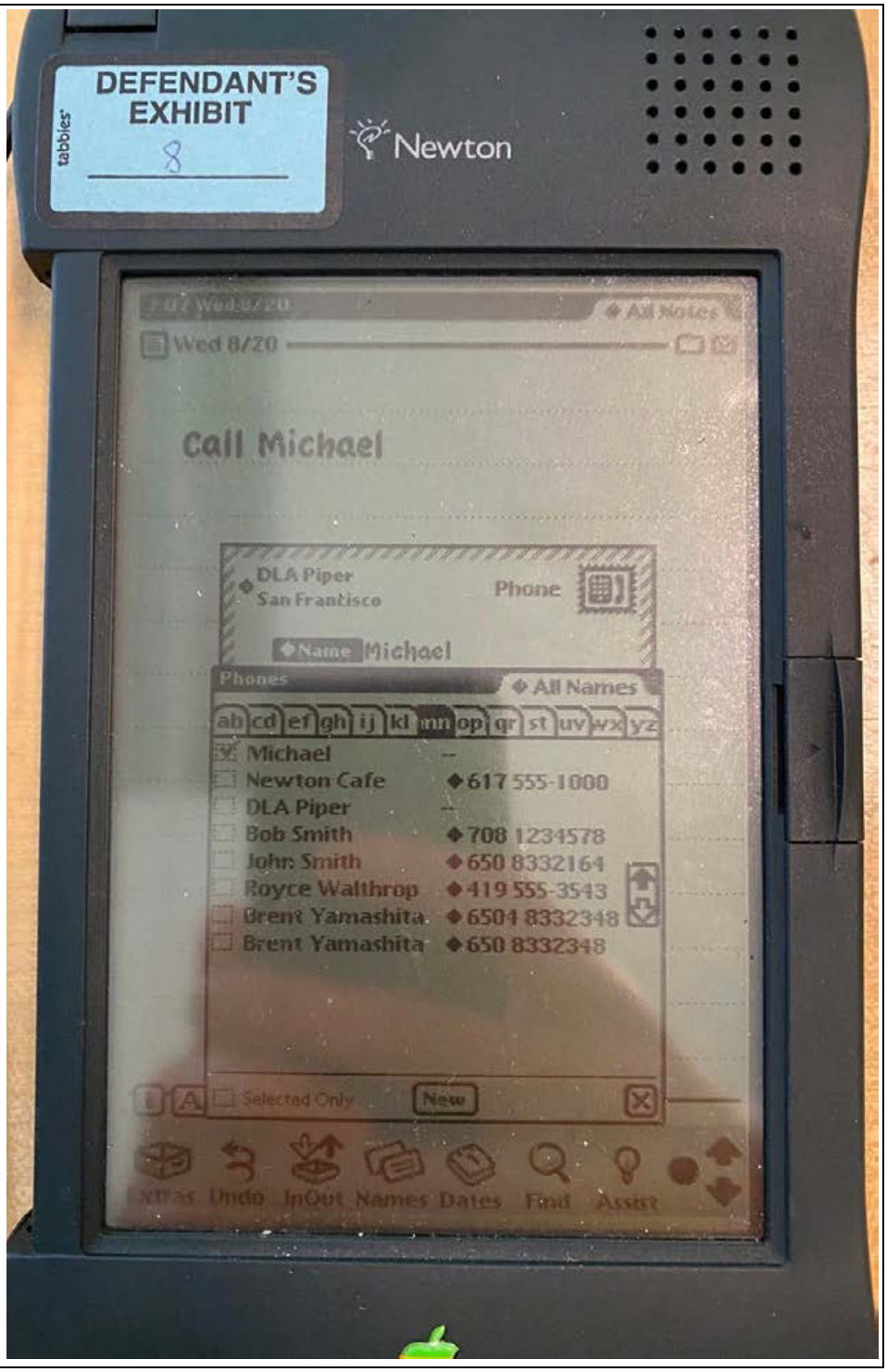

DEFENDANT'S
EXHIBIT
8
tabbies
Newton
Wed 8/20
All Notes
Call Michael
DLA Piper
San Francisco
Phone
Name Michael
Phones
All Names
ab cd ef gh ij kl mn op qr st uv wx yz
Michael
Newton Cafe ◆617 555-1000
DLA Piper
Bob Smith ◆708 1234578
John Smith ◆650 8332164
Royce Walthrop ◆419 555-3543
Brent Yamashita ◆6504 8332348
Brent Yamashita ◆650 8332348
Selected Only
New
Undo
InOut
Names
Dates
Find
Assist

Exhibit I

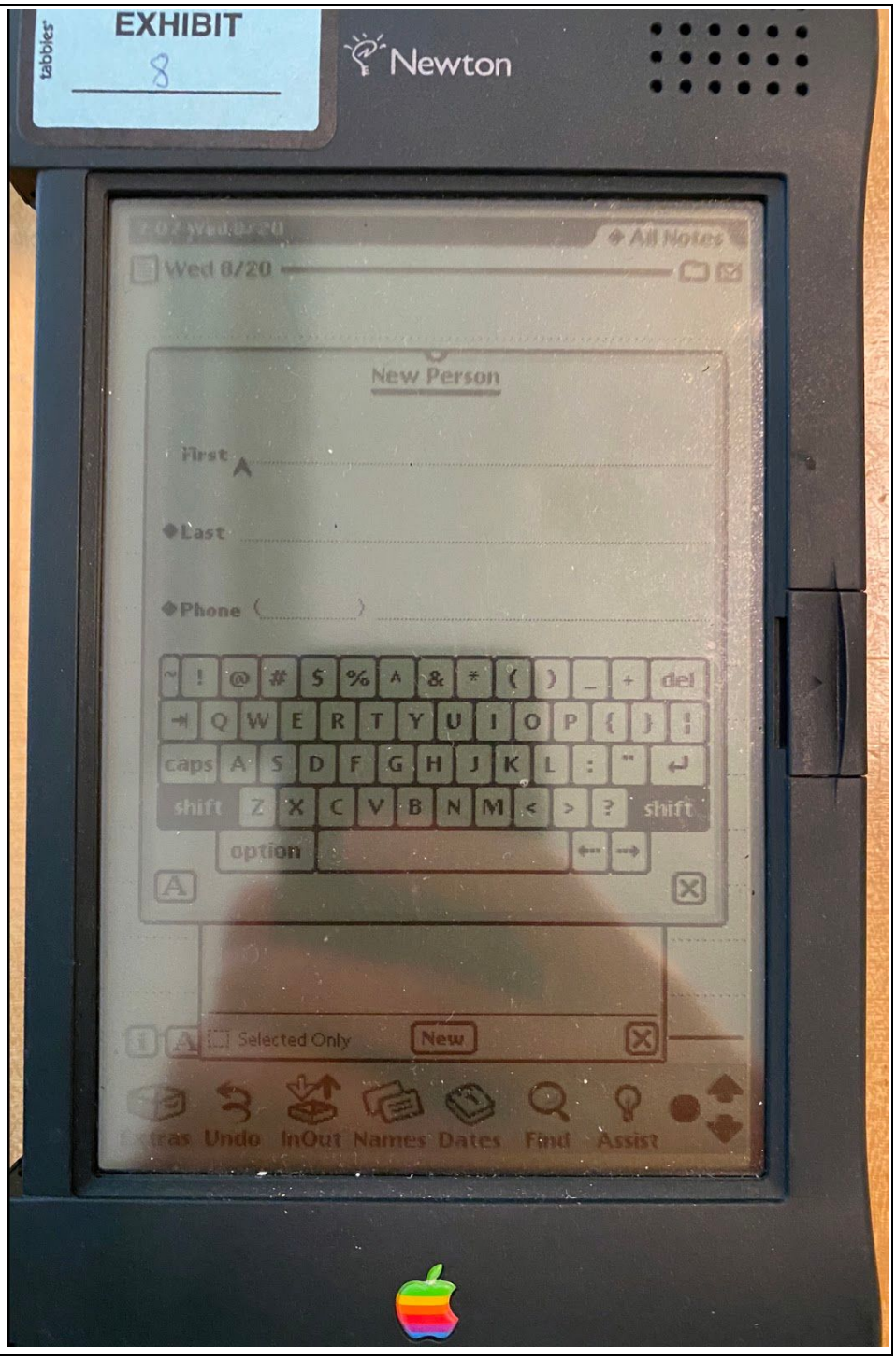


EXHIBIT
8
Newton
Wed 8/20
All Notes
New Person
First
Last
Phone
Selected Only
New
Undo
InOut
Names
Dates
Find
Assist

Exhibit I

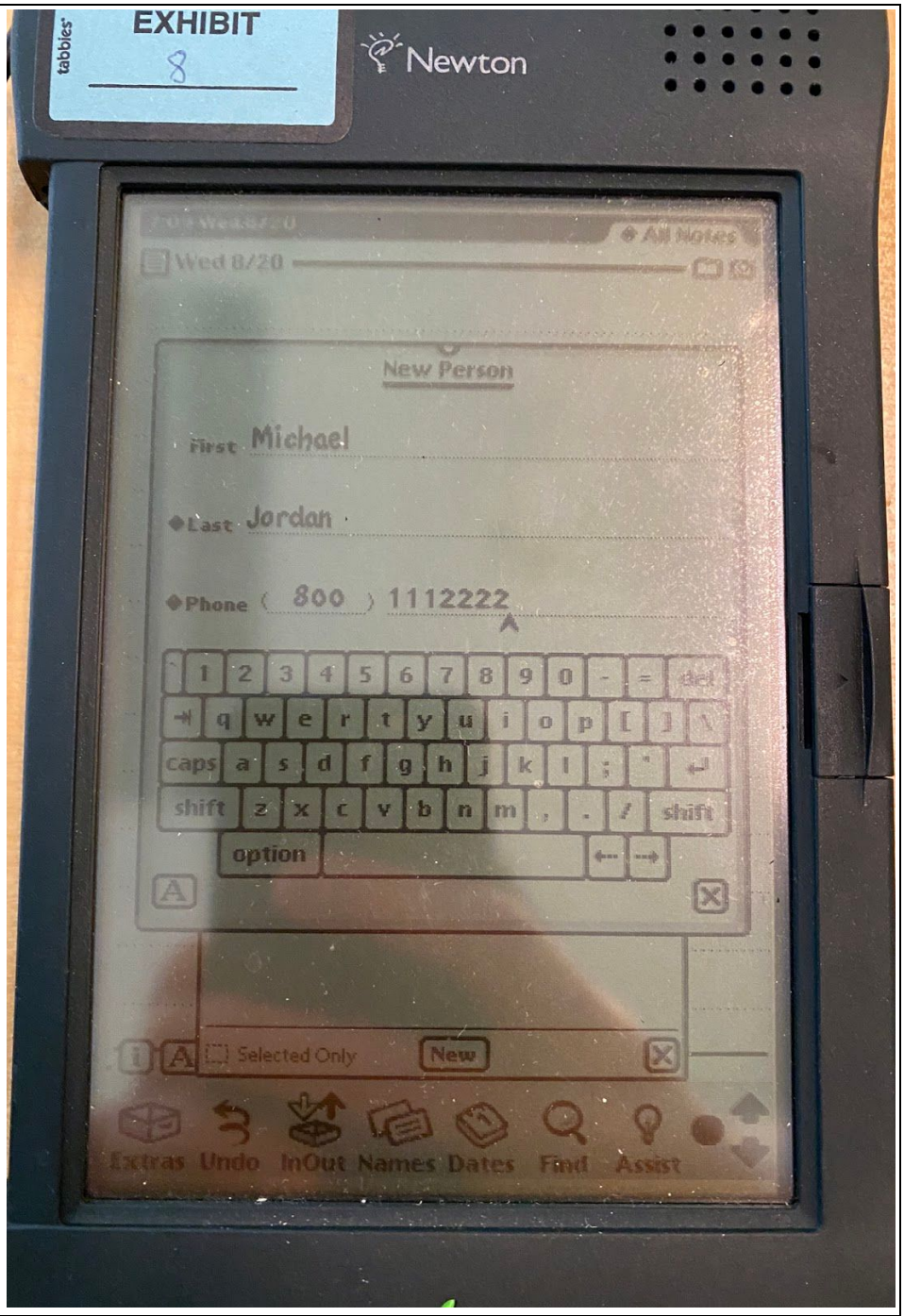
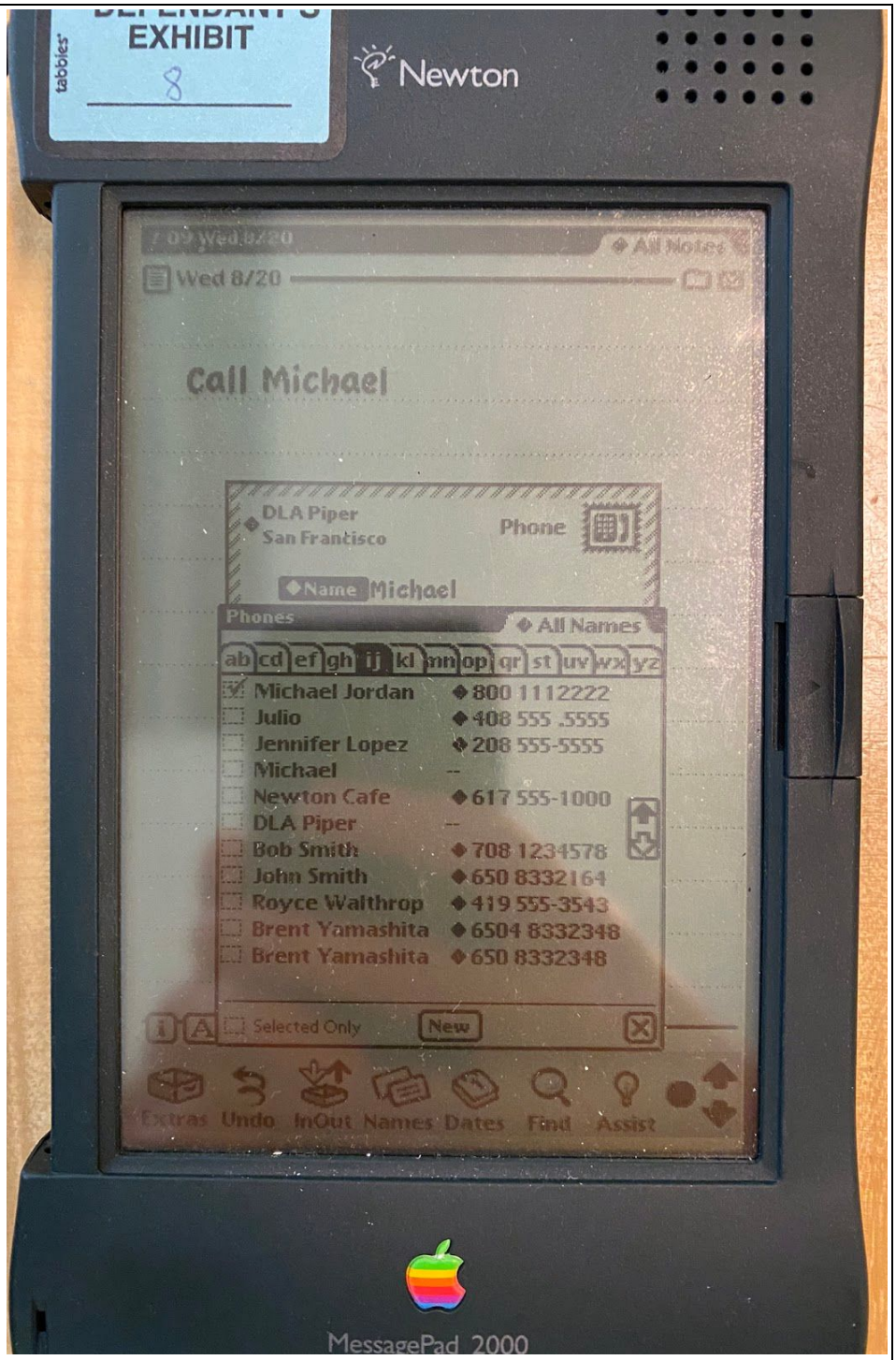


EXHIBIT
8
Newton
Wed 8/20
All Notes
New Person
First Michael
Last Jordan
Phone (800) 1112222
Selected Only
New
Extras
Undo
InOut
Names
Dates
Find
Assist

Exhibit I



For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 5. Newton and the prior art references and systems set forth in Exhibit U, Tables 5 (e.g., Miller, Pandit), relate to the same field of art, which is personal

Exhibit I

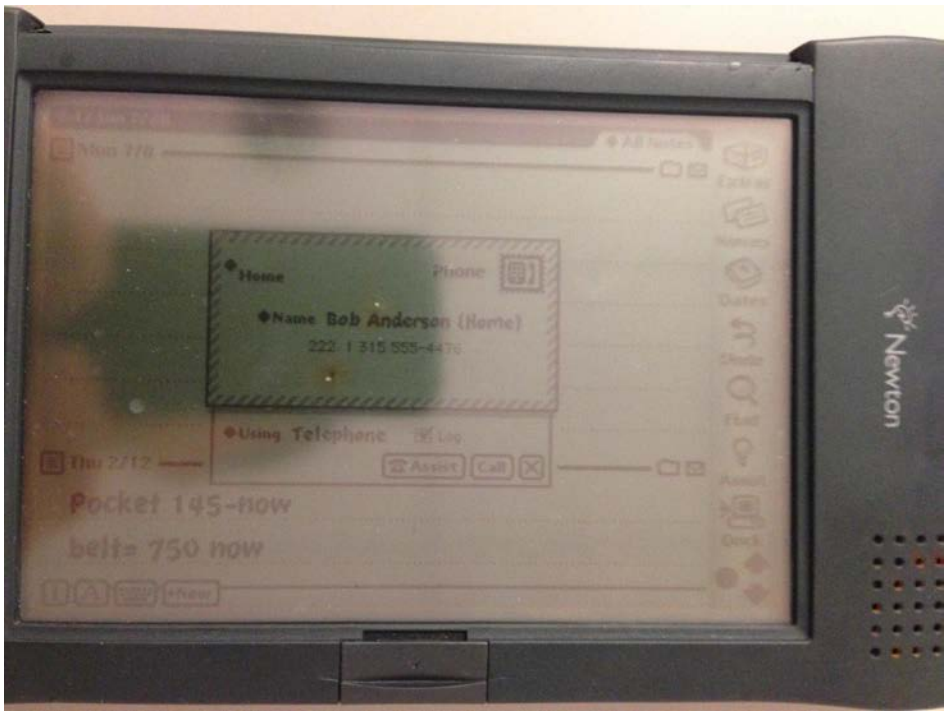
	communication devices and methods. Specifically, Newton and certain prior art references and systems set forth in Exhibit U, Table 5 (e.g., Miller, Pandit) relate to managing information in an address book/contact database. A POSITA would have reasonable expectation of success to achieve predictable results in combining Newton and the prior art references and systems in Exhibit U. A POSITA would have recognized advantages and benefits to have Newton provide a prompt for updating the information source to include the first information.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	Newton discloses claim 1. *See* claim 1 above. Newton further discloses this element. For example, tapping the "Assist" button launches the Intelligent Assist feature with text from the document. The photograph below illustrates the display of the Newton device after the user has entered the text "call bob" in the Notes application and then tapped the "Assist" button. As illustrated in the photograph below, Newton searches for second information (in this case a full name and phone number) associated with the name "bob" in the Names application, and displays this information in a dialog box.  *See also* screenshots taken during my Inspection.

Exhibit I

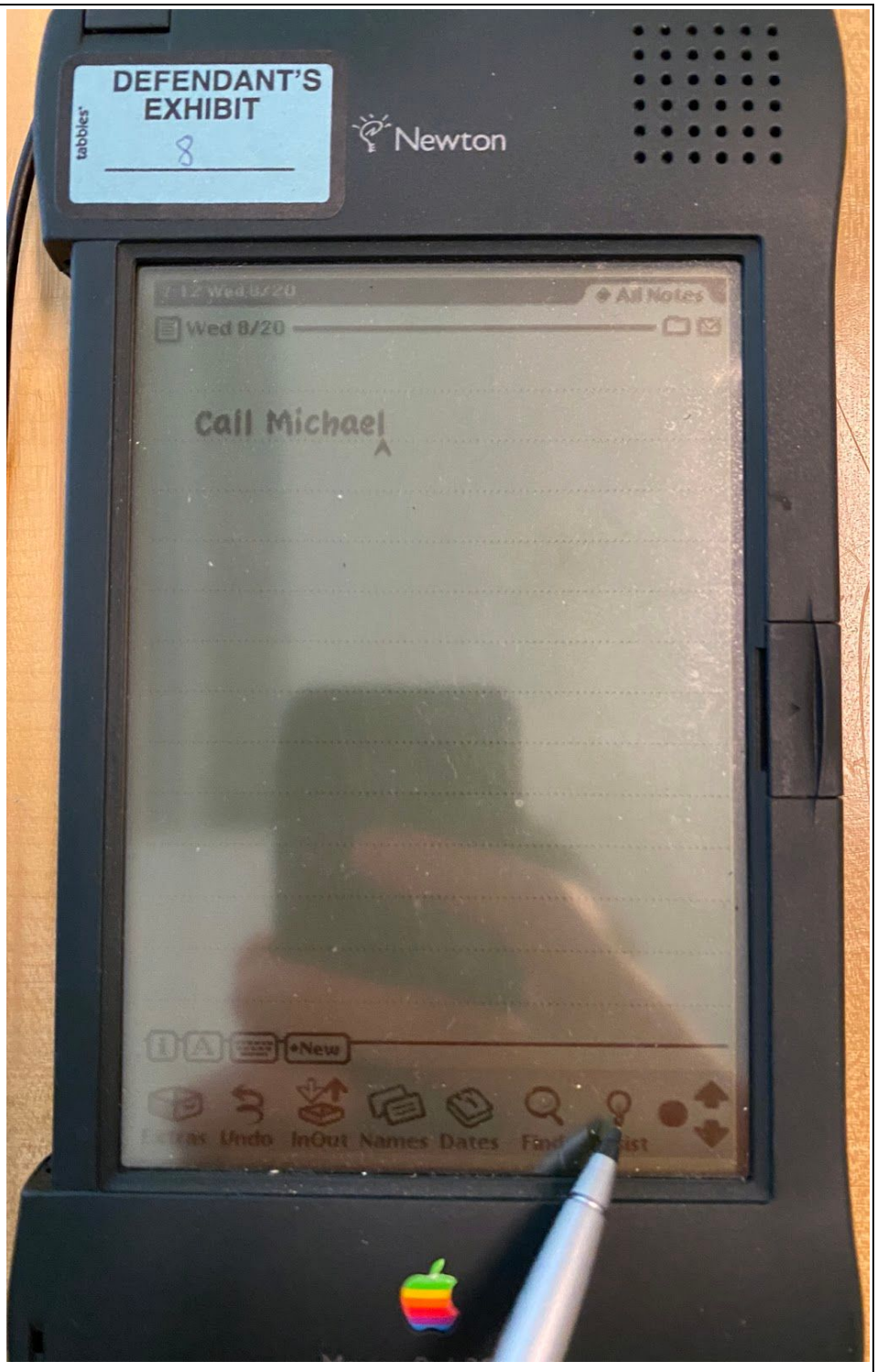

DEFENDANT'S
EXHIBIT
8
Newton
Wed 8/20
All Notes
Call Michael
New
Undo
InOut
Names
Dates

Exhibit I

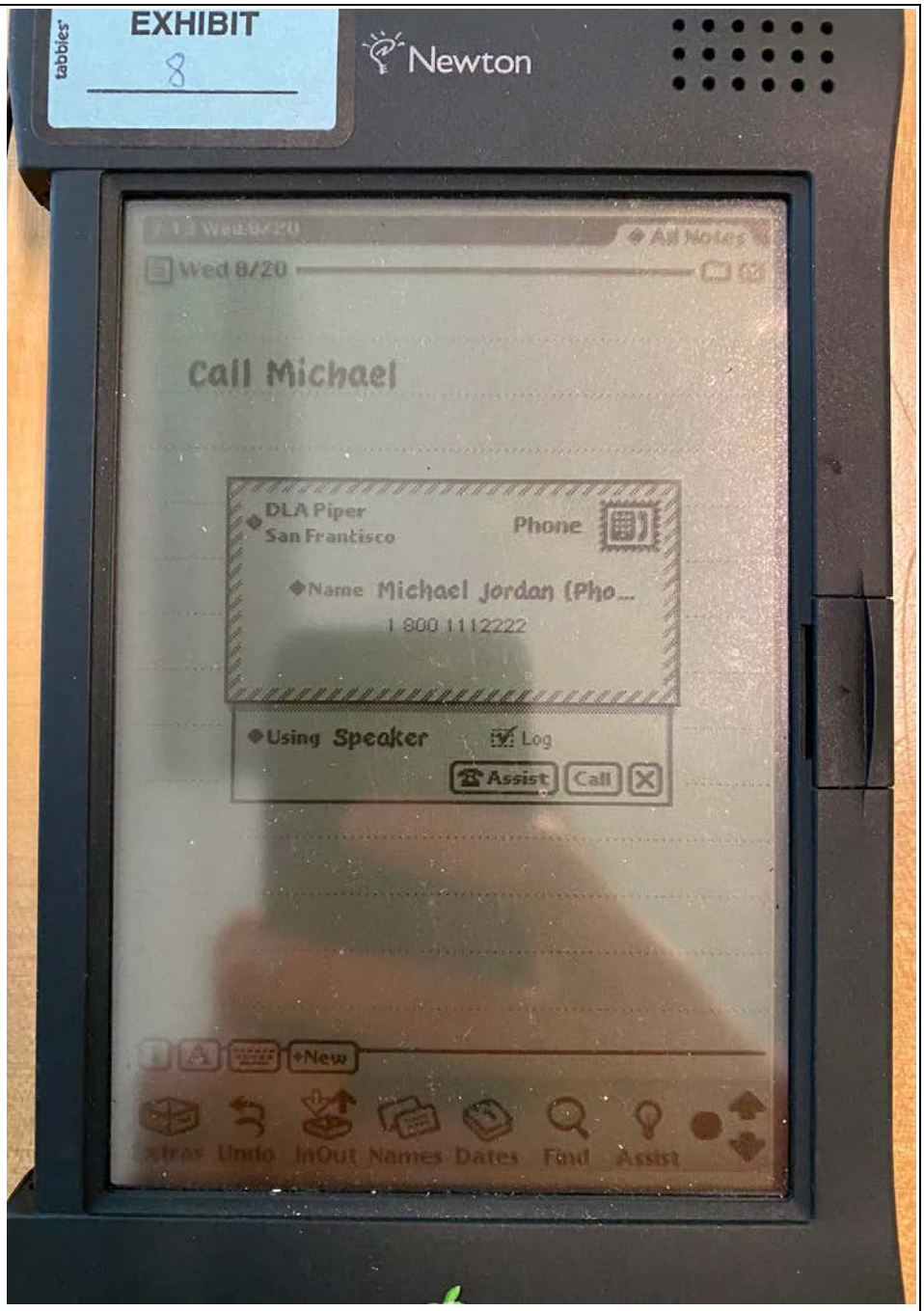
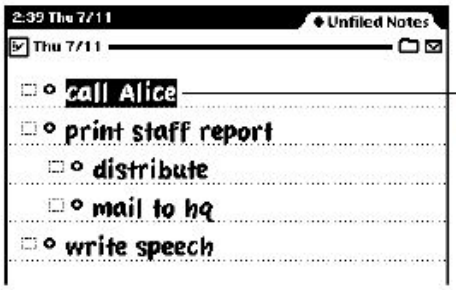
	 Newton Manual states:

Exhibit I

Writing your request

1 Write at least one word of your request, beginning with a request word or one of its synonyms.

2 Select the request. Place the pen on or near the item until a heavy mark appears under the pen. Then draw the highlighting mark over or around the item.



Newton Manual, p. 195.

3 Tap Assist.

The MessagePad interprets the request and a confirmation slip appears with some information already entered, based on your request.

4 Enter other necessary information.

5 When you're finished, tap the button near the bottom of the slip to perform the action. Tap Call, for instance, to place the call.

Newton Manual, p. 196.

"Using the correct request words

The MessagePad understands the following requests and their synonyms. (If you have other applications installed on your MessagePad, the other applications may recognize additional request words.)

Exhibit I

	● Call to dial a telephone number. Synonyms: phone, ring, dial ***Call Bob at home*** looks in the Name File to find Bob's home phone number, then puts it in the call slip. ● Fax to fax the item on the screen. Synonyms: none ***Fax Anderson*** opens a fax slip with the name Anderson and Anderson's fax number filled in. …" Newton Manual, pp. 196-97. Newton Guide states: "When the user invokes the Assistant, the system passes the current text selection to it. If no text is selected, the system passes to the Assistant the contents of a buffer that holds the most recent text input. The Assistant then attempts to match words in the input string with templates and dictionaries that classify the words as actions, targets, or unknown entities." Newton Guide, p. 18-1. "When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup. The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose." Newton Guide, p. 18-4. See also Newton Guidelines at pp. 1-11: "Remember that people will use Newton while on the move, in places where there's no place to sit or to set it down. In such settings, it's easiest to use applications with simple, straightforward screens and an obvious path through the information. Make sure that your application's controls are clearly identified, that there aren't too many 'places' for the user to navigate through, that you don't display to many container views at once, and that the user can easily see what to do. Minimize writing; tapping to pick from a list of

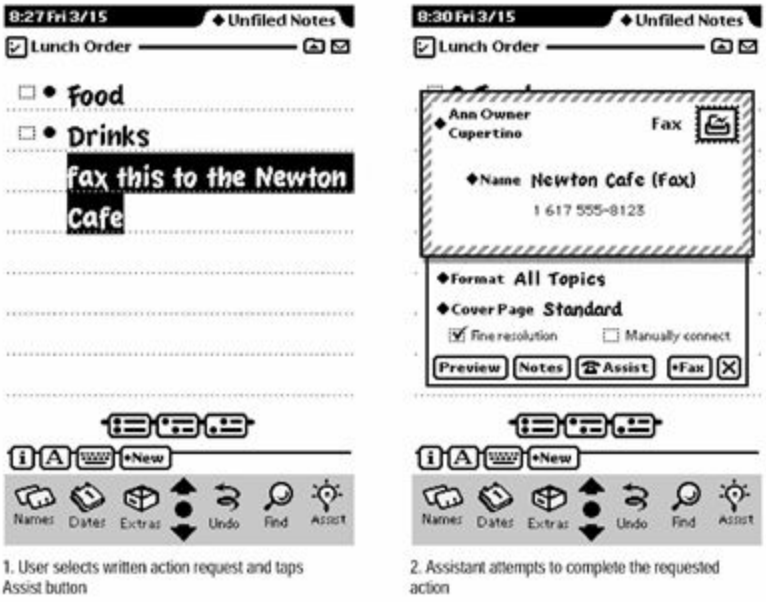
Exhibit I

alternatives is easier."

See also Newton Guidelines at 2-24, 3-1, 3-2.

See also Newton Guidelines at pp. 2-22-2-23: "A user invokes the Assistant by tapping the Assist button, which is a picture button that looks like a light bulb. Before tapping the Assist button, a user can write an action request or select existing text to be used as a request. When a user taps the Assist button the system passes the Assistant any currently selected text. If no text is selected, then the system passes the Assistant the most recently written text. The Assistant classifies the words in that text as actions, targets of actions, or unknown entities. Depending on the results of this analysis, the Assistant may attempt to complete a task or it may prompt the user to supply additional information. Figure 8-22 shows the Assist button initiating a fax operation.

Figure 8-22 The Assist button makes the Assistant try a written action request



"

"The everywhere and Selected buttons specify that the system perform searches in applications other than the currently active one. Applications must register with the Find service to participate in such searches.

Tapping the Everywhere button tells the system to conduct searches in all currently available applications registered with the Find service. This kind of search is called a Global find. Applications need not be open to participate in a Global find.

Exhibit I

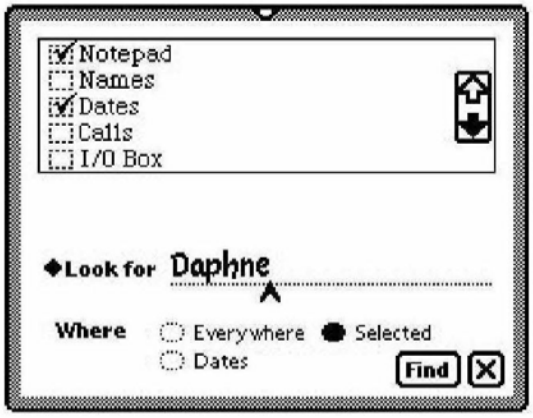
A Global find is similar to a series of Local find operations initiated by the system. When the user requests a Global find, the system executes a Local find in each application registered with the Find service.

Tapping the Selected button causes a checklist to appear at the top of the Find slip. The list includes all currently available applications registered with the Find service. Tapping items in the list places a check mark next to applications in which the system should conduct a Local find. This kind of search is called a Selected find. The slip in Figure 16-4 depicts a search for the string 'Daphne' in the Notes and Dates application."

Newton Guide at 16-3.

See also Newton Guide at Figure 16-4.

Figure 16-4 Searching selected applications



About the Find Service

"After setting the parameters of the search with the Find slip, the user initiates the search by tapping the Find button."

Newton Guide, p. 16-4.

"When the search is completed, the Find service displays an overview list of items that match the search criteria. Figure 16-6 shows the Find overview as it might appear after searching all applications for the string 'Daphne'."

Newton Guide, p. 16-4.

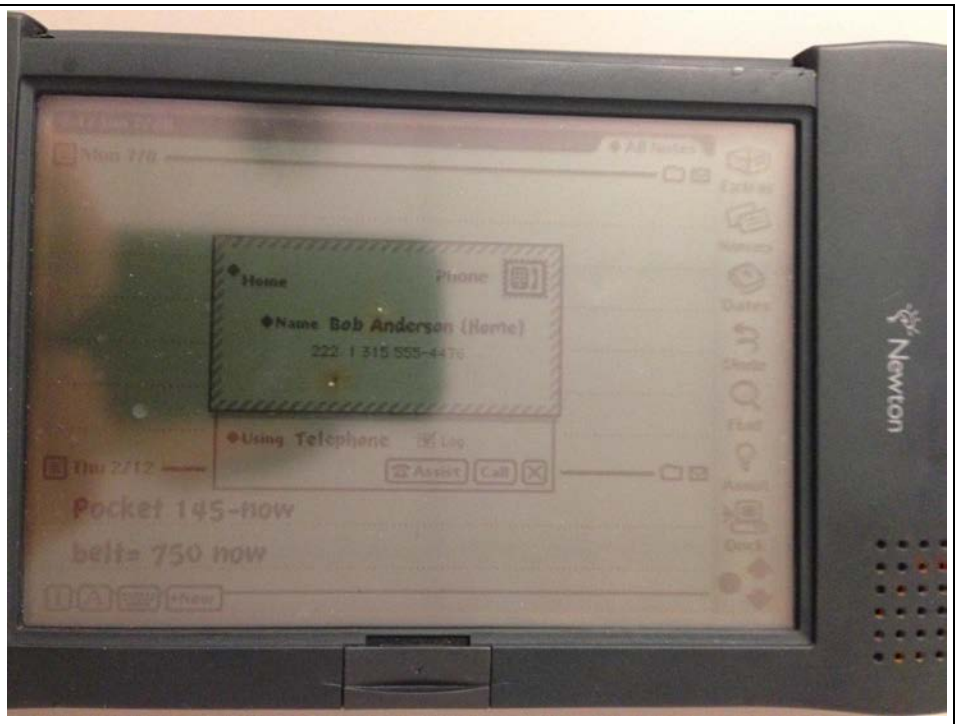
Exhibit I

	See also Newton Guide at Figure 16-6. "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using – A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh. Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in

Exhibit I

	text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * * A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work. Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant. Newton then searched the Names application contact cards (or, more specifically, the "Names soup," which contains the data for each contact card) for information associated with the name "Michael." A "Phone" window then appeared to include the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). See also screenshots from the "while the document is being displayed" element above. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	Newton discloses claim 1. *See* claim 1 above. Newton further discloses this element. For example, tapping the assist button results in the display of a dialog box that provides various menus each indicated by a diamond symbol. As illustrated in the photograph below, the dialog box includes menus for selecting a different phone number for the given person ("Home"), selecting a different person to call ("Name") or selecting the means for performing the call ("Using.")

Exhibit I



Tapping the diamond symbol next to “Name” displays a list of contacts and allows the user to select which contact to call as illustrated in the photograph below.

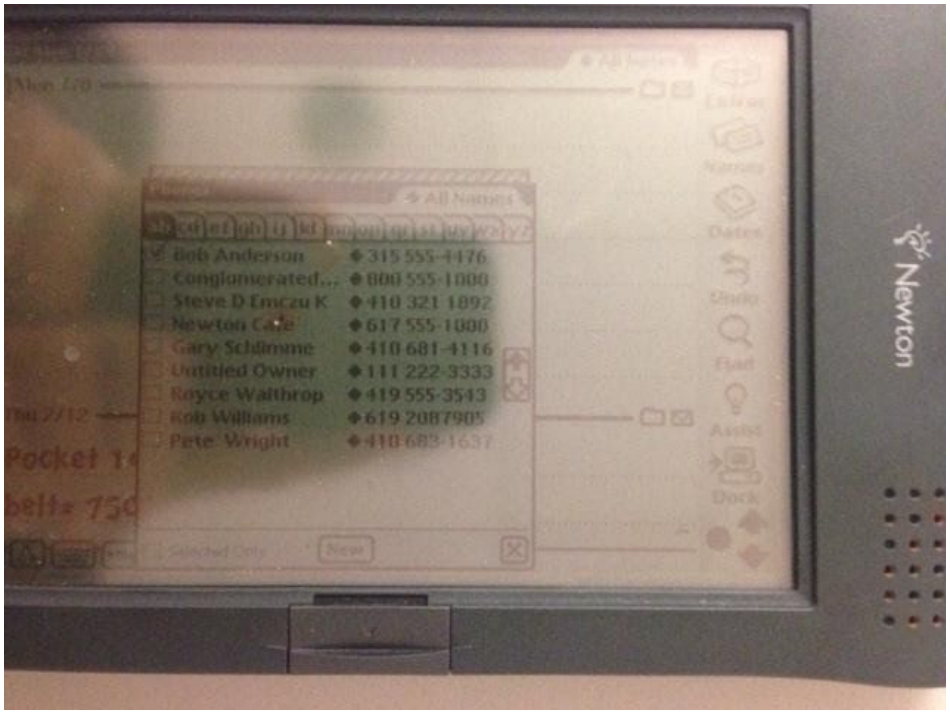


Exhibit I

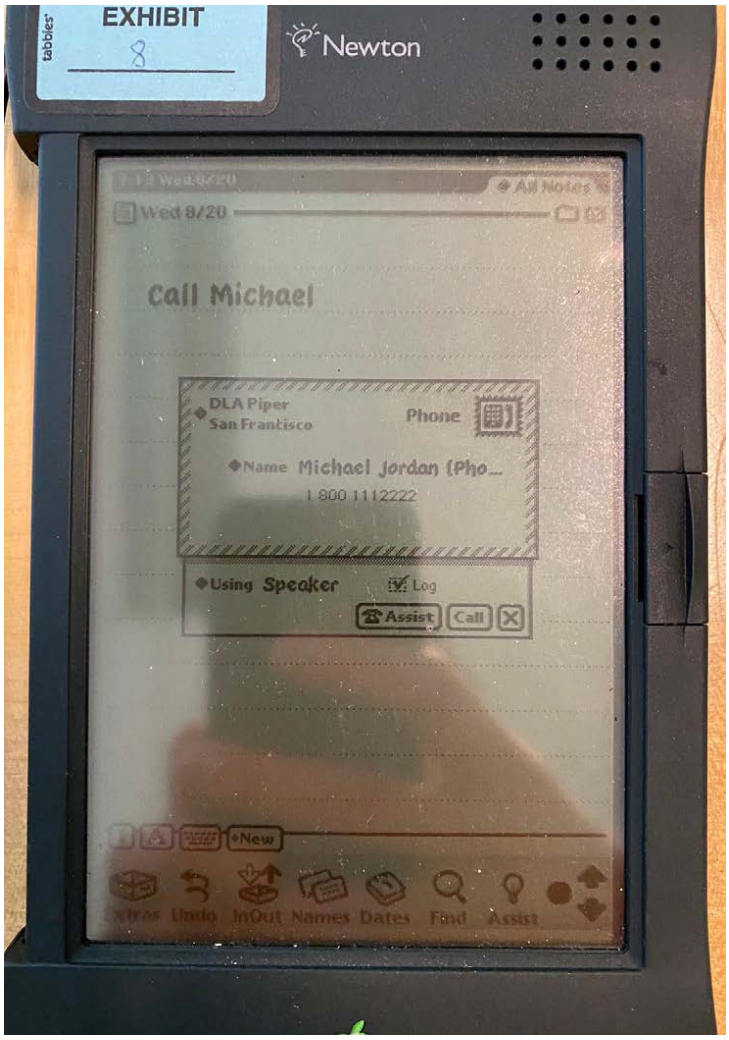
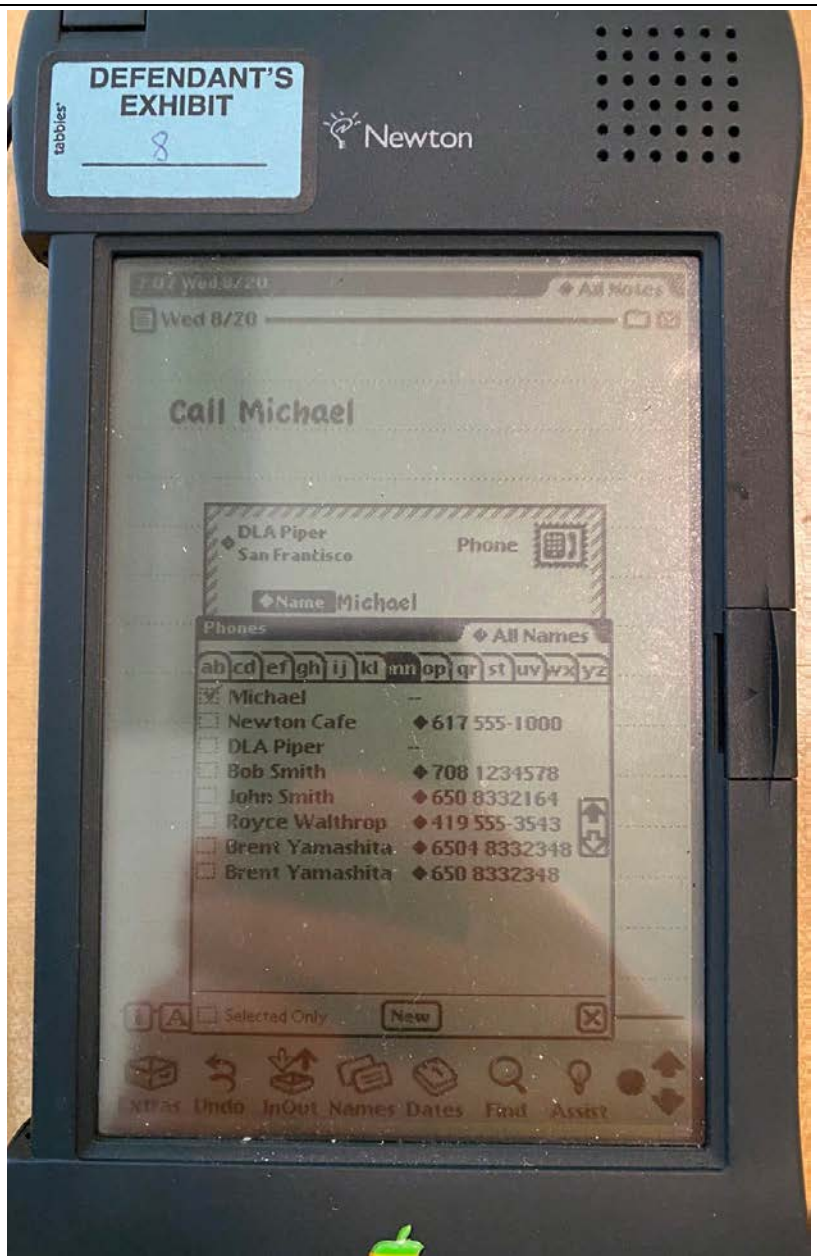
	See also screenshots taken during my Inspection. 

Exhibit I



Newton Guide states:

"When the search is completed, the Find service displays an overview list of items that match the search criteria. Figure 16-6 shows the Find overview as it might appear after searching all applications for the string 'Daphne'."

Newton Guide, p. 16-4.

See also Newton Guide at Figure 16-6.

Exhibit I

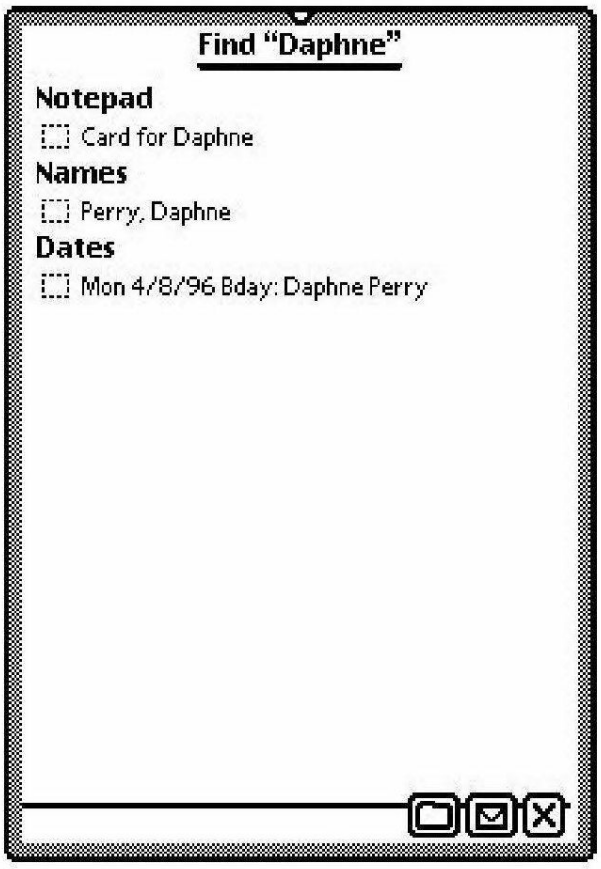
	Figure 16-6 The Find overview  "The user can tap items in the Find overview to display them. As items are displayed, a status message at the top of the Find slip indicates which item is displayed and whether there are more results to display. Figure 16-7 depicts this status message Newton Guide, p. 16-5 See also Newton Guide at Figure 16-7.

Exhibit I

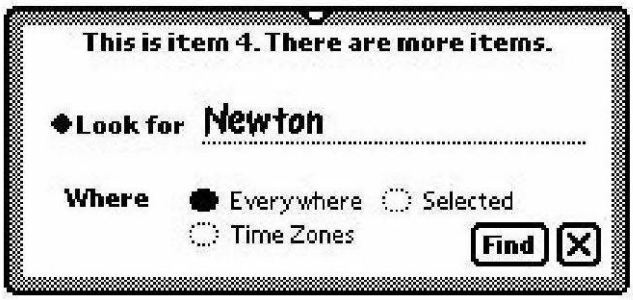
	Figure 16-7 Find status message  "When more than one item is found, the status message indicates that there are more items to display. Between uses, the Find service stores the setting of the Look For picker. The next time this service is used, it reopens in the most recently set find mode. Note that in order to conserve memory, the list of found items is not saved between finds." Newton Guide, p. 16-5. "When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup. The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose." Newton Guide, p. 18-4. See also Newton Guidelines at p. 2-36: "An application that deals with multiple instances of similar information--multiple notes in the Notepad, multiple names in the Name File, multiple days in the Date Book, and so on--can't display all the instances at once in a single view. People scroll the information to move currently displayed information out of view and bring other information into view. The information appears to roll out at one edge of the view and roll in at the opposite edge. Figure 2-29 shows a conceptual view of notes ready to be scrolled in the Notepad's main

Exhibit I

	view." See also Newton Guidelines at p. 8-10: "Keep in mind that a user may need to scroll among found items while the Find slip is displayed; therefore, when customizing or replacing this slip, avoid making it so large that it obscures the display of the found items. Figure 8-9 shows a sample application named Checkbook that adds a labeled input line with a picker to the standard Find slip." See also Newton Guidelines at p. 8-11: "If a search finds just one item, that item is displayed behind the Find slip. If a search finds more than one item, the Find service displays an overview list of the found items. Figure 8-11 depicts a Find overview as it might appear after searching all applications for 'man.'" See also Newton Guidelines at p. 8-12: "A user can alternately hide and reveal the names of items listed under an application int he Find overview by tapping the application name there. Tapping the name of an item in the Find overview displays a detail view of the item. The Find slip stays open in front of the detail view. A message at the top of the Find slip states which item is displayed and whether other items were found in the same application. Figure 8-12 shows an example of the message." See also Newton Guidelines at p. 8-13: "If more than one item was found, tapping the universal down arrow goes to the next found item, and tapping the universal up arrow goes to the previous found item. Tapping the Overview button redisplays the overview of found items." See also Newton Guidelines at figs. 8-10, 8-11, 8-12. See also Newton Guidelines at p. 8-27: "Quite often the Assistant knows what action to take but does not have enough information to complete that action. The Assistant tries to fill in as much of the required information as it can, but the user may still have to resolve ambiguities or provide additional information. In that case the Assistant displays a task slip. For example, if a user writes the request 'fax Bob,' the Assistant can get Bob's fax number from the Names File application. But what if Bob has no fax number or more than one fax number? What if there is more than one Bob or no one named Bob? Even if there is only one Bob with one fax number, the user may want to add another fax number, another Bob, or a message on the fax cover page.

Exhibit I

	The task slip for any built-in action is the same as or similar to the slip a user sees when performing the action without the Assistant. For routing actions--printing, faxing, mailing, or calling--the task slip is a routing slip (see 'Routing Slips' on page 7-12). For scheduling meetings and remembering To Do items, the task slip is similar to the slip the Date Book application displays for creating a new meeting or To Do task. For finding, the task slip is the standard Find slip (see 'Find' on page 86)." For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 7, 17, and 20 (e.g., Luciw 735).

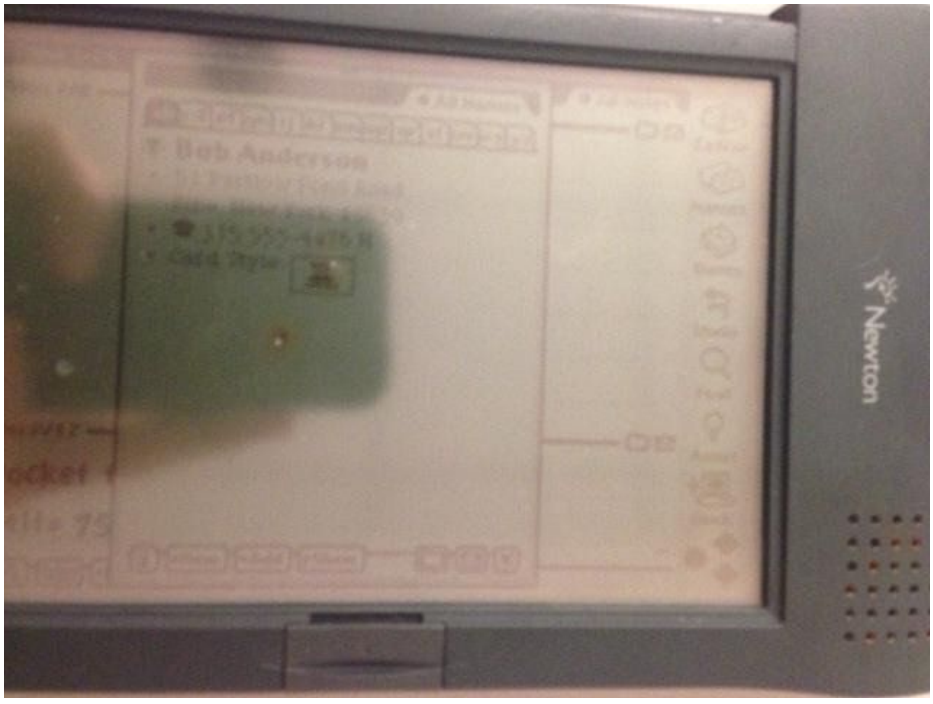
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	Newton Guide discloses claim 1. *See* claim 1 above. Newton further discloses this element. For example, Newton provides access to a contacts database that can be separately accessed and edited using the Names application. As illustrated in the photograph below, the Names application shows at least three fields of information associated with the contact (in this case, a name, phone number and address).  *See also* screenshots taken during my Inspection.

Exhibit I

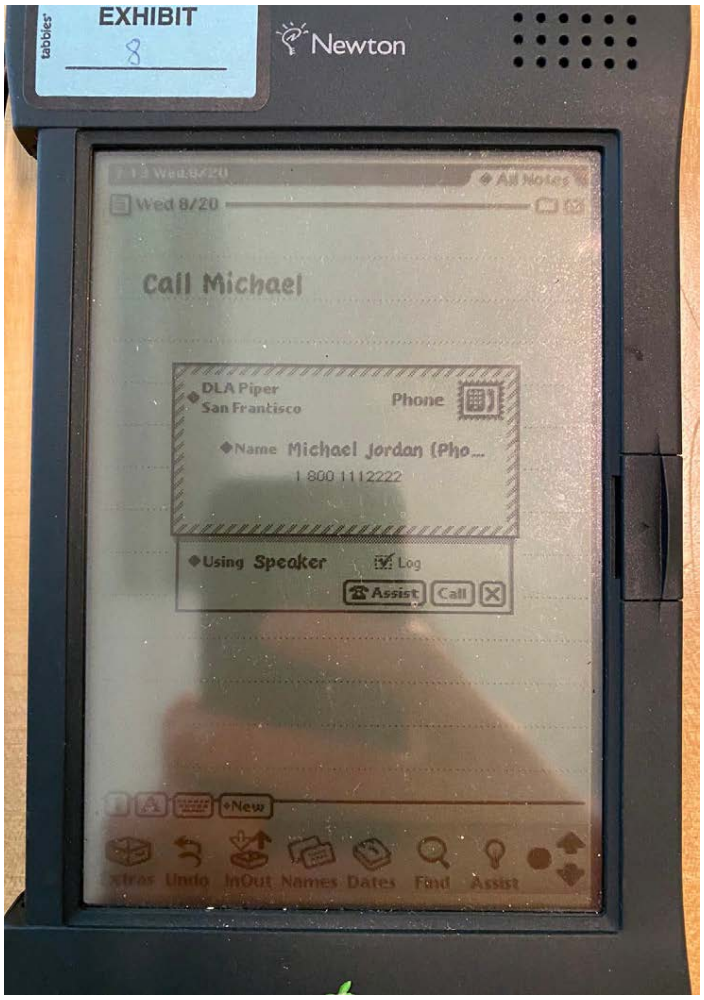
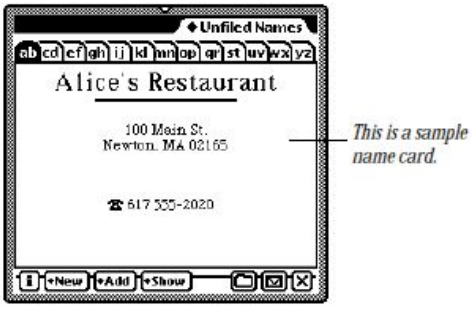
	 Newton Manual states:

Exhibit I

Using the Name File

You can use the Name File as an address book to store information about people, companies, and groups. The Name File contains name cards that you create. Each card has information such as name, address, telephone numbers, electronic mail addresses, and notes. You can also create your own field labels for special information.

Tap Names to go to the Name File. Tap it again to put away the Name File. You can also tap in the lower-right corner of the Name File to put it away.



Newton Manual, p. 55; *see also* pp. 50-51 (creating new cards), 52-54 (adding information to a card), 54 (changing information in an existing card), 56-57 (viewing) cards.

Newton Guide states:

"The everywhere and Selected buttons specify that the system perform searches in applications other than the currently active one. Applications must register with the Find service to participate in such searches.

Tapping the Everywhere button tells the system to conduct searches in all currently available applications registered with the Find service. This kind of search is called a Global find. Applications need not be open to participate in a Global find.

A Global find is similar to a series of Local find operations initiated by the system. When the user requests a Global find, the system executes a Local find in each application registered with the Find service.

Tapping the Selected button causes a checklist to appear at the top of the Find slip. The list includes all currently available applications registered with the Find service. Tapping items in the list places a check mark next to applications in which the system should conduct a Local find. This kind of search is called a Selected find. The slip in Figure 16-4 depicts a

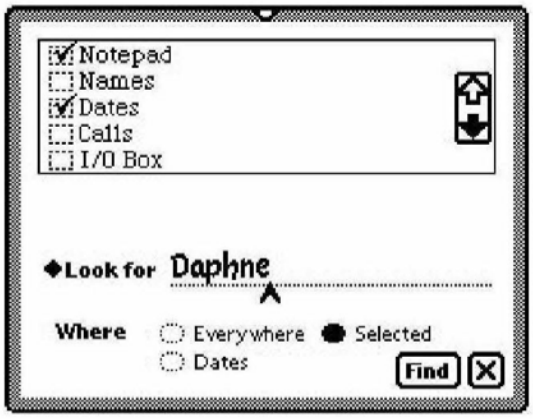
Exhibit I

search for the string ‘Daphne’ in the Notes and Dates application.”

Newton Guide at 16-3.

See also Newton Guide at Figure 16-4.

Figure 16-4 Searching selected applications



About the Find Service

“After setting the parameters of the search with the Find slip, the user initiates the search by tapping the Find button.”

Newton Guide, p. 16-4.

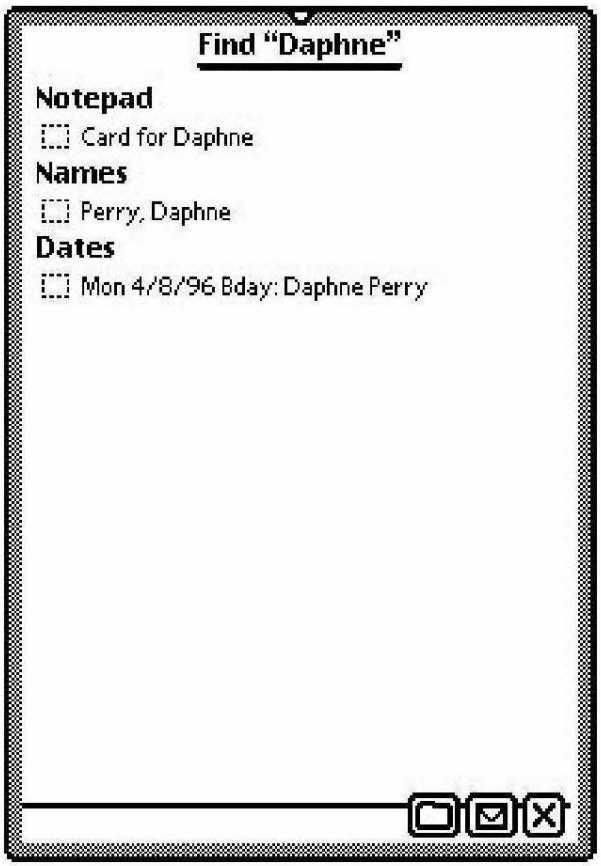
“When the search is completed, the Find service displays an overview list of items that match the search criteria. Figure 16-6 shows the Find overview as it might appear after searching all applications for the string ‘Daphne’.”

Newton Guide, p. 16-4.

See also Newton Guide at Figure 16-6.

Exhibit I

Figure 16-6 The Find overview



"When an action is specified but required information is still missing, the Assistant tries to supply as much of the required information as possible. For example, if the input string is 'fax bob', the Assistant can query the Names soup for information such as Bob's name and fax number. However, the user may still need to correct the input if the Assistant chooses the wrong Bob from the Names soup, cannot find Bob in the Names soup, or cannot find Bob's fax number in this soup.
The user can resolve ambiguities or provide additional information from within a task slip that the Assistant displays for this purpose."

Newton Guide, p. 18-4.

"Names

This section describes the application program interface (API) to the Names application. The Names application manages a database of people and places. It presents information either as a business card, or as

Exhibit I

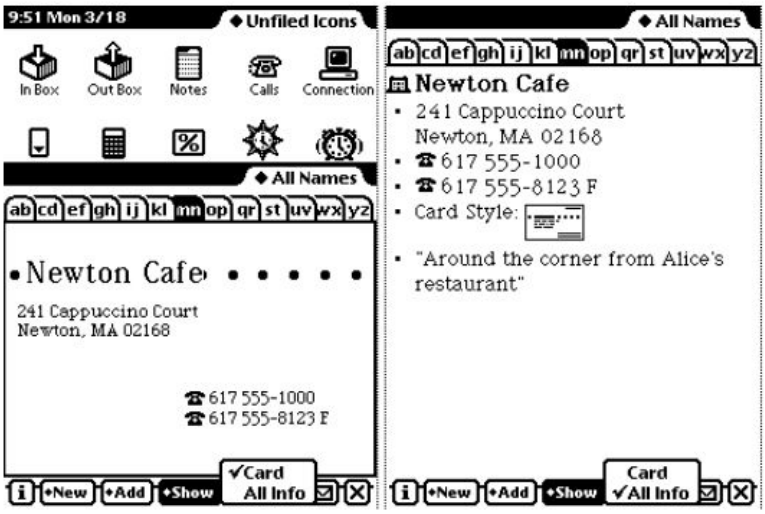
	a list of all available information. These two views are shown in Figure 19-1." Newton Guide, p. 19-2. "The Names application stores its data in the ROM_CardFileSoupName ("Names") soup. Entries in this soup are frames for either a person, an owner, a group, a company, or a worksite card." Newton Guide, p. 19-7. **Figure 19-1** Names application Card and All Info views  Newton Guide, p. 19-3. See also Newton Guidelines at pp. 2-36, 3-27, 8-13, 8-31. See also Newton Guidelines at fig. 3-30. See also Newton Guidelines at p. 8-23: "The Assistant is pre-programmed to know certain verbs that describe actions in the built-in applications." See also Newton Guidelines at p. 8-25: "An Assist slip's Please picker lists the actions that applications have currently registered with it . . . ." See also Newton Guidelines at fig. 8-24. "So if a user is operating the Newton MessagePad 2000 – A. Uh-huh. Q. – and is – regardless of what application the user is using –

Exhibit I

	A. Yes. Q. And taps the Intelligent Assist icon, what happens next? What is the user presented with? A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user. For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh. Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * *

Exhibit I

	A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work. Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number? A. Yes." Pagallo Depo at 92:6-20. "And then is the user able to then save information to the Name File after making the call? A. Um, yes. * * * Yes. Sorry, my – I got sidetracked, but yes. So the same UI that we, um, happen before when we try the Jennifer Lopez case, happens in this case, after I entered the number, because, um, I didn't have a phone number So I had to enter the phone number. And once the phone number and I place the call, the UI to add it to the Names File is presented to the user. Q. Okay. And then so the user is presented with an option to add this new contact to the new file? Okay. A. Yes." Pagallo Depo at 92:21-93:11. "And then how does the user then add the – this new contact to the Name File? Is there an icon or button that's add or – A. There is a UI that has the name card with the name, the phone number, and it says, 'Would you like to add a new name with the information,' and then you can say – select yes or no. And if I select yes,

Exhibit I

	then that new card is added to the Name File." Pagallo Depo at 93:12-19. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up. Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12. "The – the system – the Intelligent Assistant was a framework that could be invoked through the application, but it was a – as a system service." Pagallo Depo at 100:16-18. For example, as observed during my Inspection of the Newton Device, the text "Call Michael" is entered in the Notes application document. As can be seen in the screenshots, I tapped the "Assist" button on the screen with a stylus, which launched the Intelligent Assistant. Newton then searched the Names application contact cards (or, more specifically, the "Names soup," which contains the data for each contact card) for information associated with the name "Michael." A "Phone" window then appeared to include the last name ("Jordan"), which is associated with the first name ("Michael"), as well as the phone number ("1 800 1112222"), which is also associated with the first name ("Michael"). See also screenshots from the "while the document is being displayed" element above.

Exhibit I

	Prior to the aforementioned sequence of events observed during my Inspection, a new contact card was added to include the first name "Michael," last name "Jordan," and phone number "800 1112222" to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card). The last name ("Jordan") and phone number ("1 800 1112222") was associated with the first name ("Michael") in this contact card. Indeed, prior to adding this contact card to the Names application (or, more specifically, the "Names soup," which contains the data for each contact card), the same aforementioned sequence of events (i.e. "Call Michael") did not result in the display of any other information associated with the name "Michael." See also screenshots from the "while the document is being displayed" element above. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	Newton Guide discloses claim 1. *See* claim 1 above. Newton further discloses this element. "Correcting Misrecognized Text If the Newton system does not recognize a word correctly, the user can correct it by several means. For one, the user can replace any letter by writing another letter over it. The user also has the option of selecting or erasing the word and writing it again. Another alternative: the user can double-tap a word to pop up a picker that lists some alternate words. From the list of alternates the user can select one as a replacement. Figure 6-25 shows how a Correction picker works. * * * Tapping the Corrector button brings up a Corrector view, in which the user can make corrections to individual letters. The user can write over a letter to replace it, delete a letter by scrubbing it, or insert a space in which to write an additional letter. In addition, the user can tap a letter in the Corrector view to pop up a Correction picker that lists alternate letters plus the commands Insert and Delete." Newton Guidelines at 6-29-6-30. See also Newton Guidelines at fig. 6-25, 6-26.

Exhibit I

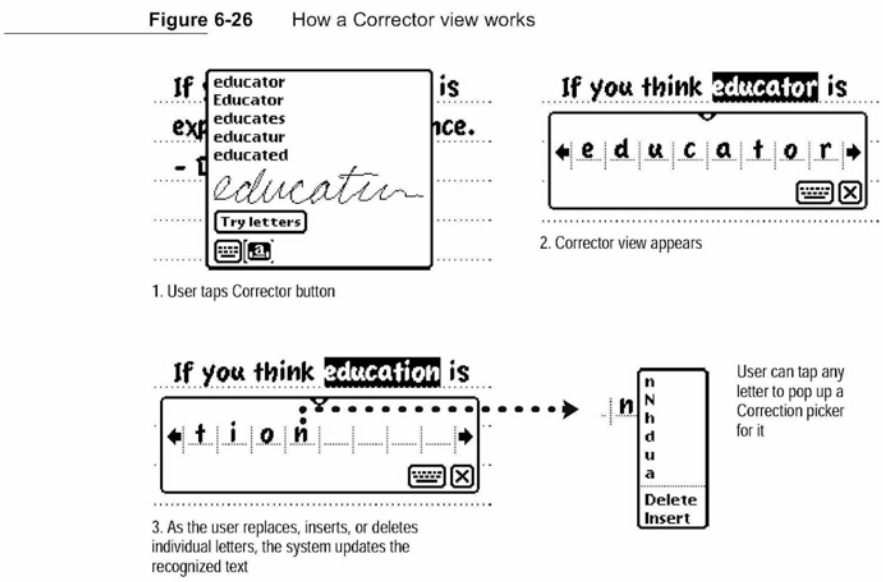
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining Newton and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have Newton cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	Newton Guide discloses claim 1. *See* claim 1 above. Newton further discloses this element. "Correcting Misrecognized Text If the Newton system does not recognize a word correctly, the user can correct it by several means. For one, the user can replace any letter by writing another letter over it. The user also has the option of selecting or erasing the word and writing it again. Another alternative: the user can double-tap a word to pop up a picker that lists some alternate words. From the list of alternates the user can select one as a replacement. Figure 6-25 shows how a Correction picker works. * * * Tapping the Corrector button brings up a Corrector view, in which the user can make corrections to individual letters. The user can write over a letter to replace it, delete a letter by scrubbing it, or insert a space in

Exhibit I

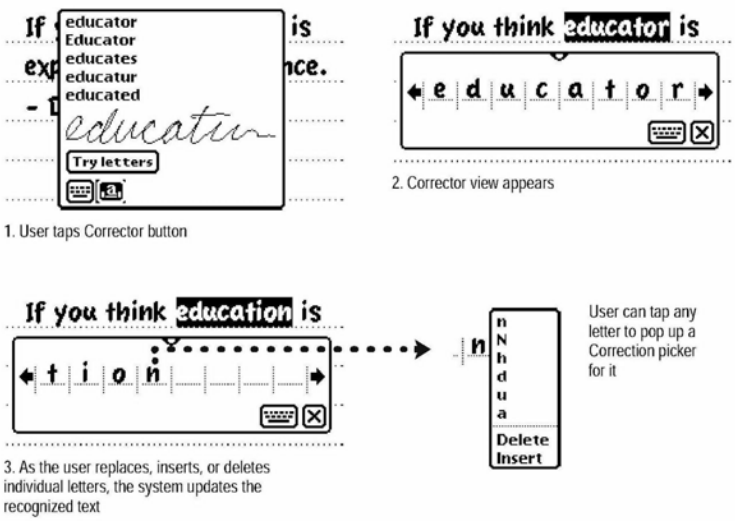
	which to write an additional letter. In addition, the user can tap a letter in the Corrector view to pop up a Correction picker that lists alternate letters plus the commands Insert and Delete." Newton Guidelines at 6-29-6-30." Newton Guidelines at p. 6-29—6-30. See also Newton Guidelines at fig. 6-25, 6-26. Figure 6-26 How a Corrector view works  For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 3 (e.g., Schulman, Domini, and Schabes). See also '843 patent at 1:17-42; My Report at paragraphs 187-189. A POSITA would have reasonable expectation of success to achieve predictable results in combining Newton and the prior art references and systems in Exhibit U, Table 3. For example, a POSITA would have recognized advantages and benefits to have Newton cause insertion of at least part of the second information into the document. *See also* My Report at paragraphs 187-189.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	Newton discloses this element. See claim 1. For example, the Newton implements a method for finding data related to the contents of a document created by the Notes application running on the device. *See* claim 1. This method was implemented in program instructions stored on a non-transitory computer readable medium.

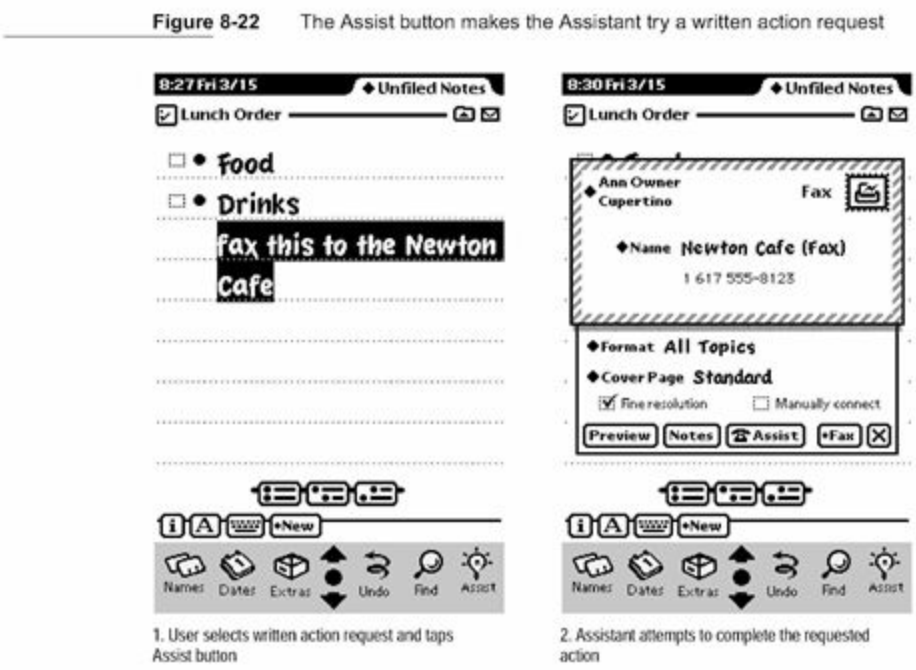
Exhibit I

	For example, Newton Datasheet states: "Newton Hardware Architecture ● StrongARM SA-110 RISC processor at 162 MHz ● Newton custom system chip set–ASIC ● 5MB RAM (1MB of DRAM, 4MB of Flash RAM) ● 8MB of mask ROM" Newton Datasheet, p. 2. "Included software ● Includes business software programs: word processor, Internet e-mail client, World Wide Web browser, and spreadsheet (in some configurations) ● Allows freeform notes, outlines, lists, and phone logs via digital ink or handwriting recognition ● Helps you plan activities using the calendar, and manage priorities using the To Do List ● Includes name and address files ● Offers productivity tools such as time-zone maps, a calculator, and currency exchange" Newton Datasheet, p. 1. Newton Manual states: "A hard reset erases all data and information in the MessagePad's internal memory. This includes any applications that were already on your MessagePad when you bought it (including the Newton Tour and HW Instructor). It also erases information you have stored, such as name cards, notes, calls, and pages in the Date Book. This procedure does not remove the MessagePad's built-in programs, operating system software, or system updates." Newton Manual, p. 230. Newton Guide states: "Using System Software Most of the routines and application components that comprise the Newton system software reside in ROM, provided in special chips contained in every Newton device. When you application calls a system routine, the operating system executes the appropriate code contained in ROM."

Exhibit I

	Newton Guide, p. 1-17. "Memory It is helpful to understand the user of random access memory (RAM) in the system, since this resource is shared by the operating system and all applications. Newton RAM is divided into separate domains, or sections, that have controlled access. Each domain has its own heap and stack…" See also Newton Guidelines at p. 1-3: What People Do With Newton The features and capabilities that make Newton what it is also strongly influence what people want to do with Newton devices. These expectations indirectly affect the user interface of Newton software. An application must make it easy for people to accomplish the following tasks on demand: ■ Capture information fragments--write, sketch, pick from lists, specify dates and times, and select options ■ Organize information--file, sort, schedule, prioritize, copy, delete, and format ■ Retrieve information--find, recall, browse, skim, read, and view ■ Send and in some cases receive information by various means--print, fax, mail, and direct transfer" See also Newton Guidelines at p. 1-12: "Also keep in mind when designing your application that future Newton devices may have larger or smaller screens than current Newton devices. To work with different screen sizes, a Newton application must check the screen size and make adjustments as needed to the size and location of the things it displays so that everything fits. If you want your application to work in either of the two display orientations available on an Apple MessagePad 120, your application needs to be able to adjust the position and configuration of everything it displays for regular or sideways orientation of the display. Figure 1-3 shows how the build-in Notepad application and the on-screen keyboard adjust their size, position, and layout when a user rotates the display." See also Newton Guidelines at p. 6-8. See also Newton Guidelines at figs. 1-3. 8-22.

Exhibit I



"Q. What is the Intelligent Assist feature?
A. Um, the Intelligent Assist feature is in the Newton, is a system service that was designed to facilitate interactions and user actions.

Like, for instance, placing a call, sending an email, creating a reminder, so that the users could do that from, say, the Notes application without necessarily having to go into the calendar or the date, the Names application.
Q. Or even going –
A. The call.
Q. – into the Call application.
A. For instance, yes." Pagallo Depo. at 68:5-17.

"So if a user is operating the Newton MessagePad 2000 –
A. Uh-huh.
Q. – and is – regardless of what application the user is using –
A. Yes.
Q. And taps the Intelligent Assist icon, what happens next?
What is the user presented with?
A. Um, if the text is, um, interpreted as a action that Intelligent Assist can assist with, the UI relevant to that action is presented to user.

For instance, one of the canonical demos that we had was fax and a phone number. And if you tap Intelligent Assist, then the fax user interface will show with a phone number already populated in the right field, so the user would just tap on the right UI to initiate the fax." Pagallo Depo. at 72:8-25.

Exhibit I

	"Q. So if a user is within the Notepad application and the user has entered in text that says fax and then a phone number – A. Uh-huh. Q. – how can the user then invoke Intelligent Assist after entering in that text? What are the steps that need to be taken? A. Once the text is entered and is recognized, then the user can tap Assist." Pagallo Depo. at 73:19-74:2. "What feature is recognizing the text on the – in the Notepad application for the Intelligent Assist feature to work? A. The handwriting recognition feature that we talked about earlier today." Pagallo Depo. at 74:10-14. "Q. And then after the text is entered in, then the user presses the Intelligent Assist button, the Intelligent Assist feature is what is recognizing or the event – A. What is understanding the command. Q. Okay. So within Intelligent Assist, are there specific commands – only specific commands or actions that the Intelligent Assist feature can recognize? A. Yes." Pagallo Depo. at 75:1-9. "And so after the user enters in the text, fax, and a phone number, and then taps the Intelligent Assist button within the Notepad application, we had discussed the Intelligent Assist feature recognizing the command word, which would, in that example be fax. A. Correct. Q. And then what does the Intelligent Assist feature to before it's able to perform the command, such as in this instance, faxing a particular phone number? So after it recognizes the command, fax, how does it recognize the command – the – not the command, but the phone number that's entered in by the user? A. The – in the – as assistant service, there is a – there was a software that would recognize structure text, like phone numbers and dates and times and tag a particular sequence of characters as a token of that type. And so then, um, upon being – um, invoking the Assistant in this – let's continue with the sample of fax, the UI related to faxing, would come up and the field for phone number would be populated with a phone number that the user wrote." Pagallo Depo. at 78:19-79:15. "So if a user taps the word Please, then it – um, is the user presented with different options for actions that could occur?

Exhibit I

	A. The different option for action that could occur with a person's name." Pagallo Depo. at 85:5-9. "So underneath the action word, call, fax, find, mail, print, remember, schedule in time, the other items that appear under the Please Picker are the recent action – A. I believe so." Pagallo Depo at 85:21-25. "So if the user chooses one of the either recent items that occurred in their history or one of the action items and selects that – A. Uh-huh. Q. What would then happen next in order to execute the command? What would a user need to do? A. Once they – the user is presented with a Please Picker, they use – the user may – this – if it chooses one of the commands in the top part of the picker, the UI for that command will show up with Bob in this case, in the right field. Q. Okay. And when using the Notepad application, after a user enters in text and before the user selects the – well, before the user selects the Intelligent Assist agent – or, sorry, Intelligent Assist feature, um, does the user need to do anything to the text, before – such as highlight or underline before the Intelligent Assist feature can be launched? * * * A. No. These – there's no requirement for the text to be select – to be selected to – for the Assist to be invoked and work. Q. Okay. So the user does not need to select certain text before invoking the Intelligent Assistant? A. No." Pagallo Depo. at 86:1-87:4 "But if you do the selection, you – the Intelligent Assistant will act on the selected text only." Pagallo Depo. at 87:15-17. "So it recognized the command words, I think, that we had previously discussed, which were call, fax or find, mail, print, remember schedule. A. Yes. Q. And then it also recognizes synonyms of those words? A. Yes. Q. And then if there is a name of an individual that's in the text that's entered into your Notepad application, and the name of the individual follows the command word such as call, fax, find, mail, print, remember or schedule, does the Intelligent Assist feature look to the Name File application to see if there is a contact with that name that already exists? A. Yes." Pagallo Depo. at 88:10-24. "And if a contact with that name does not already exist, what would the

Exhibit I

	user be presented with or what options? So if you have – for instance, if the user enters in Call Jennifer Lopez – A. Uh-huh. Q. – which was not a contact that is present in the example device that we're using today. Um, so if the user enters Call Jennifer Lopez and then presses the Intelligent Assist feature, what would the user be presented with? * * * Um, so the example that we walked through, I had entered Jennifer Lopez in the Name field, so there was a recognition that Jennifer Lopez is a name. That's why when I was presented with a card with Jennifer Lopez was the name." Pagallo Depo. at 88:25-89:16. "Um, so I type, Call Julio, which is not in the Names File. And after that, I type Assist, and then the app Call UI comes up and the Name field is populated with string Julio." Pagallo Depo. at 91:23-92:1. "And then after the Name field is populated, does it give the user – what actions are presented to the user at in that point in time? A. So the user can start the call as previously." Pagallo Depo at 92:2-5. "But if the – Julio is not in the Name File application, so there's not any contact information – A. Right. Q. – for the individual, Julio, at this point in time. So if the user just enters in the phrase, Call Julio, and then invokes Intelligent Assist and then the Name Field pops up with Julio, how does the user then call Julio? A. Um, there's a – I just did that. There is a UI that comes up that gives you – give me the opportunity to enter the phone number. Q. Okay. So then the user has to enter the phone number? A. Yes." Pagallo Depo at 92:6-20. "And then is the user able to then save information to the Name File after making the call? A. Um, yes. * * * Yes. Sorry, my – I got sidetracked, but yes. So the same UI that we, um, happen before when we try the Jennifer Lopez case, happens in this case, after I entered the number, because, um, I didn't have a phone number So I had to enter the phone number. And once the phone number and I place the call, the UI to add it to the Names File is presented to the user. Q. Okay. And then so the user is presented with an option to add this

Exhibit I

	new contact to the new file? Okay. A. Yes." Pagallo Depo at 92:21-93:11. "And then how does the user then add the – this new contact to the Name File? Is there an icon or button that's add or – A. There is a UI that has the name card with the name, the phone number, and it says, 'Would you like to add a new name with the information,' and then you can say – select yes or no. And if I select yes, then that new card is added to the Name File." Pagallo Depo at 93:12-19. "So if a user selects the phrase or the command Call and then writes the name Bob on this template, what does the user need to do next, to have the Intelligent Assist feature execute the command or – A. Tap. Q. – the phrase Call back? A. Tap on the button Do. Q. Okay. There's a button called Do? A. Uh-huh. Q. Okay. And then after the user presses the button Do, what happens next? A. The call UI shows up with the name of Bob and the phone number and it shows that it's gonna be using the speaker and then has an Assist call and a Close Slip button. So, essentially, now I'm in the position as a user to tap on the button Call to – to complete the call." Pagallo Depo at 96:12-97:4. "So the screen that shows for the user that you were just describing – A. Uh-huh. Q. – where you can see the specifications that are being used for the phone call, does the user have the opportunity to change any of the information that's presented in the screen, so if the phone number – if the user realizes that this phone number for Bob is incorrect and wants to change the phone number that's being used, does the user have the opportunity to do that? A. Yes. The user has the opportunity to tap a name and select a different name or select a different entry." Q. Okay. And that's presented to the user as options before a call – A. Correct. Q. – is executed?" Pagallo Depo at 97:5-22. "And then if, in the instance that the information that's presented to the user for Bob is correct, and the user decides to place the phone call, what does the user need to do next after it's presented with the – the screen? A. Tap on the button that says Call." Pagallo Depo at 97:23-98:3.

Exhibit I

	"So I just entered Call Bob as one of my items in the Date application. I tap Assist, and the same UI that allows me to either change date, entry of the name for the call or place a call show up. Q. Okay. So the same Intelligent Assist feature that was available in the Notepad application, is also available in the Datebook application as well? A. Yes." Pagallo Depo at 99:5-12. "The – the system – the Intelligent Assistant was a framework that could be invoked through the application, but it was a – as a system service." Pagallo Depo at 100:16-18. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Newton discloses this element. *See* claim 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Newton discloses this element. *See* claim 1.
retrieving the first information;	Newton discloses this element. *See* claim 1.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types	Newton discloses this element. *See* claim 1.

Exhibit I

of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Newton discloses this element. *See* claim 1.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Newton discloses this element. *See* claim 1.
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising:	Newton discloses claim 23. *See* claim 23.
providing a prompt for updating the information source to include the first information.	Newton discloses this element. *See* claim 8.

Exhibit J

Claim Chart Applying U.S. Patent No. 5,644,735 Against the '843 Patent

U.S. Patent No. 5,644,735 to Luciw et al. ("Luciw '735") was filed on April 19, 1995 and issued on July 1, 1997. It therefore constitutes prior art under pre-AIA 35 U.S.C. § 102(a), (b) and (e). As shown below, Luciw anticipates claims 1, 8, 13, 15, 17, 18, 19, 23, and 30 of the '843 patent.

"Obviousness Statement" - To the extent that the Judge or Jury finds that Luciw '735 does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, Luciw '735 discloses the preamble. "The present invention relates generally to computer systems, and more particularly to computer-implemented assistance methods and apparatus." 1:20-22. "In operation, information is input into the pen-based computer system 10 by 'writing' on the screen of display assembly 20 with the stylus 38." 5:29-31. "The screen illustrated in FIG. 2 is referred to as the 'notepad', and is preferably an application program running under the operating system of the pen based computer system 10. In this preferred embodiment, the notepad is a special or 'base' application which is always available beneath higher level applications."

Exhibit J

	6:49-54. "Example of object types used in the following description include paragraph, line, and word objects." 7:19-20. "FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. "FIG. 8a illustrates details of the operation of step 123 of FIG. 3 dealing with the updating of information and linked information in smart fields. In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation. In the absence of there being any linked fields available, the data obtaining step of 204 is skipped and the data is entered manually, if available. Otherwise, the phone data field will remain vacant as to that particular data element. Operation of updating information and linked information in accordance with step 123 of FIG. 3 is completed with step 208 in FIG. 8a." 12:41-54. *See* Figs. 2, 3, 6b, & 8a and accompanying text:

Exhibit J

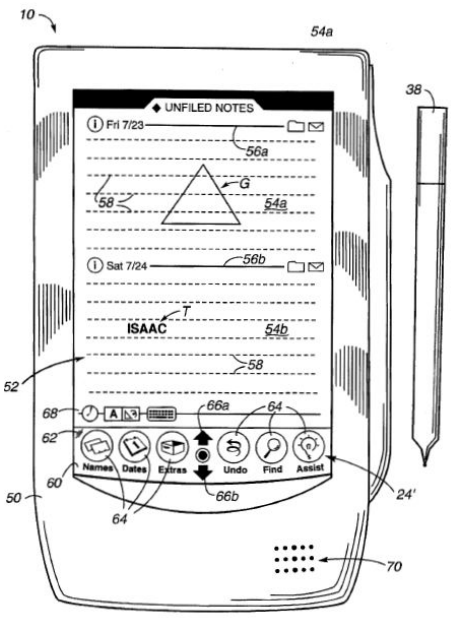


Figure 2

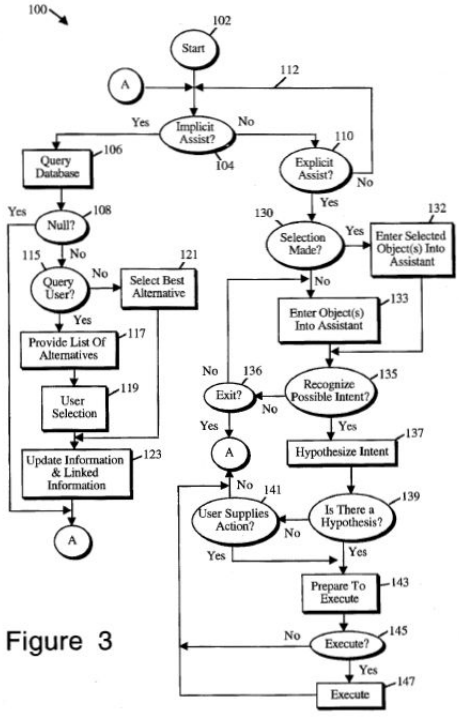


Figure 3

Exhibit J

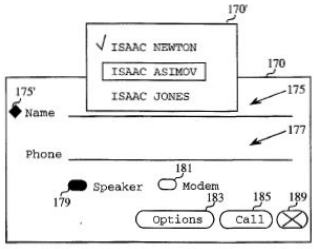
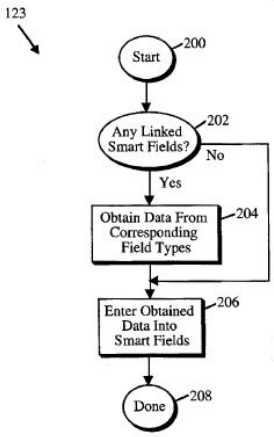
	 *Figure 6b*  Figure 8a For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Luciw '735 discloses this element. "In operation, information is input into the pen-based computer system 10 by 'writing' on the screen of display assembly 20 with the stylus 38." 5:29-31. "The screen illustrated in FIG. 2 is referred to as the 'notepad', and is preferably an application program running under the operating system of the pen based computer system 10. In this preferred embodiment, the notepad is a special or 'base' application which is always available beneath higher level applications. The notepad application, like other applications, runs within a window, which in this instance comprises the entire viewing screen 52. Therefore, as used herein, a 'window' is the entire screen or any portion of an entire screen which is dedicated to a particular application program." 6:49-59.

Exhibit J

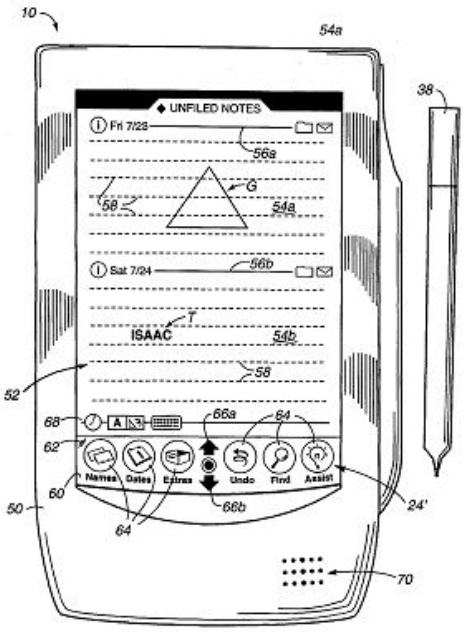
	See Fig. 2 and accompanying text:  *Figure 2* "Additional note areas, such as a note area 54b, can be formed by the user by drawing a substantially horizontal line across the screen 52 with the stylus 38. The substantially horizontal line is recognized by the system 10 and is converted into a second header bar 56b. Additional text, graphical, and other data can then be entered into this second note area 54b. For example, the text object T comprising 'ISAAC' has been entered into second note area 54b." 6:24-31. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Luciw '735 discloses this element. "Examples of object types used in the following description include paragraph, line and word objects." 7:18-20. "For example, the text object T comprising 'ISAAC' has been entered into second note area 54b."

Exhibit J

	6:29-30. "If a user selection has been made, particular selected objects are entered into the assistance operation in step 132 [of FIG. 3]. If no user selection has been made, objects entered since a delimiter are entered into the assistant in a step 133." 9:27-30. "The process of checking for the selection of a particular object begins at 220 in FIG. 9a and is conducted at 222." 13:10-12. "FIGS. 9b-9c indicate graphically the performance of the selection query operation as expressed in FIG. 9a. In FIG. 9b the objects CALL ISAAC are indicated." 13:27-29. "FIG. 9d shows the highlighted objects transferred to window 170, to perform entry of the selected object(s) into the assistant. Such entry into the assistant function need not be accompanied by actual transferal into window 170 and may be transparently performed without direct user awareness." 13:31-36. "The recognition of possible user intent process called for at 135 in FIG. 3 and expressed in example form at FIG. 11a, calls for a matching operation between particular noted object(s) such as those illustrated in FIG. 11b and those expressed in the template of FIG. 11c." 14:18-22. "In many instances, intent will not so clearly be evident. For example, them [sic] may be multiple function matches based upon a particular combination of objects. Accordingly, it will be imperative to hypothesize the actual user intent based upon a selection of alternatives, as suggested at 137 in FIG. 3." 14:29-34. "FIG. 11c illustrates selected example functions such as scheduling, finding, filing, formatting, mailing, faxing, printing, and calling, just to

Exhibit J

	cite a few of the possibilities." 14:2-4. "FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17. "A smart field is considered to be a predefined region on screen 52 of computer system 10 shown in FIG. 2, or a predefined region within a window which appears on screen 52, as suggested below with reference to FIG. 46b and which will be discussed in greater detail in the text description below associated with that Figure." 8:15-18. "However, implicit assist may be indicated not just by entry of an indication in a smart field, but by the happening of any of a number of predefined allowable events which lead to a query of the database at process step 106. A user entry made into a smart field is not the only way computer system 10 is caused to undertake an implicit assist operation. Certain kinds of events on screen 52, for example, such as the writing of a particular indication or word on screen 52 outside of a particular smart field may trigger an implicit assist. In general, implicit assist can be triggered by the happening of any of a number of predefined allowable events." 8:30-41. "If the entry in the smart field has been made by the user, the assistance process takes action to identify or recognize the kind of implicit assistance indicated at a step 154. After recognition has been accomplished, operation continues as suggested in FIG. 3 at step 106

Exhibit J

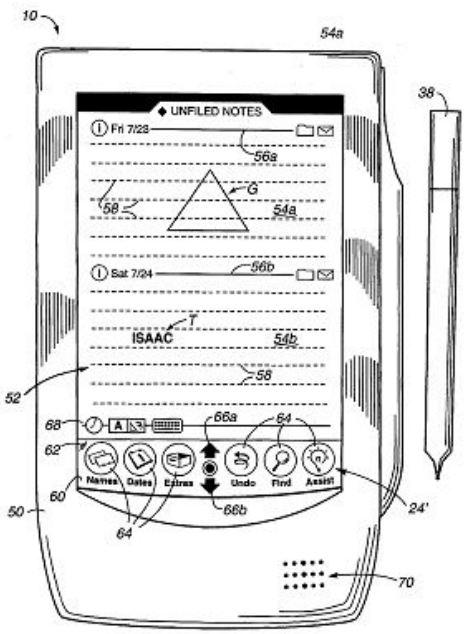
	with a query of the database." 10:15-20. "At step 104, the process recognizes whether or not an implicit assistance function is to be provided by computer system 10. As will be seen, implicit assistance may, for example, arise from an entry into a smart field by a user. If a user does enter information into a "smart field," the computer database will be queried at step 106 to determine whether assistance is possible given the user input." 8:7-13. *See also, e.g.*, Figs. 2, 3, 4a, 9a-c, & 11a-c and accompanying text:  *Figure 2*

Exhibit J

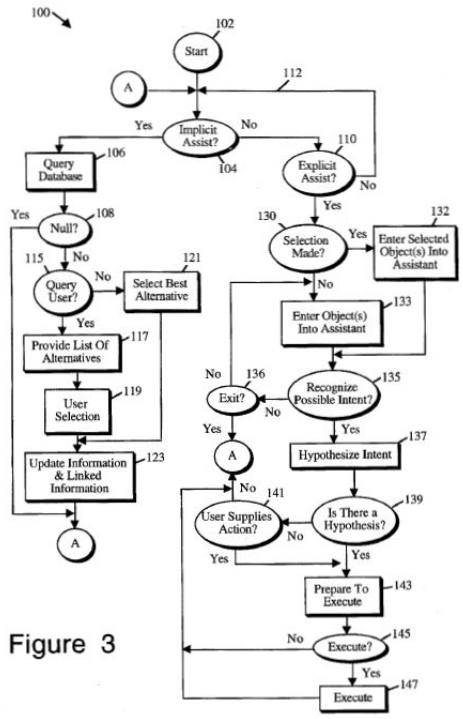


Figure 3

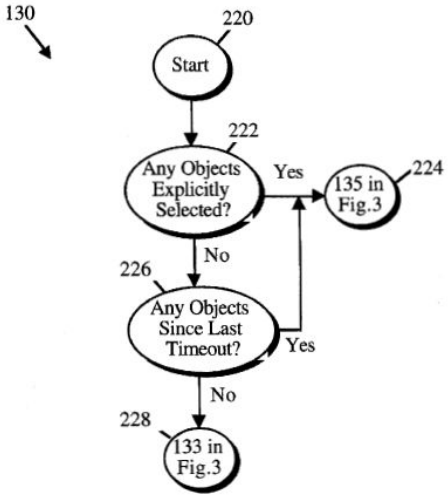


Figure 9a

CALL ISAAC

Figure 9b

CALL ISAAC

Figure 9c

Exhibit J

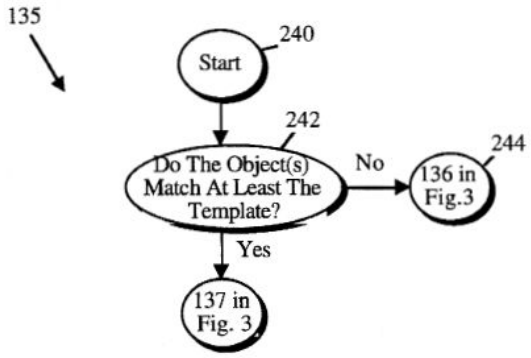
<table>
<tr>
<td></td>
<td>


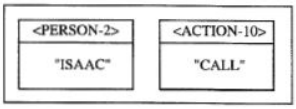
Figure 11a


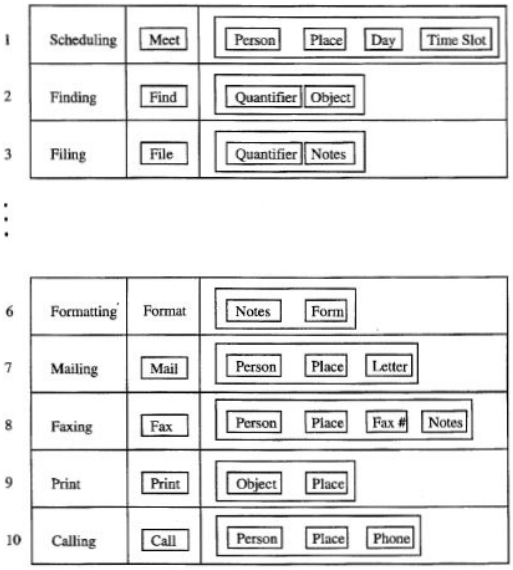
Figure 11b


Figure 11c

For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.</td>
</tr>
<tr>
<td>retrieving the first information;</td>
<td>Luciw ’735 discloses this element.</td>
</tr>
</table>

Exhibit J

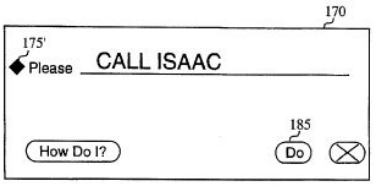
	"An example of how the delimiter process can be accomplished, for example, involves the entry of only those objects on the screen 52 which are delimited in some fashion from the other objects which may have been entered on the screen. For example, if several paragraphs have been entered on the screen, only the last paragraph's objects will be considered for entry as objects into the assistant. Time may also be used as a delimiter. For example, if a considerable period of time separates a given object on the screen from another, only the most recent object will be entered into the assistant." 9:34-44. "If a user selection has been made, particular selected objects are entered into the assistance operation in step 132. If no user selection has been made, objects entered since a delimiter are entered into the assistant in a step 133." 9:27-30. "FIG. 9d shows the highlighted objects transferred to window 170, to perform entry of the selected object(s) into the assistant." 13:31-33. "means for providing a smart field responsive to information of a predefined type" Claim 1(c). *See* Fig. 9d and accompanying text:  *Figure 9d*
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the	Luciw '735 discloses this element. "The keypad 24 can comprise an array of mechanical buttons or switches coupled to I/O circuitry 18 by a data bus 42. Alternatively, keypad 24 can comprise an entire, standard QWERTY keyboard. In the present embodiment, a separate keypad 24 is not used in favor of a 'pseudo' keypad 24'. This 'pseudo' keypad 24' comprises 'button' areas which are associated with a bottom edge of the tablet membrane that extends

Exhibit J

second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	beyond the lower edge of the LCD display. These button areas are defined by a printed or silk-screened icons which can be seen through the transparent membrane of the input tablet. When the 'buttons' are selected by engaging the stylus 38 with the membrane over these printed icons, the membrane senses the pressure and communicates that fact to the CPU 12 via data bus 38 and I/O 18. An example of pseudo keypad 24' is shown in FIG. 2." 5:7-21. "An example of an indication of user desire to have explicit assistance undertaken is the act of using pen 38 in FIG. 2 to tap or click on the assist icon or button 64." 8:51-53. "Both the action 'call' and the person to be called are present in the template, permitting an effective, though not complete match. The place and the phone number are yet to be determined." 14:25-28. "The process calls for example for the filling in of a plan template and the identification of any missing preconditions, as set forth at step 292 of FIG. 13. Next, a step 293 resolves missing preconditions to the extent possible." 15:9-13. "In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation." 12:43-54. "FIG. 11a shows the recognition of object(s) process which is part of FIG. 3 at 135, in order to enable recognition of possible user intent. The recognition process is started at step 240 in FIG. 11a. Next, a decision

Exhibit J

	step 242 determines whether the object(s) match at least one template. If not, the process continues at a step 244 which corresponds to step 136 of FIG. 3. If so, the process continues at step 137 of FIG. 3. In substance, the process aims to determine whether the object(s) match at least one of the templates of object combinations set forth in FIG. 11c. FIG. 11b illustrates the object combination under operation, denoted by kind of object. The verb CALL is considered to be an action object and ISAAC is considered to be a person object. The two objects in combination are subject to template comparison. The template in FIG. 11c is effective for organizing in preset form the various object combinations which are capable of further operation as particular functions to be accomplished. FIG. 11c illustrates selected example functions such as scheduling, finding, filing, formatting, mailing, faxing, printing, and calling, just to cite a few of the possibilities." 13:52-14:4. "FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17. "The recognition of possible user intent process called for at 135 in FIG. 3 and expressed in example form at FIG. 11a, calls for a matching operation between particular noted object(s) such as those illustrated in FIG. 11b and those expressed in the template of FIG. 11c." 14:18-22. "Details of one way to carry out the database query process indicated in FIG. 3 at step 106 can be understood in connection with FIG. 5. In particular, FIG. 5 illustrates a frame 180 which is a special case of a frame, referred to commonly as a "type" frame, as the frame refers to a particular type, i.e., the type <PERSON>." 10:49-54.

Exhibit J

	"For example, if it was desired to retrieve all of the frames that were colored red, a typical frame accessor language query would be in the form of: (QUERY (MEMBER-VALUE COLOR ?XRED) and would return a list of frames that have a COLOR slot whose value is red." 11:33-37. "In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation." 12:43-54. "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled 'options' at 183 can be produced as a help menu. Further, a "call" activity can be undertaken by selecting a ''call' button 185 indicated on the face of window 170." 12:61-13:3. "However, implicit assist may be indicated not just by entry of an indication in a smart field, but by the happening of any of a number of predefined allowable events which lead to a query of the database at process step 106. A user entry made into a smart field is not the only way computer system 10 is caused to undertake an implicit assist operation. Certain kinds of events on screen 52, for example, such as the writing of a particular indication or word on screen 52 outside of a particular smart field may trigger an implicit assist. In general, implicit assist can be triggered by the happening of any of a number of predefined allowable events."

Exhibit J

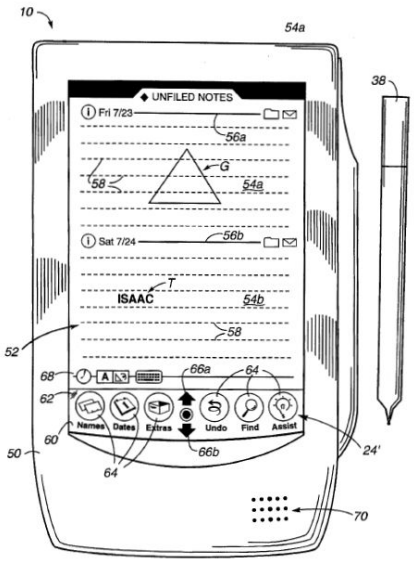
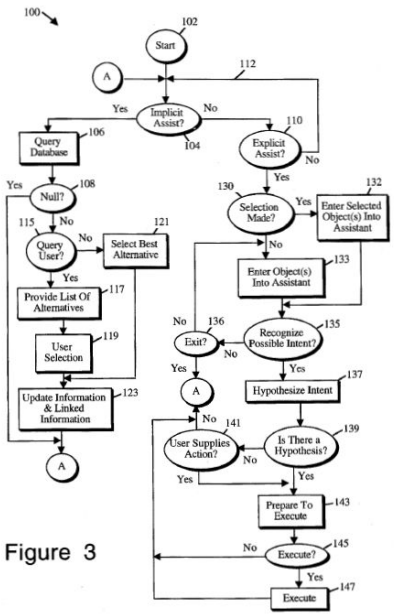
	8:30-41. "FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. "Next, an attempt is made at step 135 to recognize the possible intent expressed by the objects entered into the assistance process." 9:46-48. *See* Figs. 2, 3, 5, 6a-b, 8a-b, 11a-c, & 12c and accompanying text:  *Figure 2*  Figure 3

Exhibit J

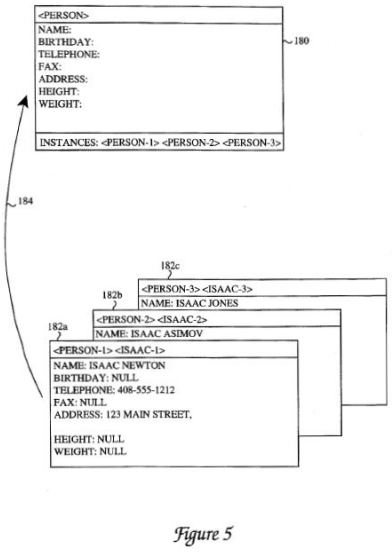


Figure 5

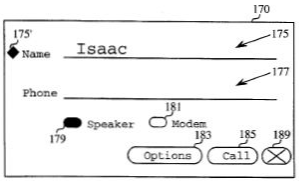


Figure 6a

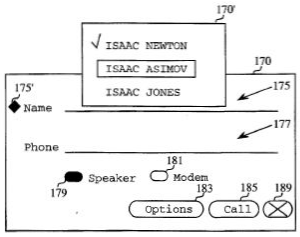


Figure 6b

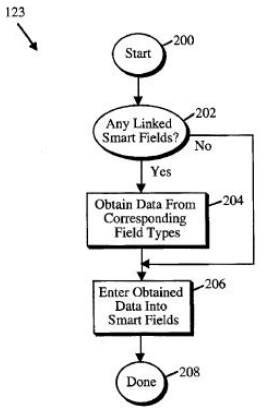


Figure 8a

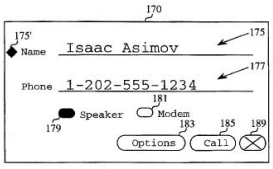


Figure 8b

Exhibit J

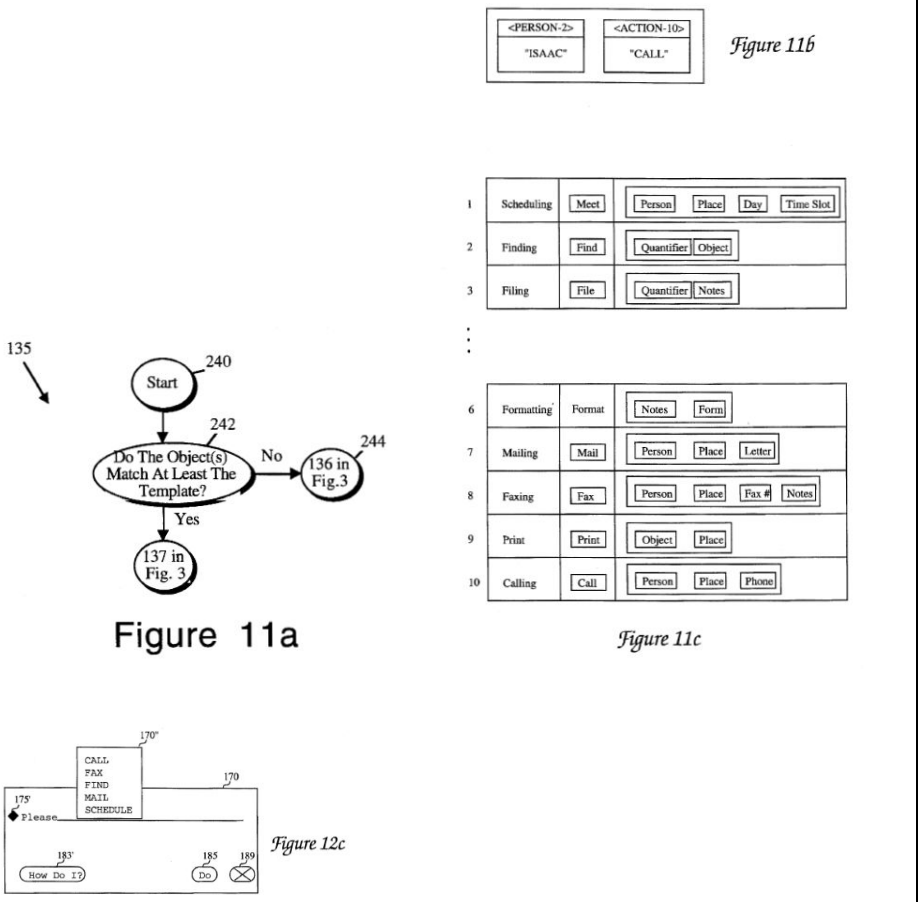
	 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Luciw '735 discloses this element. "An example of an indication of user desire to have explicit assistance undertaken is the act of using pen 38 in FIG. 2 to tap or click on the assist icon or button 64." 8:51-53. "The screen illustrated in FIG. 2 is referred to as the 'notepad', and is preferably an application program running under the operating system of the pen based computer system 10. In this preferred embodiment, the notepad is a special or 'base' application which is always available beneath higher level applications."

Exhibit J

	6:49-54. "Both the action 'call' and the person to be called are present in the template, permitting an effective, though not complete match. The place and the phone number are yet to be determined." 14:25-28. "The process calls for example for the filling in of a plan template and the identification of any missing preconditions, as set forth at step 292 of FIG. 13. Next, a step 293 resolves missing preconditions to the extent possible." 15:9-13. "In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation." 12:43-54. "Details of one way to carry out the database query process indicated in FIG. 3 at step 106 can be understood in connection with FIG. 5. In particular, FIG. 5 illustrates a frame 180 which is a special case of a frame, referred to commonly as a "type" frame, as the frame refers to a particular type, i.e., the type <PERSON>." 10:49-54. "For example, if it was desired to retrieve all of the frames that were colored red, a typical frame accessor language query would be in the form of: (QUERY (MEMBER-VALUE COLOR ?XRED) and would return a list of frames that have a COLOR slot whose value is red." 11:33-37.

Exhibit J

	"Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3." 11:60-12:6 *See* Figs. 2, 3, 5, 6c & 13 and accompanying text:

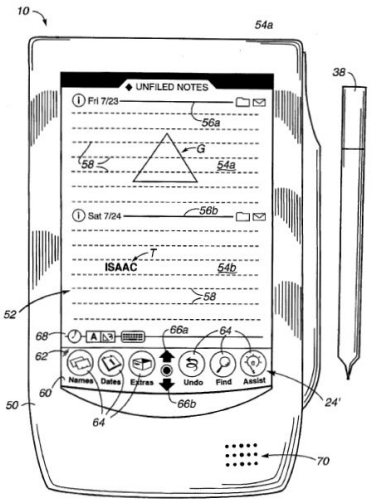


Figure 2

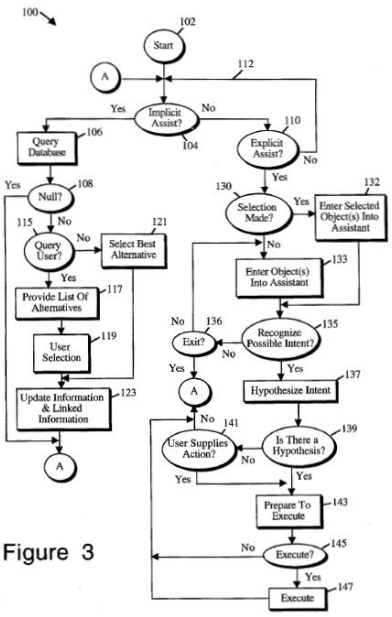


Figure 3

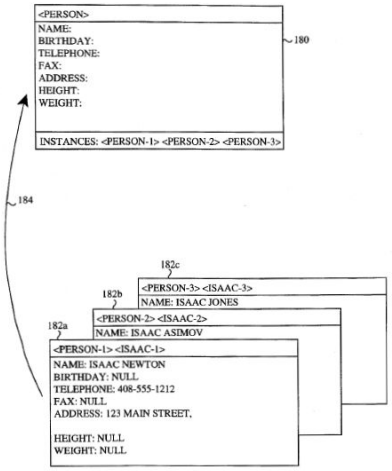


Figure 5

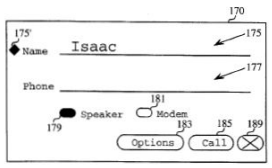


Figure 6a

Exhibit J

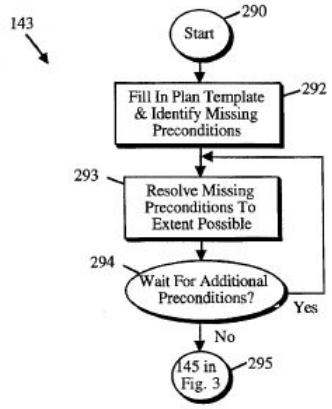
	 Figure 13 For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Luciw '735 discloses this element. "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled 'options' at 183 can be produced as a help menu. Further, a 'call' activity can be undertaken by selecting a "call" button 185 indicated on the face of window 170." 12:61-13:3. "FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17.

Exhibit J

	"FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. "FIG. 8a illustrates details of the operation of step 123 of FIG. 3 dealing with the updating of information and linked information in smart fields. In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation. In the absence of there being any linked fields available, the data obtaining step of 204 is skipped and the data is entered manually, if available. Otherwise, the phone data field will remain vacant as to that particular data element. Operation of updating information and linked information in accordance with step 123 of FIG. 3 is completed with step 208 in FIG. 8a." 12:41-54. *See* Figs. 3, 6a-c, 8a-b, & 11c and accompanying text:

Exhibit J

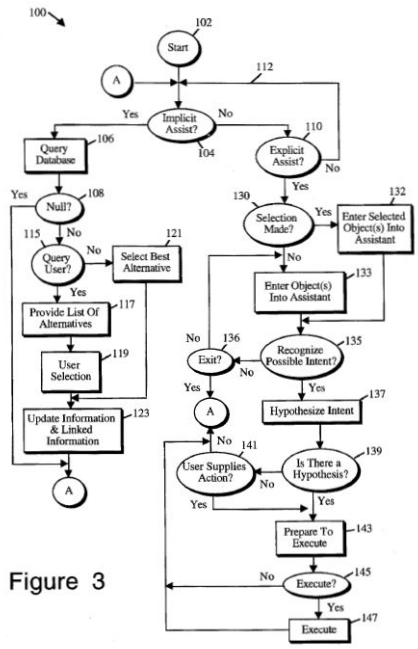


Figure 3

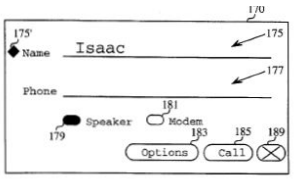


Figure 6a

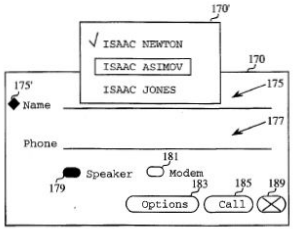


Figure 6b

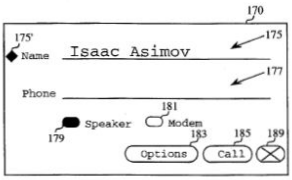


Figure 6c

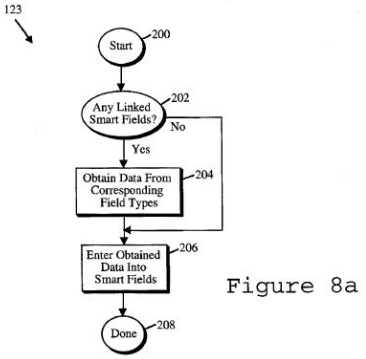


Figure 8a

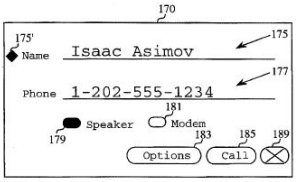


Figure 8b

Exhibit J

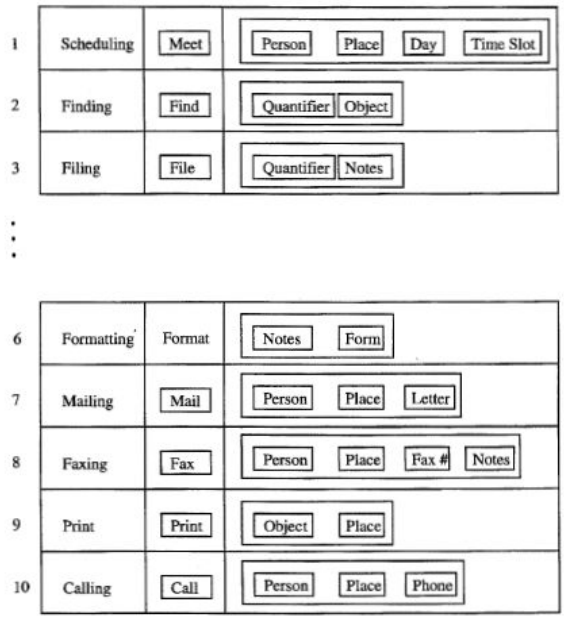
	 *Figure 11c* For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. "Computerized personal organizers tend to be small, lightweight, and relatively inexpensive, and can perform such functions as keeping a calendar, an address book, a to-do list, etc." 1:26-28. "For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot." 14:9-11. "If no hypothesis has been produced, as determined by the answer to the query made at 139 of FIG. 3, then the user may supply a proposed assistance course of action, as suggested at 141 of FIG. 3. This is made more explicit in FIG. 12b. For example, a user proposed course of action

Exhibit J

	can be determined by the process beginning at step 270 of FIG. 12b. As a threshold step, it is asked whether the user wishes to enter a particular action, according to the step noted at step 272 of FIG. 12b. If there is no desire by the user to enter a particular course of action, operation returns to point A of FIG. 3, and the cycle of inquiring whether an implicit assist is desired is made, according to 104 of FIG. 3. Alternatively, if the user does wish to provide or enter a particular action, the process can continue for example with the presentation of a particular list of applicable actions, as indicated at step 274 of FIG. 12b. This approach is graphically illustrated in FIG. 12c, which shows presentation of the list of actions being made as a pull-down menu 170" partially superimposing over window 170." 14:43-61. "[M]eans for enabling a user of the computer system to select an alternative from the compiled list." Claim 5. "[M]eans for updating the database to contain information regarding the selected alternative." Claim 6. "[U]pdating the database to contain information regarding the selected alternative." Claim 13. "Upon accomplishment of the selected assistance action, the database information and any linked information are updated at step 123." 9:13-15. "In the absence of there being any linked fields available, the data obtaining step of 204 is skipped and the data is entered manually, if available. Otherwise, the phone data field will remain vacant as to that particular data element. Operation of updating information and linked information in accordance with step 123 of FIG. 3 is completed with step 208 in FIG. 8a." 12:54-60. *See* Figs. 3, 8a, 12b, & 12c and accompanying text:

Exhibit J

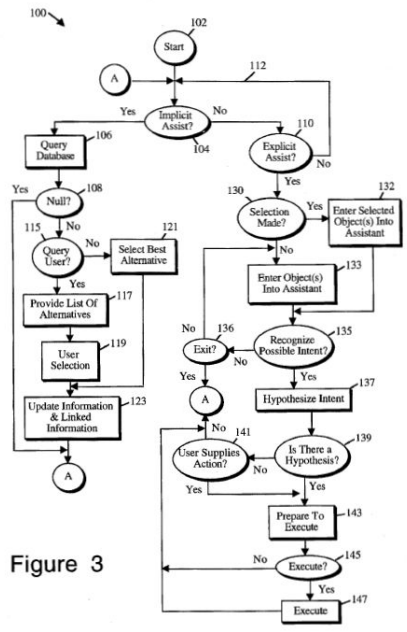
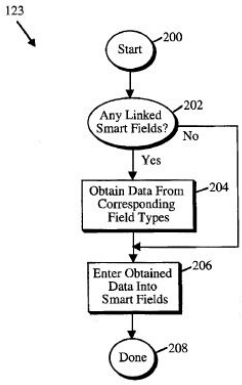
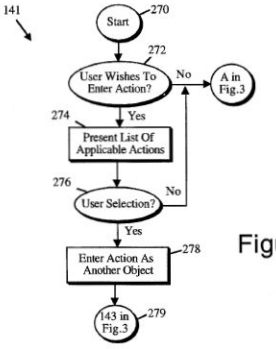
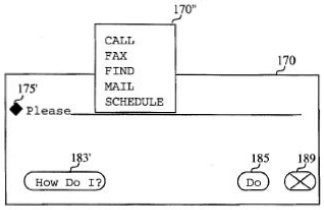
	 Figure 3  Figure 8a  Figure 12b  *Figure 12c* For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. *See* Figs. 6a-6c and accompanying text:

Exhibit J

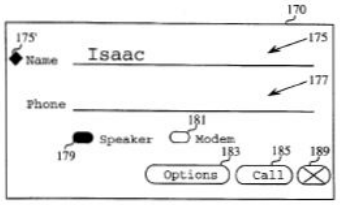


Figure 6a

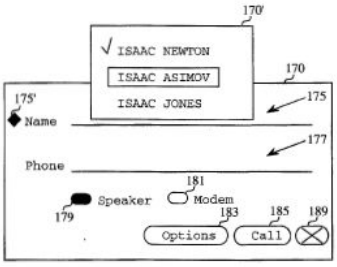


Figure 6b

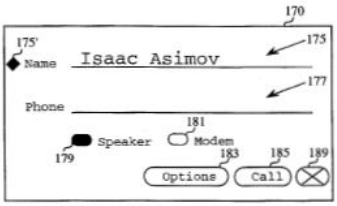


Figure 6c

See 11:40-12:6: "Shown in FIGS. 6a-6c is the process of user selection of a particular assistance option indicated at steps 117 and 119 in FIG. 3. The phone slip window 170 in FIG. 6a is shown with a smart name field 175. The name ISAAC has been recognized in smart field 175 and displays the recognized name in formal font form. As noted above, window 170 in FIG. 4b contains an additional smart field, i.e., "phone" field 177. Additionally, speaker block 179 has been selected, indicating that a tone is capable of being evoked. Alternatively, a modem option, indicated at step 181, can be selected. Further options can be displayed in a pull-down menu button entitled "options" at step 183 which can be presented as a help menu. As already noted above, a "call" activity can be undertaken by selecting "call" button 185 indicated on the face of window 170. Window 170 can be closed simply by selecting the "x" block shown in window 170. Significantly, to the left of the name field is a diamond icon 175' which can be invoked to produce a pull-down menu of selection items (not shown) which permit the user to initiate further assistance operations. FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them,

Exhibit J

	the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user- selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. See, e.g.: "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled 'options' at 183 can be produced as a help menu. Further, a 'call' activity can be undertaken by selecting a "call" button 185 indicated on the face of window 170." 12:61-13:3. "FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. *See* Figs. 3, 6a-c, and 8b and accompanying text:

Exhibit J

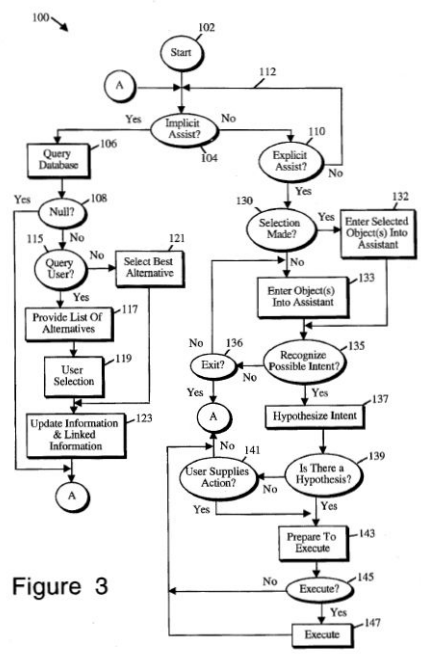


Figure 3

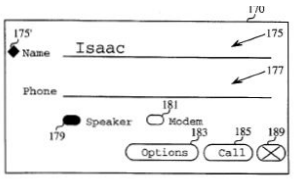


Figure 6a

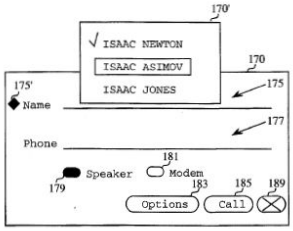


Figure 6b

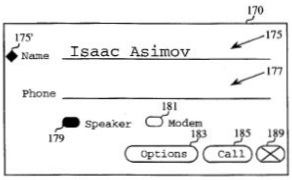


Figure 6c

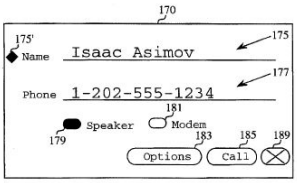


Figure 8b

For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7, 17, and 20.

Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. See, e.g.: Claim 1 disclosure. "In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as

Exhibit J

	indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation." 12:43-54. "FIG. 11a shows the recognition of object(s) process which is part of FIG. 3 at 135, in order to enable recognition of possible user intent. The recognition process is started at step 240 in FIG. 11a. Next, a decision step 242 determines whether the object(s) match at least one template. If not, the process continues at a step 244 which corresponds to step 136 of FIG. 3. If so, the process continues at step 137 of FIG. 3. In substance, the process aims to determine whether the object(s) match at least one of the templates of object combinations set forth in FIG. 11c. FIG. 11b illustrates the object combination under operation, denoted by kind of object. The verb CALL is considered to be an action object and ISAAC is considered to be a person object. The two objects in combination are subject to template comparison. The template in FIG. 11c is effective for organizing in preset form the various object combinations which are capable of further operation as particular functions to be accomplished. FIG. 11c illustrates selected example functions such as scheduling, finding, filing, formatting, mailing, faxing, printing, and calling, just to cite a few of the possibilities." 13:52-14:4. "FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17. "The recognition of possible user intent process called for at 135 in FIG. 3 and expressed in example form at FIG. 11a, calls for a matching operation between particular noted object(s) such as those illustrated in

Exhibit J

	FIG. 11b and those expressed in the template of FIG. 11c." 14:18-22. "Details of one way to carry out the database query process indicated in FIG. 3 at step 106 can be understood in connection with FIG. 5. In particular, FIG. 5 illustrates a frame 180 which is a special case of a frame, referred to commonly as a "type" frame, as the frame refers to a particular type, i.e., the type <PERSON>." 10:49-54. "For example, if it was desired to retrieve all of the frames that were colored red, a typical frame accessor language query would be in the form of: (QUERY (MEMBER-VALUE COLOR ?XRED) and would return a list of frames that have a COLOR slot whose value is red." 11:33-37. "In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation." 12:43-54. "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled 'options' at 183 can be produced as a help menu. Further, a "call" activity can be undertaken by selecting a ''call' button 185 indicated on the face of window 170." 12:61-13:3.

Exhibit J

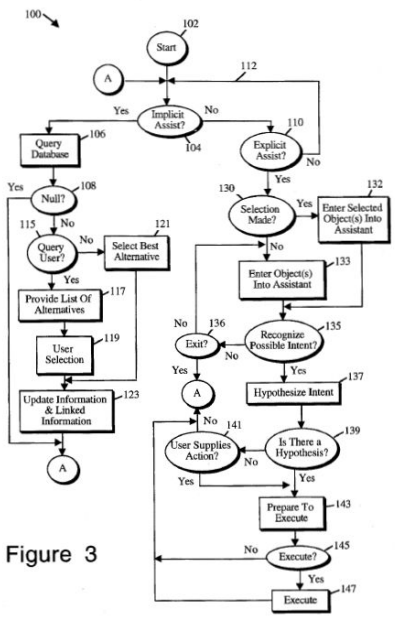
	"However, implicit assist may be indicated not just by entry of an indication in a smart field, but by the happening of any of a number of predefined allowable events which lead to a query of the database at process step 106. A user entry made into a smart field is not the only way computer system 10 is caused to undertake an implicit assist operation. Certain kinds of events on screen 52, for example, such as the writing of a particular indication or word on screen 52 outside of a particular smart field may trigger an implicit assist. In general, implicit assist can be triggered by the happening of any of a number of predefined allowable events." 8:30-41. *See* Figs. 3, 5, 6a-b, 8a-b, 11a-c, & 12c and accompanying text:  Figure 3

Exhibit J

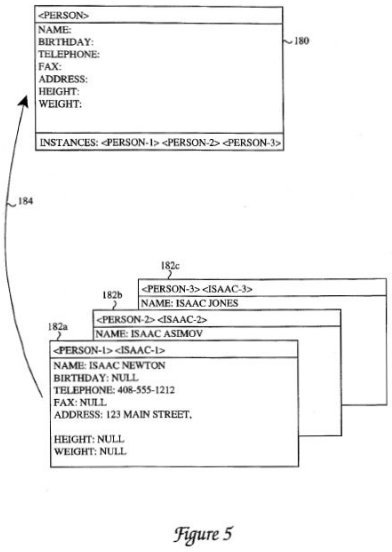


Figure 5

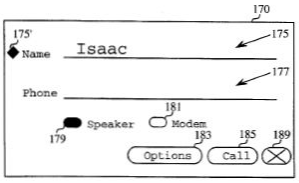


Figure 6a

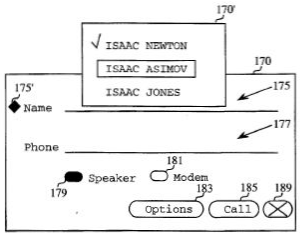


Figure 6b

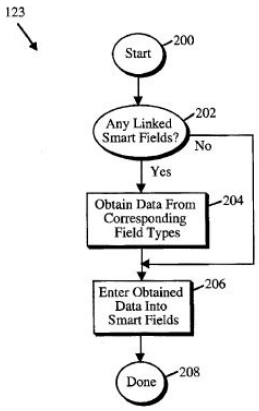


Figure 8a

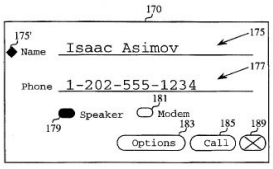


Figure 8b

Exhibit J

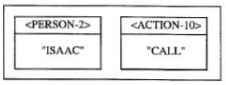
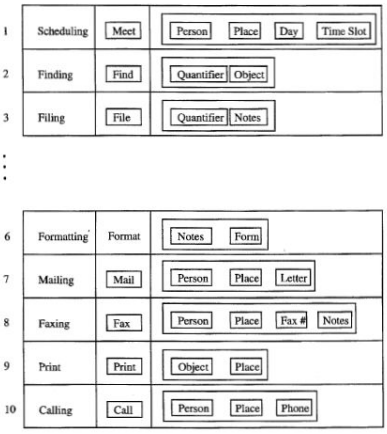
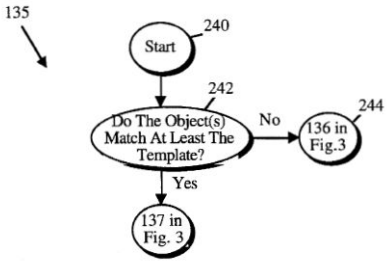
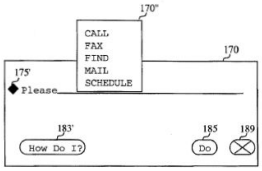
	 *Figure 11b*  *Figure 11c*  Figure 11a  *Figure 12c* For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 10 and 19.
Claim 18	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. See, e.g.: Claim 1 disclosure. "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled

Exhibit J

	'options' at 183 can be produced as a help menu. Further, a 'call' activity can be undertaken by selecting a "call" button 185 indicated on the face of window 170." 12:61-13:3. "FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17. "FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. "FIG. 8a illustrates details of the operation of step 123 of FIG. 3 dealing with the updating of information and linked information in smart fields. In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the

Exhibit J

window 170 under operation. In the absence of there being any linked fields available, the data obtaining step of 204 is skipped and the data is entered manually, if available. Otherwise, the phone data field will remain vacant as to that particular data element. Operation of updating information and linked information in accordance with step 123 of FIG. 3 is completed with step 208 in FIG. 8a."

12:41-54.

See Figs. 3, 6a-c, 8a-b, & 11c and accompanying text:

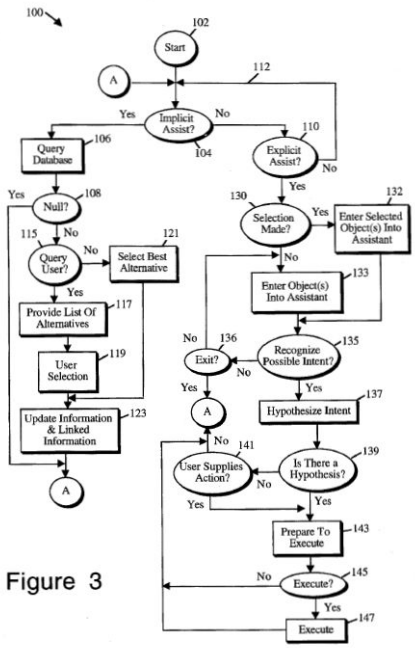


Figure 3

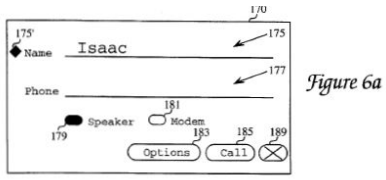


Figure 6a

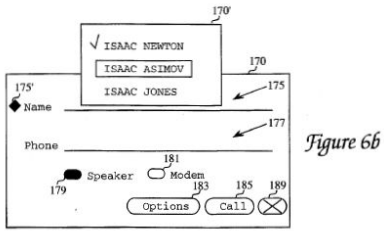


Figure 6b

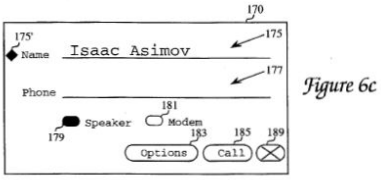


Figure 6c

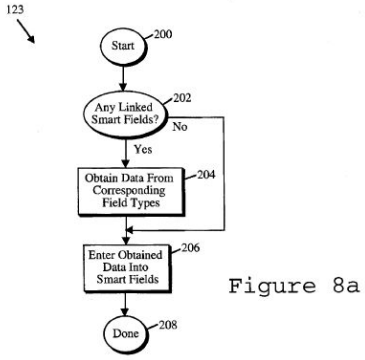


Figure 8a

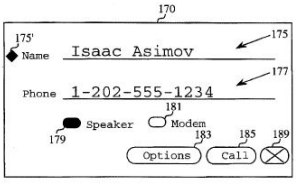


Figure 8b

Exhibit J

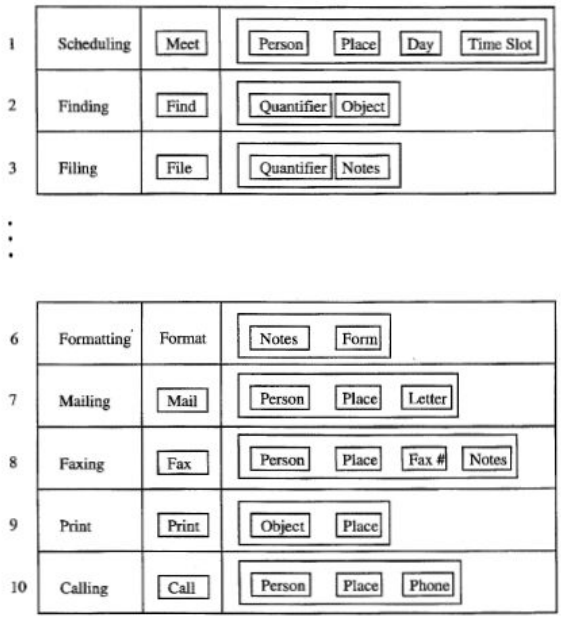
	 *Figure 11c* For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.
Claim 19	
A method according to claim 1, wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.	Luciw '735 discloses claim 1. *See* claim 1. Luciw '735 further discloses this element. See, e.g.: Claim 1 disclosure. "FIG. 8b illustrates the completion of a successful data updating operation performed according to the procedure of FIG. 8a. Speaker block 179 in FIG. 8b has been selected, to enable tone evocation of the phone number as for tone dialing in connection with a telephone call. Alternatively, a modem option, indicated at 181, can be selected." Further options can be displayed in "[a] pull-down menu button entitled 'options' at 183 can be produced as a help menu. Further, a 'call' activity can be undertaken by selecting a "call" button 185 indicated on the face of window 170." 12:61-13:3.

Exhibit J

	"FIG. 11c further provides example kinds of action objects, such as meet, find, file, format, mail, fax, print, and call. The Figure provides examples of allowable combinations of objects which correspond to the indicated functions and actions. For example, essential objects for scheduling a meeting include four objects, such as person, place, day, and time slot. Finding activities require the combination of two objects, which are for example a quantifier and an object. Filing requires a quantifier and notes. Formatting require notes and form, mailing requires a person, a place, and a letter. Faxing requires a person, a place, a fax number, and notes. Printing requires an object and a place. Calling requires a person, a place, and a phone number." 14:5-17. "FIG. 6b illustrates a presentation of assistance options to the user in connection with step 117 in FIG. 3. Responsive to the recognition of the name ISAAC, the assistance process has produced a list of alternatives by earlier query of the database per step 106 in FIG. 3. In particular, three ISAAC are presented for selection of one of them, the presentation being made in an overlay window 170, positioned partially over the underlying window 170. The user-selected "ISAAC ASIMOV" is shown having been marked for selection by a rectangle indicating a highlighting operation. FIG. 6c illustrates the completion of the selection process, with the full name in formal font of ISAAC ASIMOV being presented in the name field 175 of window 170." 11:60-12:6. "FIG. 8a illustrates details of the operation of step 123 of FIG. 3 dealing with the updating of information and linked information in smart fields. In the earlier example of FIG. 6c in which it was decided that Isaac Asimov was the desired ISAAC, the phone information in window 170 had not yet been entered. This information may be available and can be accessed according to the process of FIG. 8a. The process starts at 200 and immediately checks the data base for any linked smart fields as indicated at 202. If there are applicable smart fields which contain the desired phone number information, this data is obtained from the corresponding linked field types as suggested at 203. Then, as suggested at 206, the data obtained is entered into the applicable smart field of the window 170 under operation. In the absence of there being any linked fields available, the data obtaining step of 204 is skipped and the data is entered manually, if available. Otherwise, the phone data field will remain vacant as to that particular data element. Operation of updating information and linked information in accordance with step 123 of FIG.

Exhibit J

	3 is completed with step 208 in FIG. 8a." 12:41-54. *See* Figs. 3, 6a-c, 8a-b, & 11c and accompanying text:

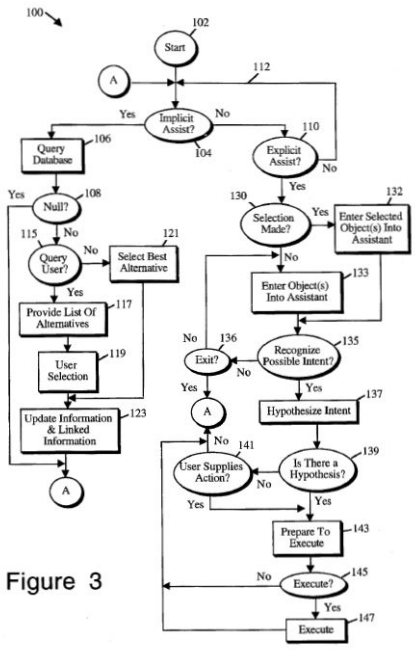


Figure 3

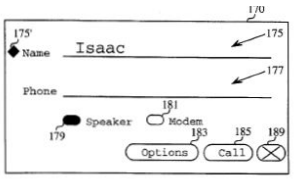


Figure 6a

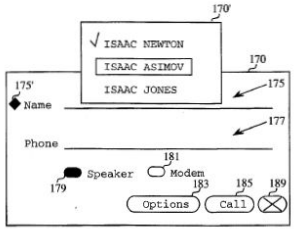


Figure 6b

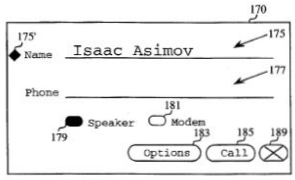


Figure 6c

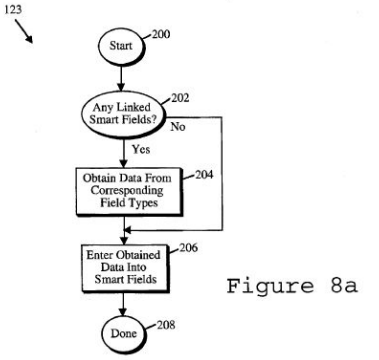


Figure 8a

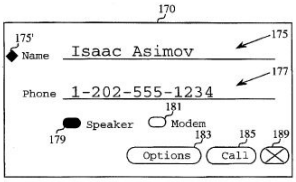


Figure 8b

Exhibit J

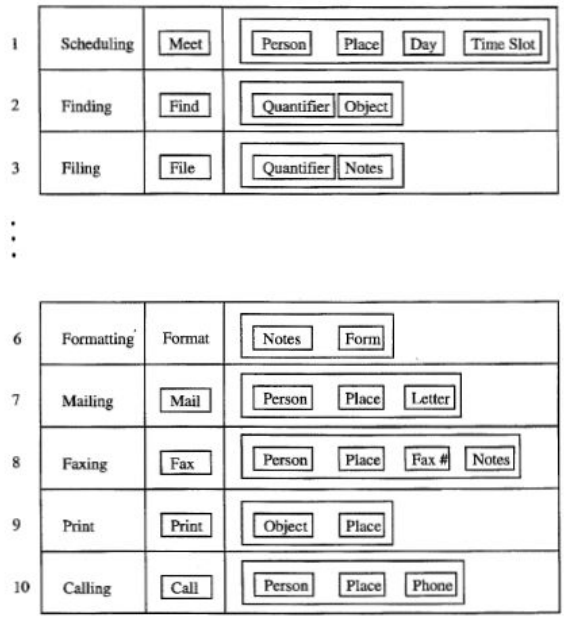
	 *Figure 11c* For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 3, 12, 13, 18, and 21.
Claim 23	
At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for finding data related to the contents of a document using a first computer program running on a computer, the processes comprising:	To the extent the preamble is limiting, Luciw '735 discloses the preamble. *See* claim 1.
displaying the document electronically using the first computer program;	Luciw '735 discloses this element. *See* claim 1
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to	Luciw '735 discloses this element. *See* claim 1

Exhibit J

find second information related to the first information;	
retrieving the first information;	Luciw '735 discloses this element. *See* claim 1
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types of the first information, and (ii) performing an action using at least part of the second information;	Luciw '735 discloses this element. *See* claim 1
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Luciw '735 discloses this element. *See* claim 1
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Luciw '735 discloses this element. *See* claim 1
Claim 30	
At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising:	Luciw '735 discloses claim 23. *See* claim 23.

Exhibit J

providing a prompt for updating the information source to include the first information.	Luciw '735 discloses this element. *See* claim 8.

Exhibit K

Claim Chart Applying Eudora Against the '843 Patent

Qualcomm Inc.'s Eudora product was offered for sale, sold, and/or publicly used in the United States at least by February 1997. It therefore constitutes prior art under pre-AIA 35 U.S.C. §§ 102(a), 102(b), and 103. As shown below, Eudora anticipates and/or renders obvious claims 1, 8, 13, 15, 17-19, 23, and 30 of the '843 patent.

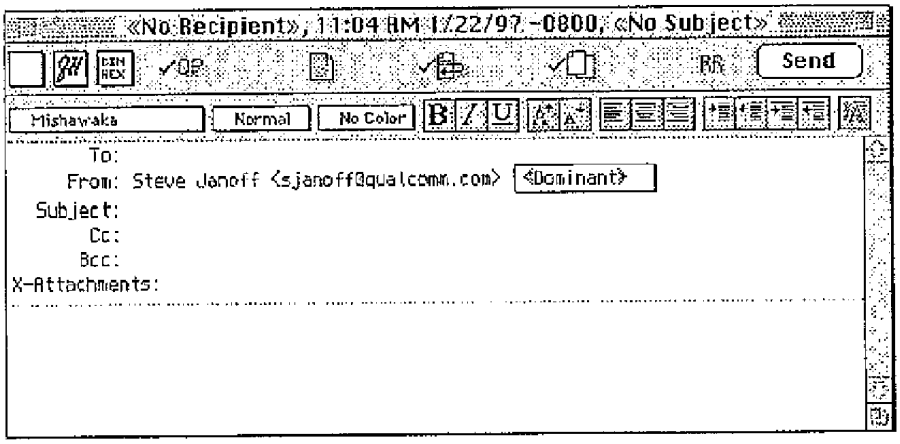
"Obviousness Statement" - To the extent that the Judge or Jury finds that Eudora does not teach an element either expressly or inherently, then the claim element is obvious to a POSITA based on the state of the art (*see, e.g.*, Section V of my Report), including the admissions of the prior art functionalities and motivations to combine those prior art functionalities in the '843 patent, as well as the motivations to combine and understandings of a POSITA discussed in my Report (*see, e.g.*, Section IX of my Report and Exhibit U), in light of the teachings of, at least, the prior art listed and discussed in Exhibit U, and each prior art system and/or reference listed in my Report, including, without limitation, Pandit, Chalas, Domini, Hachamovitch, Tso, Person, CyberDesk System (including specific publications describing aspects of the CyberDesk System), Eudora System (including specific publications describing aspects of the Eudora System), Apple Data Detectors System (including specific publications describing aspects of the Apple Data Detectors System), LiveDoc System (including specific publications describing aspects of the LiveDoc System), Newton System (including specific publications describing aspects of the Newton System), Microsoft Outlook 97 (including specific publications describing aspects of Microsoft Outlook 97), Selection Recognition Agent System (including specific publications describing aspects of the Selection Recognition Agent System), and Microsoft Word 97 (including specific publications describing aspects of Microsoft Word 97).

Evidence of the availability of Eudora as well as the design and operation of Eudora include the following:

- "Review: Eudora PRO 3.0" by Robert Madill (February 1997) (hereinafter, "Eudora Review")
- "Version 3.1 for Macintosh User Manual – Eudora Mail Pro" (June 1997) (hereinafter, "Eudora Mac Manual")
- "Version 3.0 for Windows User Manual – Eudora Mail Pro" (June 1997) (hereinafter, "Eudora Windows Manual")

'843 Patent Claims	**Disclosure**
Claim 1	
A computer-implemented method for finding data related to the contents of a document using a first computer program running on a computer, the method comprising:	To the extent the preamble is limiting, Eudora discloses the preamble. See, e.g.: "**Using the Composition Window** The composition window consists of the title bar, the icon bar, the formatting toolbar, the message header, and the message body. The formatting toolbar can be turned on or off."

Exhibit K



The composition window

Eudora Mac Manual at 23.

"Using the Composition Window

The composition window consists of the title bar, the Toolbar, the message header, and the message body."

The composition window

Eudora Windows Manual at 19.

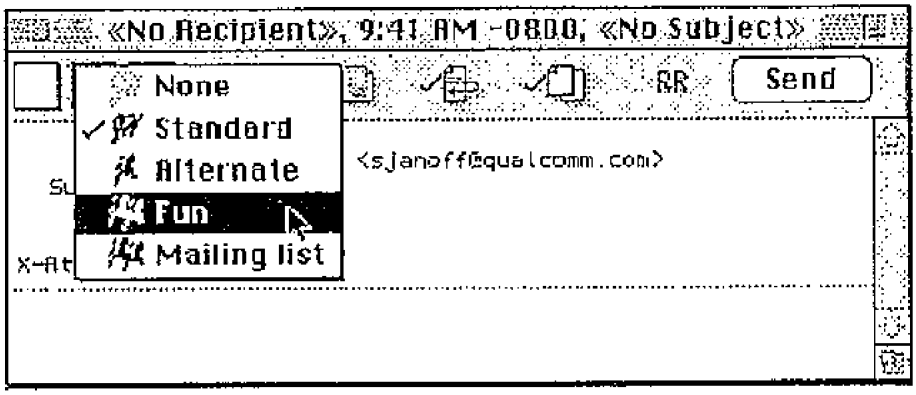
"**Message Header**

Exhibit K

	Outgoing mail headers consist of six fields: **To, From, Subject, Cc, Bcc,** and **X-Attachments**. Each field is described below. The **To, Subject, Cc,** and **Bcc** fields can be directly edited. To move the cursor from field to field, press the tab key or click in the desired field with the mouse. ****** **Message body** After filling in the header fields, move the insertion point to the space below the message header. Type the body of the message here. For information about formatting your message text, see the sections 'Formatting Text' and 'Formatting Toolbar.' " Eudora Mac Manual at 27-28. **"Message Header** Outgoing mail headers consist of six fields: **To, From, Subject, Cc, Bc** and **Attachments**. Each field is described below. The **To, Subject, Cc,** and **Bcc** fields can be directly edited. To move the cursor from field to field, press the tab key or click in the desired field with the mouse. ****** **Message body** After filling in the header fields, move the insertion point to the space below the message header. Type the body of the message here. For information about formatting your message text, see the section 'Formatting Text' " *Eudora Windows Manual.* at 23-24. "**Using Signatures** A signature is a few lines of text that are automatically added to the end of an outgoing message when it is sent. A signature can be whatever you want, but it is mostly used to give contact information (telephone, address, etc.). You can have as many signatures as you want. *Note: You signature is not displayed in the Eudora message window, but is added to the end of the message when it is sent.* ****** To include a signature in an outgoing message, select the signature you

Exhibit K

want from the Signature popup on the icon bar.



Setting the signature for a particular message

To include a particular signature in all of your outgoing messages, select a signature in the Sending Mail Settings (for your dominant account only), or in the Personality Extras Settings (for any of your personalities). You can change this for a particular message by selecting a different signature or **None** from the Signature popup on the icon bar of the composition window.

Note: If you are using a default stationery file, or if you send or reply to a message with a stationery file, then the signature saved with the stationery file will override your signature selections in the Sending Mail Settings and Personality Extras Settings. For more details, see the section 'Using Stationery Files.' "
Eudora Mac Manual at 34-35.

"Using a Signature

A signature is a few lines of text that are automatically added to the end of an outgoing message when it is sent. A signature can be whatever you want, but it is mostly used to give contact information (telephone, address, etc.). You can have as many signatures as you want.

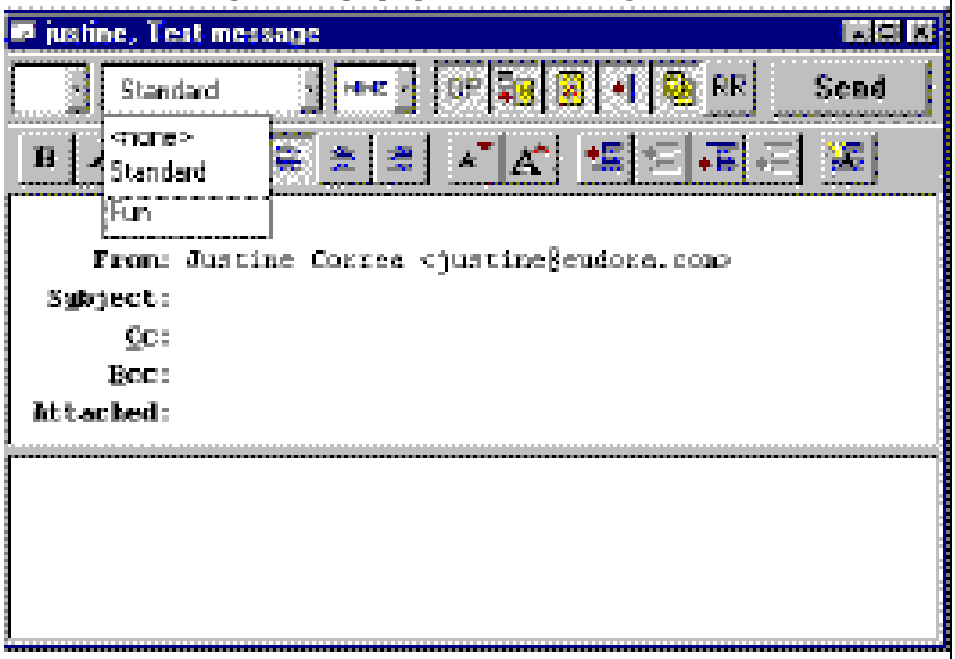
Note: Your signature is not displayed in the Eudora message window, but is added to the end of the message when it is sent.

Exhibit K

Creating the Standard signature

******

To include a signature in an outgoing message, select the signature yo want from the Signature popup on the message toolbar.



Setting the signature for a particular message

To include a particular signature in all of your outgoing messages, select a signature in the Sending Mail Options (only for your dominan account), or in the Personalities Options (for any of your personalities). You can change this for a particular message by

Exhibit K

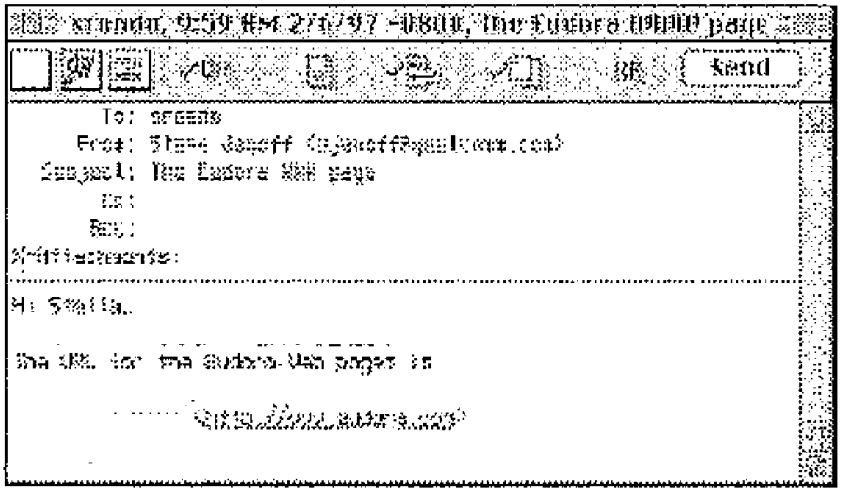
selecting a different signature or **None** from the Signature popup" *Eudora Windows Manual.* at 27-28.

"**Inserting the Contents of a Text File into a Message**

The contents of a text file can be inserted directly into a message (and then edited if desired). To insert a text file into a message, put the cursor where you want the text inserted, and select Attach Document… from the Message menu. Then select the text file you want and click on the Insert button. The text from the file is inserted into your message and you can edit it as normal."
Eudora Mac Manual at 42.

"**Including a URL in a Message**

To include a *hot link* in a message—also known as a URL, for Uniform Resource Locator—enclose it with *less than* and *greater than* signs (angle brackets) to ensure that your recipient's e-mail application can identify it as a URL. For example, <http://www.eudora.com>. The URL is automatically highlighted as an active URL in your message window. The default highlighting is blue underlined text."



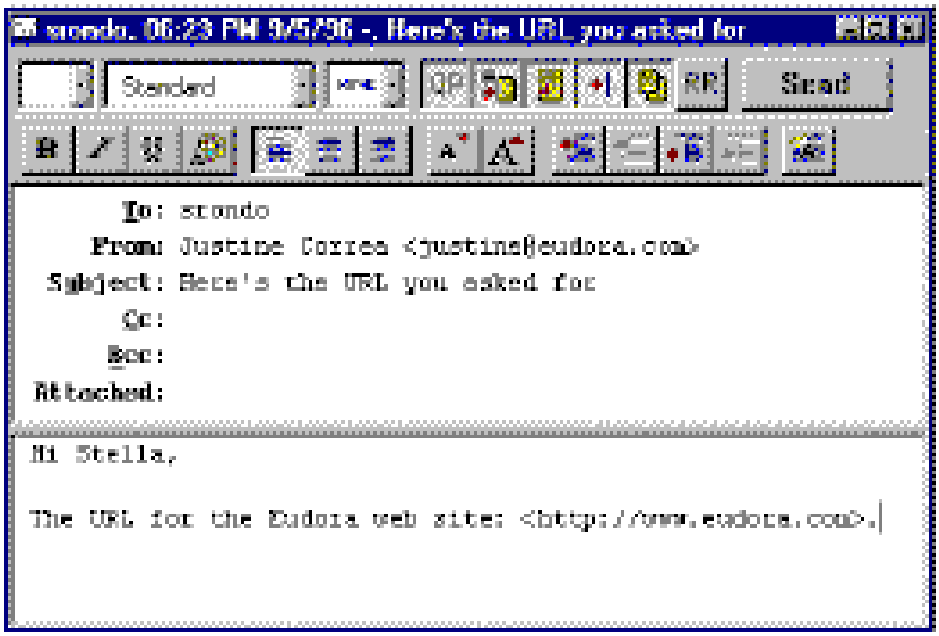
A URL in an outgoing message

Eudora Mac Manual at 42.

"Including a URL in a Message

To include a URL (Uniform Resource Locator) in a message, enclose it with *less than* and *greater than* signs to ensure that your recipient's application can identify it as a URL. For example, <http://www.eudora.com>."

Exhibit K



A URL in an outgoing message

Eudora Windows Manual at 33.

"**Checking Your Spelling**

Eudora includes the Spellswell 7 Spelling Checker, developed by Working Software, Inc. Because it is an Apple Events Word Services Suite application, Spellswell 7 can be used with Eudora. This section describes the spelling checker's basic functions when it is used with Eudora. For more information on Spellswell 7, how it functions with other applications, specialized dictionaries, etc., see the Spellswell 7 User Manual. It is located in the Documentation folder within the Eudora Pro Application Folder.

The spelling checker includes a customizable 93,000+ word dictionary. It can be used to check for spelling mistakes and typographical errors in message composition windows, text files, and signature files.

Besides finding ordinary misspellings, the spelling checker has many additional options for finding errors. For details, see the section "Spelling Options."

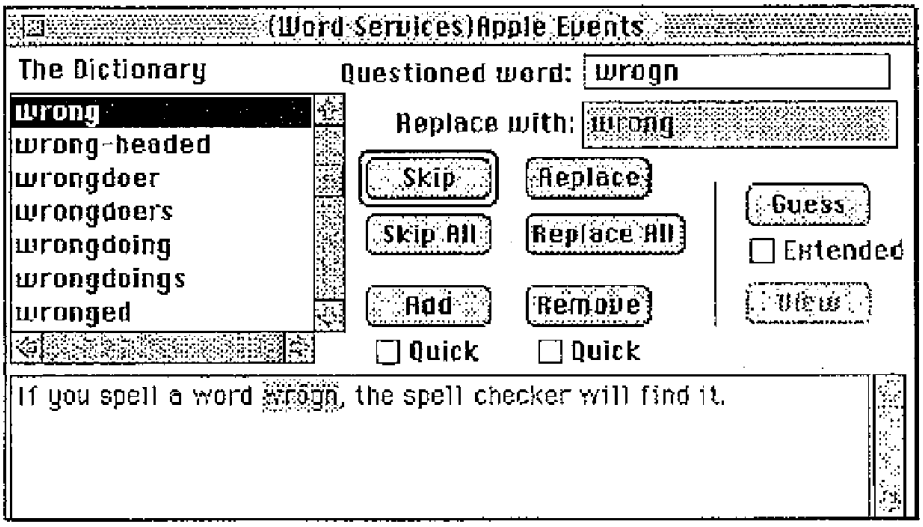
To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the

Exhibit K

parts of the body that are identified as quoted text. You can also highlight a word or a block of text to check only that text and not the rest of the message.

If no misspellings are found, the spelling checker quits.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.



The Check Spelling dialog

The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check.

Note: If you select **Check Spelling** from the **Edit** menu and an error dialog appears telling you that the application (Spellswell cannot be found, it means that the link between Eudora and Spellswell has somehow been broken and must be reestablished. This is rare, but easy to fix. Take these steps: Select **Add Word Service…** from the **Edit** menu. Browse through the dialog and find the Spellswell application on your hard drive. Double-click on the application, or click once on it to highlight it and then click **Open**. Now Spellswell is available again: you can select **Check Spelling** from the **Edit** menu to spell-check your messages.

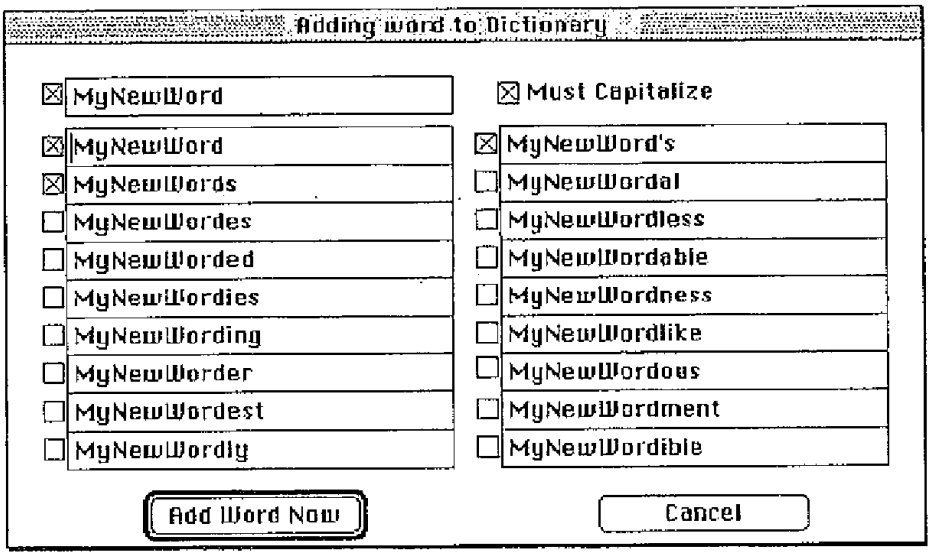
Exhibit K

	The Check Spelling Dialog The Check Spelling dialog allows you to skip a questioned word, replace it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below. Questioned word A word that is not found in the spelling checker dictionary. Replace with Replace the questioned word with the word in this field. You can select a word from the Dictionary/Guesses field, or type a new one. Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. **Skip (All)** Ignore this occurrence of the questioned word. If you use **Skip All**, you ignore this and all subsequent occurrences of the questioned word. **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View**

Exhibit K

Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field.

Add
Add the questioned word to the dictionary. If the **Quick** option is on, then the questioned word is added to the dictionary immediately when you click this button. If this option is off, the **Adding word to Dictionary** dialog is displayed. This dialog provides you with options for adding the word and its various forms to the dictionary.



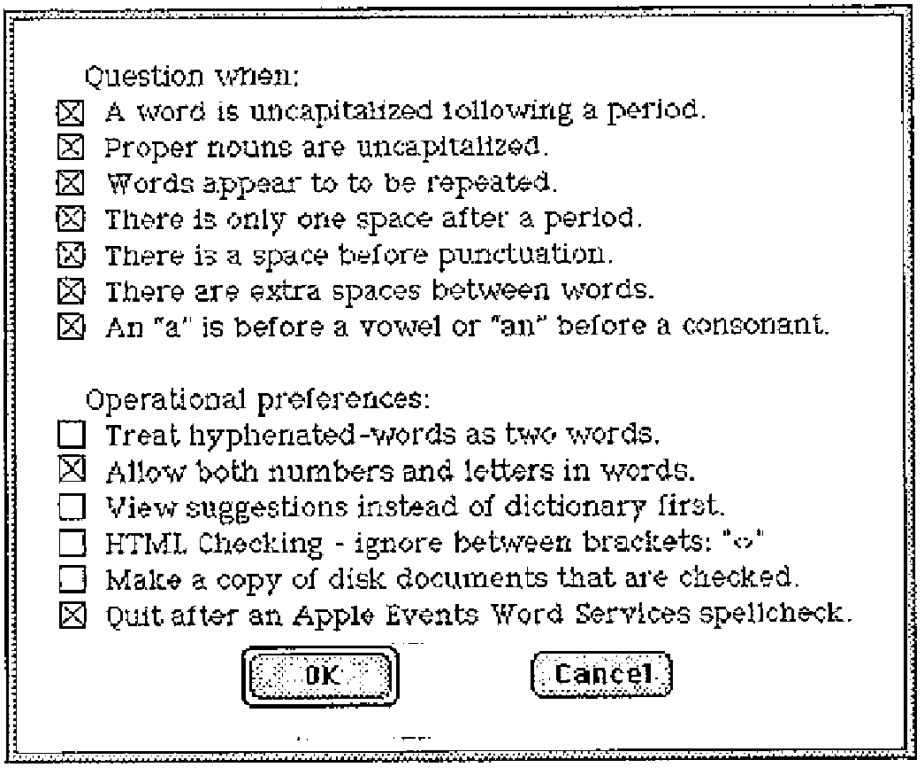
The Adding word to Dictionary dialog

Remove
Remove a word from the dictionary. If the **Quick** option is on, then the word is deleted from the dictionary immediately when you click this button. If this option is off, you are prompted to confirm the deletion.

Spell Checking Options

There are many spell checking options to use when checking for errors. To set these options, select **Options…** from the Spellswell **File** menu. The available options are described below.

Exhibit K



Spelling Options Dialog Box

Question when a word is uncapitalized following a period
Questions the capitalization of the first word following a period, question mark, exclamation point, or carriage return/new line character.

Not: The spelling checker does not distinguish between a period ending a sentence and a period ending an abbreviation.

Question when proper nouns are uncapitalized
Questions the capitalization of words that appear to be proper nouns.

Question when words appear to to be repeated
Questions words that appear twice in sequence.

Question when there is only one space after a period
Questions period, question marks, and exclamation points followed by a single space (represented by a square in the **Questioned word** field).

Question when there is a space before punctuation
Question occurrences of spaces before a punctuation mark (comma, semicolon, period, question mark, or exclamation point).

Question when there are extra spaces between words

Exhibit K

	Questions occurrences of extra spaces between words. **Question when an 'a' is before a vowel or 'an' before a consonant** Questions usage of the indefinite articles 'a' and 'an.' **Treat hyphenated-words as two words** Treats improperly hyphenated words as separate words. **Allow both numbers and letters in words** Allows mixed numbers and letters. Ignores words that contain both upper and lower case characters or words that contain numbers. **View suggestions instead of dictionary first** Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word. **HTML checking - ignore between brackets: '<>'** Ignores URLs between brackets. **Make a copy of disk documents that are checked** This option does not apply to spell checking from Eudora. It automatically creates a backup copy of your original document with 'sbk' appended to the file name. **Quit after an Apple Events Word Services spellcheck** Quits the spelling checker once the document is checked." Eudora Mac Manual at 42-47. **"Checking Your Spelling** Eudora includes a built-in spelling checker. It can be used to check for misspellings in the body of current message composition windows, text files, and signature files. It includes a built-in dictionary and also allows for the creation of a custom user dictionary. Additionally, it can be configured to ignore capitalized words, words with numbers, and mixed case words, to report mixed case and double (repeated) words, and to suggest alternative spellings. *Note: Changes to the spell checking options can be made in the Options dialog (Check Spelling) or by selecting* ***Options*** *from the* ***Check Spelling*** *dialog.* See Appendix A for information on how to get dictionaries for languages other than U.S. English.

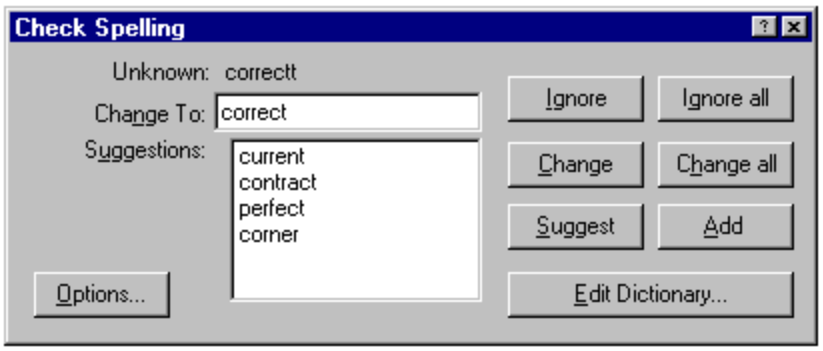
Exhibit K

To automatically check spelling when you send or queue a message, turn on the **Check when message queue/send selected** option in the Spell Checking Options. If this is on, when you send or queue a message the message is checked for spelling errors. If you go through the spell checking process, the message is automatically sent or queued. If you click Cancel, or leave spelling errors in the message, a dialog is displayed asking you if you still want to send or queue the message. If you don't want that dialog to be displayed, turn on the Don't warn me anymore option (this can also be set in the Spell Checking Options).

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

To correct the misspelled word, type the correct spelling of the word in the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check.

Check Spelling Dialog
The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below.

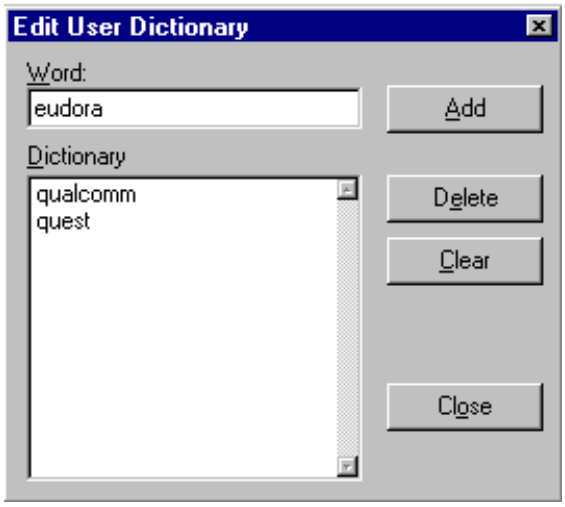
Unknown Field

Exhibit K

	An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions. **Ignore Button** This button causes the spelling checker to ignore this occurrence of the unknown word. **Ignore all Button** This button causes the spelling checker to ignore this occurrence and all subsequent occurrences of the unknown word. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word. **Suggest Button** This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* **Add Button** This button adds the unknown word to your custom user dictionary.

Exhibit K

Edit Dictionary Button
This button displays the Edit User Dictionary dialog.



The Edit User Dictionary dialog

The Edit User Dictionary dialog lists all of the words in your user dictionary in the Dictionary field. It also allows you to add words to or delete words from your personal user dictionary, or even clear the entire dictionary.

Note: Words in the user dictionary are saved in all lower case.

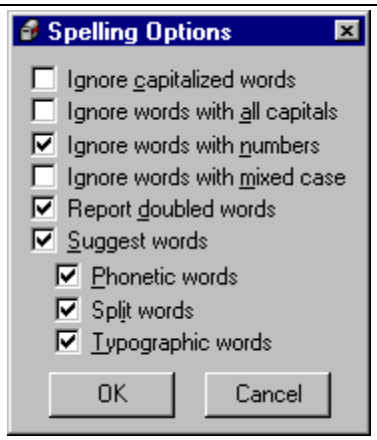
To add a word to the dictionary using this dialog, type the correct spelling of the word in the Word field and click the **Add** button. The word is then added to the dictionary and displayed in the Dictionary field.

Note: The Add button in this dialog works the same as the Add button in the Check Spelling dialog.

To remove a word from the user dictionary, type it in the Word field or locate it in the Dictionary field and single-click on it to display it in the Word field. Then, click the **Delete** button. To delete the entire user dictionary, click on the **Clear** button. You will then be prompted to confirm the deletions. If you click **Yes**, all of the words are deleted from the user dictionary.

Options Button
This button displays the spell checking Preferences dialog.

Exhibit K



Spell checking Preferences dialog

The spell checking Preferences dialog lists the six spell checking options. A check mark in the box next to the option name indicates that it is turned on.

Note: The spell checking options can also be modified in the Options dialog (Spell Checking).

The available options are as follows:

Ignore capitalized words
Ignores words that start with capital letters, such as proper nouns.

Ignore words with numbers
Ignores words that contain numbers.

Ignore words with mixed case
Ignores words that contain both upper and lower case characters.

Report doubled words
Reports words that appear twice in sequence in text and identifies them as Doubled words.

Make Suggestions
Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options.

Note: If Eudora doesn't have suggestions in its dictionary, then none are listed."
Eudora Windows Manual at 33-37.

"Included with Eudora Pro[TM] 3.0 is the spell check application,

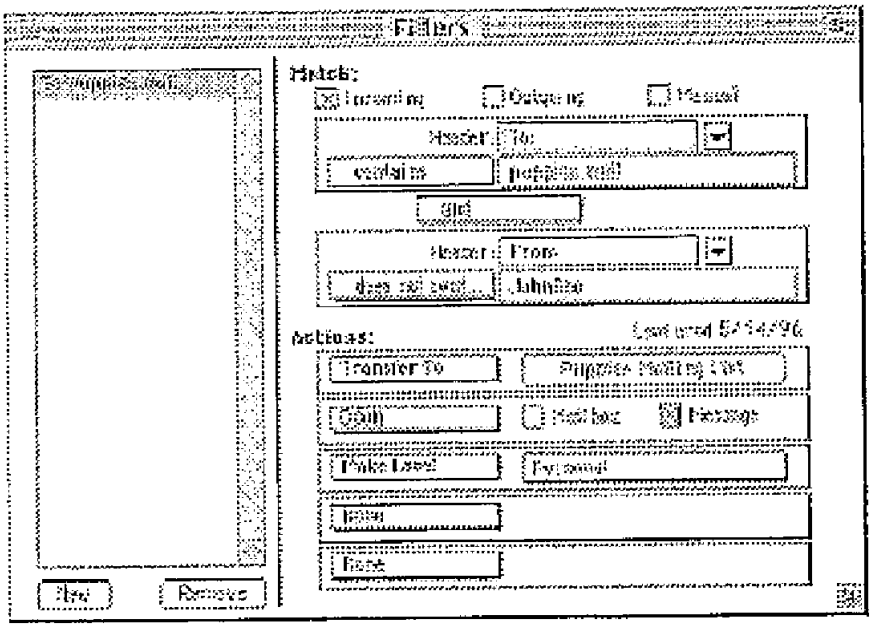
Exhibit K

	Spellswell from **Working Software**, which performs simple checks for grammar and spelling errors. The 93,000+ word dictionary is considerably robust for its size, has an intuitive interface, and is user-expandable. In a world where spelling errors are an embarrassment, this feature is a welcome addition." Eudora Review at 4. "**Using Active URLs** Any string of text that Eudora recognizes as a 'hot link,' also known as a URL (Uniform Resource Locator: http, ftp, gopher, ph, finger, etc.), is active. Active URLs are normally highlighted in blue text and underlined, and are often enclosed by less-than and greater-than signs, '<>.' You can hold down the command key and click on a URL (or just double-click on it) to open a World Wide Web location, transfer a file, do a gopher search, use the finger tool, etc. To specify what Internet application you want to use for a URL type, hold down the option key and double-click on the URL. A standard file dialog is displayed. Select the application you want, then click on **Open**. The application is opened and goes to the selected URL. The next time you select a URL of that type, the same application is automatically used. Sometimes users will send you messages containing hot links that are poorly formatted, so that Eudora won't recognize them as URLs. Eudora won't highlight these text strings in blue and underline, but you can still use them as active links. Simply select the entire string of text that you think is a hot link, and choose **Open Selection** from the **File** menu, or hold down the command key and click on the selection. We recommend that you encourage your correspondents to enclose hot links in angle brackets, < >, like this: <**http://www.eudora.com/new.html**>. This will help Eudora and other software to recognize the hot link." Eudora Mac Manual at 64-65. **"Using Active URLs** Any string of text that Eudora recognizes as a URL (Uniform Resource Locator: http, ftp, gopher, ph, finger, etc.) is active. Double-click on a URL to open a World Wide Web location, transfer a file, do a gopher search, use the finger tool, etc. URLs are highlighted and underlined to show that they are active (32-bit Eudora only).

Exhibit K

	To setup Eudora to automatically open a new message when you use a mailto link within a Netscape Web browser, turn on the **Intercept Netscape mailto URLs** option in the Miscellaneous Options. (Be sure you are not running the Mailto Watcher application at the same time.)" *Eudora Windows Manual* at 50. "Speaking of color, a real bonus of Eudora Pro 3.0 is its ability to acce 'Embedded URLs' in the body of a message. Any URL you enter nto th body of your mail text (designated by ,. Brackets) is automatically colored blue and underlined." Eudora Review at 3. "[I]ncoming e-mail correspondence containing URLs written in the standard 'http://....' format are highlighted for your use, no matter what application the sender uses to generate the message." Eudora Review at 4. "**Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." To open the Filters window, select **Filters** from the **Special** menu. The Filters window is displayed, and any filters you have created are listed on the left.

Exhibit K



The Filters window with a sample filter

To create or modify a filter, first click on the **New** button or select an existing filter.

Second, select the options for how you want the filter to be used: as an automatic filter to be invoked on any **Incoming** and/or **Outgoing** mail, and as a **Manual** filter that can be invoked when you select **Filter Messages** from the **Special** menu. Any combination of these options works.

Third, define the criteria for the filter: use the header item popups and the text fields to specify which header items should include a particular string of text. You can define two related terms for the criteria so that your filter is as specific as possible (see the section 'Filter Criteria').

Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters.

When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered.

Note: Filters are automatically named based on the criteria for the filter. They are invoked in order from top to bottom, and you can re-order them by dragging a filter up or down in the list.

Exhibit K

	To print your filters, open the Filters window and select **Print…** from the **File** menu. The standard Print dialog is displayed so that you can make your print selections. You can change the width of the filters list to create more or less space for your list. To do this, put the arrow over the vertical dividing line to the right of the list, until the pointer changes to a resize cursor [], then hold down the mouse button and drag the line to the left or right. Filter Criteria (the Match Area) Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup. Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself (this is helpful if you want to search for a header item that does not appear on the menu, such as X-Priority). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ● «Body» ● «Personality» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **BLAH BLAH BLAH BLAH** option); the «Body» option searches the message body; and the «Personality» option searches the name of the personality associated with the message. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are: **contains** or **does not contain** If the specified header item contains or does not contain the text string, filter the message. **is** or **is not** If the specified header item is or is not a complete match of the text string, filter the message.

Exhibit K

starts with or **ends with**
If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored.

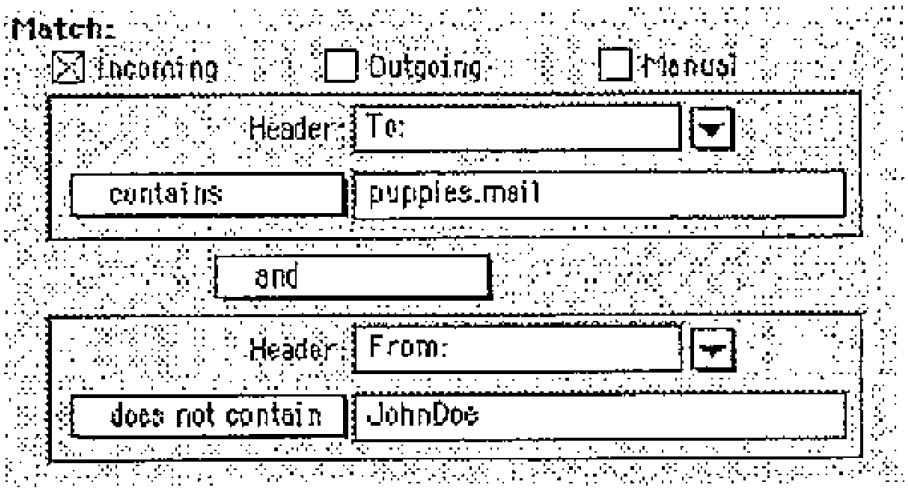
appears or **does not appears**
If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain.

intersects nickname
If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message.

Use the **Text** fields to specify the text strings that the filter is searching for.

Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.

*Note: Be sure not to enter a header label in the **Text** field as part of the text string (e.g., **To:**, **From:**, **Bcc**☺. For example, to filter all messages from **Justine**, do not enter **From: Justine** in the Text field. Rather, select **From:** in the Header field, and enter simply **Justine** in th eText field (choosing, for example, **contains** from the Match Type popup). As another example, not this Filters window Match Area:*



Sample Match Area in Filters window

Exhibit K

	Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches the first term, filter it *unless* the message also matches the second term, in which case *do not* filter it. (This lets you exclude certain variations of the first term.) Filter Actions All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to messages. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages. Label colors and names are set in the Macintosh Labels control panel and the Eudora Labels Settings. **Make Personality** Assigns the selected personality to messages. For outgoing messages, the

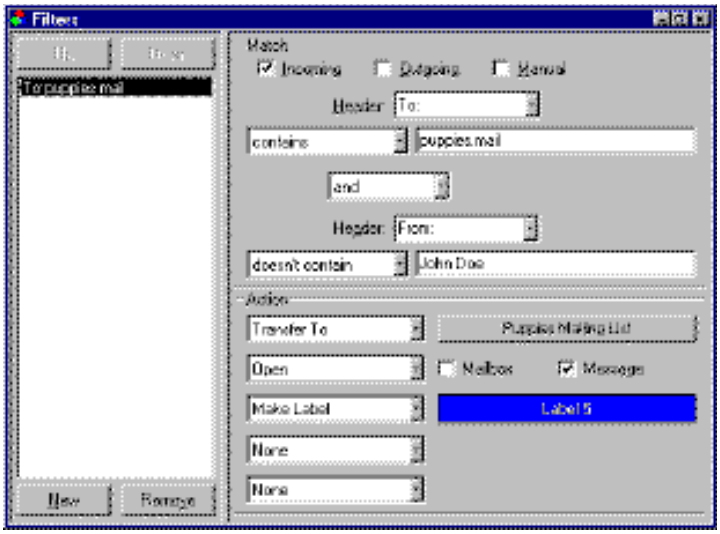
Exhibit K

	message is sent from the assigned personality. For incoming messages, all your responses to the message will be from the assigned personality until you change the personality associated with the incoming message or your response. For more information, see the section 'Using an Alternate E-mail Account.' **Make Subject** Assigns the new subject to messages. If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol to stand for the old subject if you want to add the new subject to the old subject. For example, entering **New Subject [was &]** results in **New Subject [was Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Settings. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. One typical use of this action is to reply to specific

Exhibit K

	senders with stationery telling them that you're on vacation: "I'm out till the 10th. I'll reply to your message when I get back." For more details, see the section 'Using Stationery Files.' **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the filter list). The **Last used** field displays the date the filter was last used on a message. This helps you identify filters that are no longer useful and can be safely deleted." Eudora Mac Manual at 82-87. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." To open the Filters window, select **Filters** from the **Tools** menu. The Filters window is displayed, and any filters you have created are listed on the left.

Exhibit K



The Filters window with an example filter

To create or modify a filter, first click on the **New** button or select an existing filter.

Second, select the options for how you want the filter to be used: as an automatic filter to be invoked on any **Incoming** and/or **Outgoing** mail, and as a **Manual** filter that can be invoked when you select **Filter Messages** from the **Special** menu. Any combination of these options works.

Third, define the criteria for the filter: use the header item popups and the text fields to specify which header items should include a particular string of text. You can define two related terms for the criteria so that your filter is as specific as possible (see the section 'Filter Criteria').

Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters.

*Note: Filters are automatically named based on the criteria for the filter. You can re-order them using the **Up** and **Down** buttons above the list.*

When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered.

You can change the width of the filters list to create more or less space

Exhibit K

for your list. To do this, put the arrow over the bar to the right of the list and drag the line to the left or right.

Filter Criteria (the Match Area)

Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup.

Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows:

- To
- Cc
- From
- Subject
- Reply-To
- «Any Recipient»
- «Any Header»
- «Body»

The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **Blah Blah Blah Blah** option, and the «Body» option searches the message body.

Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are:

contains or **doesn't contain**
If the specified header item contains or does not contain the text string, filter the message.

is or **is not**
If the specified header item is or is not a complete match of the text string, filter the message.

starts with or **ends with**
If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored.

appears or **doesn't appear**
If the header item appears or does not appear in the message, filter the

Exhibit K

	message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain. **intersects nickname** If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message. Use the **Text** fields to specify the text strings that the filter is searching for. *Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.* Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches both the first and second terms, do not filter it. (This lets you exclude certain variations of the first term.) **Filter Actions** All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to message summaries.

Exhibit K

	Make Priority Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages. **Make Subject** Assigns the new subject to messages (does not affect the subject in the message itself). If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol if you want to add the new subject to the old subject. For example, entering **New Subject:&** results in **New Subject:Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Options. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Notify Application** Notifies the selected application when messages are received, and provides information from the message. Specify the application to use and the part of the message to be included. Use the Browse button to select an application, or enter the command line yourself. The command line should include the path to the executable, any options, and the following substitution variables, all separated by blank spaces: %1 Date

Exhibit K

	%2 To %3 From %4 Subject %5 Cc %6 The entire message For example, the command line to send the subject of a message to a pager might look like this: C:\apps\pager.exe -c %4 **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the list)." Eudora Windows Manual at 72-77. "One of Eudora Pro's most powerful features, however, lies buried in the section called 'Filters.' This interface may initially look slightly overpowering, but a quick read of the manual or an intuitive experiment or two will prove that this option is user-friendly and in invaluable asset.

Exhibit K

	Five 'Action Boxes' allow for the automatic management of e-mail through numerous actions and reactions to incoming and outgoing mail. Commands such as 'Forward,' 'File in Mailbox…,' 'Sound Alert,' 'Auto Reply,' 'Notify Sender,' 'Open,' and 'Print' are only a few of the possible 'filter journeys' you can assign to mail moving in either direction." Eudora Review at 5. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.  Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog. If you need to specify how the text should appear, use the **Whole word** and **Match case** options: **Whole word** If this option is on, the text is found only if it appears by itself and is not part of another word. For example, if the text is 'info' then the word 'information' will be passed over. **Match case** If this option is on, exact matches of the text are found, taking the capitalization into account. When the **Find** field is filled in and the appropriate options are set, you can use one of two functions in the dialog to find the text: **Find** and **Search**. Use the **Find** button to find instances of the text in just the

Exhibit K

current open message, or use any of the **Search** buttons to find instances of the text by searching mailboxes or mail folders."
Eudora Mac Manual at 89-90.

"Finding Text Within Messages
Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.

The Find submenu

Finding Text Within One Message
To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field.

Type the text you want to find in the text field. When finished entering the\ desired text, click the **Find** button.

Finding text

Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed.

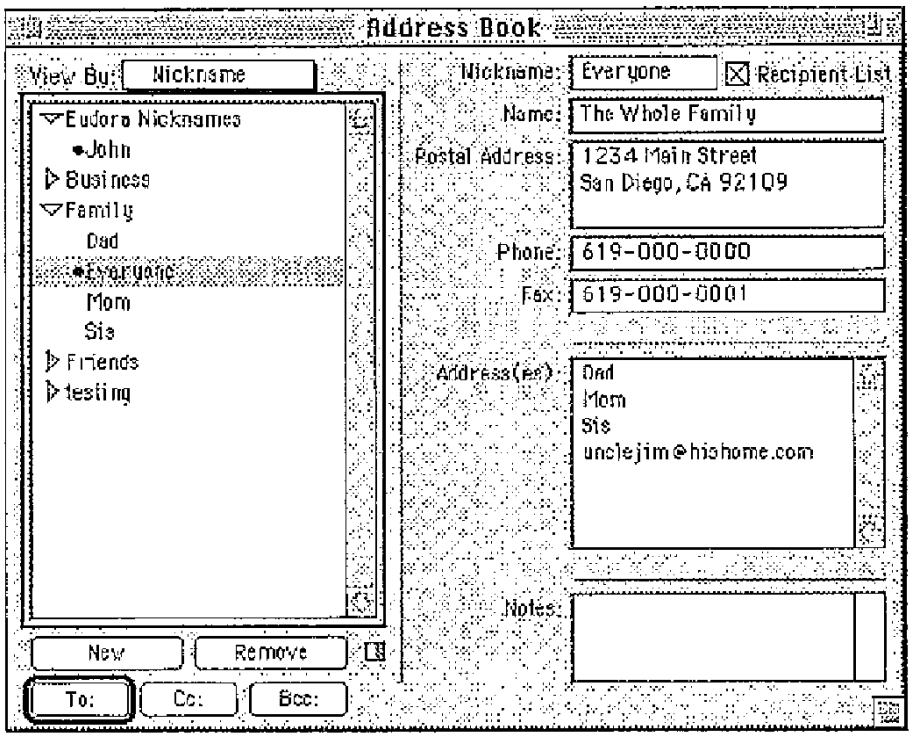
If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted.

To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only.

Exhibit K

	***** **Enter Selection Command** If you don't want to actually type the text in the Find dialog (for example, the text is very long or complex), highlight it in an existing message, and then select **Enter Selection** from the **Find** submenu. This automatically inserts the selected text at the insertion point in the Find dialog. Then, select the **Find** command from the **Find** submenu to start the search. **Stopping a Find** If you want to stop Eudora from continuing a search, click the Stop button on the progress window or press the Esc key. *Eudora Windows Manual* at 78 and 80. "Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox." Eudora Review at 5. "**Using the Address Book** The Address Book is where you keep information about individuals or groups that you correspond with. Each entry in the Address Book includes a nickname for a person or group, their full e-mail addresses, a real name, any contact information, and any notes. You can also use the Address Book to put nicknames on the Quick Recipient List, and to address a new message. To open your Address Book, select **Address Book** from the **Special** menu.

Exhibit K



The Address Book with example entries

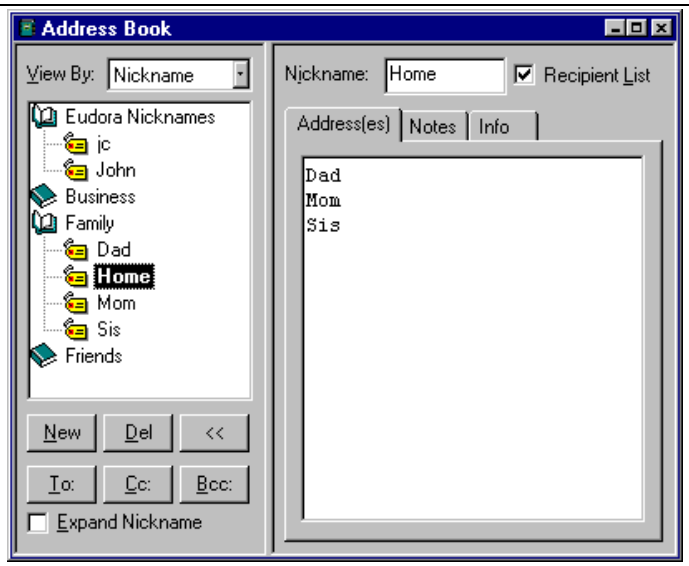
All of the Address Book entries are kept in files, so you can group your entries by putting them in different files. The example above shows files for Business, Family, and Friends (Eudora Nicknames is the default file). To show the entries in a file, click on the arrow to the left of the file. When the arrow points down, all the entries for that file are displayed." Eudora Mac Manual at 95-96.

"Using the Address Book

The Address Book is where you keep information about individuals or groups that you correspond with. Each entry in the Address Book includes a nickname for a person or group, their full e-mail addresses, a real name, any contact information, and any notes. You can also use the Address Book to put nicknames on the Quick Recipient List, and to address a new message.

To open your Address Book, select **Address Book** from the **Tools** menu."

Exhibit K



The 32-bit Address Book with example entries

All of the Address Book entries are kept in files. The example above shows files for Business, Family, and Friends (Eudora Nicknames is the default file). In the 32-bit Address Book, you can show or hide the entries in a file by double-clicking on the file. The icon shows an open or closed book, depending on whether the file is open or closed. In the 16-bit Address Book, files are flush to the left, and their entries are listed under them.

You can use the **View By** option to display the entries using any of the fields except the **Notes** field. For example, if you want to view the entries by nicknames, select **View By Nicknames**. If you view by a field that doesn't contain any data, the entry is displayed with «».

You can also start typing in the list of entries, and the appropriate entry will be selected when you enter enough unique characters to identify it.

To page up and down in the list of entries, use the arrow keys. To resize the list, drag the divider.

To close and open the right-hand side of the Address Book, use the « and » buttons."
Eudora Windows Manual at 83-84.

"An adjunct to the 'Filter' system is a more powerful 'Address Book,' which stores addressee information. E-mail addresses, street address, phone numbers and incidental notations concerning your correspondents

Exhibit K

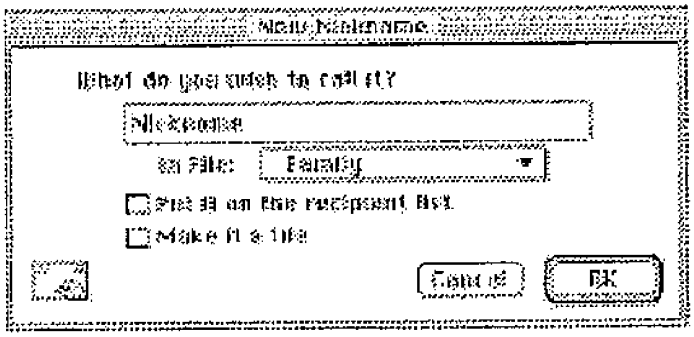
	are easily stored and accessed. You may also initiate e-mail messages directly from the Address Book, either to individuals or groups." Eudora Review at 6. "**Creating New Entries** To create a new file of Address Book entries, click on **New**. A dialog is displayed asking what you want to call it. Enter a name for the file and select **Make it a file**, then click on **OK** to create it. The file is displayed in the list, and you can now add entries to the file."  *Creating a new file or entry* To create a new entry to be included in an existing file, click on **New**. A dialog is displayed asking what you want to call it. Enter a Nickname for the entry. A Nickname (sometimes called an alias) is an easily remembered, shorter substitute for the e-mail addresses in the entry. Nicknames can be used in place of proper e-mail addresses in the **To**, **Cc**, and **Bcc** fields of outgoing messages. Specify which file this entry belongs in (if you have multiple files), and select the **Put it on the recipient list** option if you want the nickname on your recipient list. You cannot create a file within a file, so do not use the **Make it a file** option. Click on **OK** to create the entry. Then you can enter the information for that entry. In the **Address(es)** field, enter the complete e-mail addresses of the people (or person) to be included in the nickname, separating the addresses with commas or returns (this is the only place you can use a return to separate addresses). You can also use nicknames in this field, but be sure that any nicknames you use are defined in their own entry. You can use a mix of nicknames and complete e-mail addresses. *Note: Be sure there is no other information in this field except addresses or nicknames, or your messages will be addresses incorrectly.*

Exhibit K

In the **Name** field, enter the real name of the person or group. If there is just one address for the entry, the real name and the address is included in the **To** field for your recipient to see. If there is more than one address for the entry, the real name is the only thing included in the **To** field for your recipients—they do not see the whole list of recipients.

In the other fields (**Phone**, **Fax**, **Postal Address**, and **Notes**), you can enter contact information for the person or group, and any notes to yourself. This information is not included in outgoing messages.

We recommend that you have at most 2,000 nicknames per file. If the files are too large, they may appear collapsed in the Address Book, but you can still use all the nicknames when addressing messages. If you have a large number of entries, you may want to consider using a Ph server (for information, see Appendix A).

To save your changes to the Address Book, select **Save** from the **File** menu.

Changing and Removing Entries

To make changes to an Address Book entry, select the entry from the list and edit the fields as appropriate."
Eudora Mac Manual at 97-98.

"Creating New Entries

To create a new file of Address Book entries, click on **New**. A dialog is displayed asking what you want to call it.

Creating a new file or entry

Enter a name for the file and select **Make it a file**, then click on **OK** to

Exhibit K

	create it. The file is displayed in the list, and you can now add entries to the file. To create a new entry to be included in an existing file, do one of the following: click on **New** in the Address Book, select an address from anywhere in Eudora and drag it to the Address Book, or drag a message summary to the Address Book. A dialog is displayed. Enter a Nickname for the entry. A Nickname (sometimes called an alias) is an easily remembered, shorter substitute for the email addresses in the entry. Nicknames can be used in place of proper e-mail addresses in the **To**, **Cc**, and **Bcc** fields of outgoing messages. Specify which file this entry belongs in (if you have multiple files), and select the **Put it on the recipient list** option if you want the nickname on your list. You cannot create a file within a file, so do not use the **Make it a file** option. Click **OK** to create the entry. Then you can enter the information for that entry. In the **Address(es)** tab, enter the complete e-mail addresses of the people (or person) to be included in the nickname, separating the addresses with commas or returns (this is the only place you can use a return to separate addresses). You can also use nicknames in this field, but be sure that any nicknames you use are defined in their own entry. You can use a mix of nicknames and complete e-mail addresses. *Note: Be sure there is no other information in this field except addresses or nicknames, or your messages will be addressed incorrectly.* In the **Name** field (in the Info tab), enter the real name of the person or group. If there is just one address for the entry, the real name and the address are included in the **To** field for your recipient to see. If there is more than one address for the entry, the real name is the only thing included in the **To** field for your recipients— they do not see the whole list of recipients. If there is nothing in the Name field, the recipients do see the whole list. In the other fields provided in the **Info** (**Phone**, **Fax**, and **Postal Address**) and **Notes** tabs, you can enter contact information for the person or group, and any notes to yourself. This information is not included in outgoing messages. It is recommended that you have at most 2,500 entries per file. If you have a large number of entries, you may want to consider using a Ph server (for information, see Appendix A).

Exhibit K

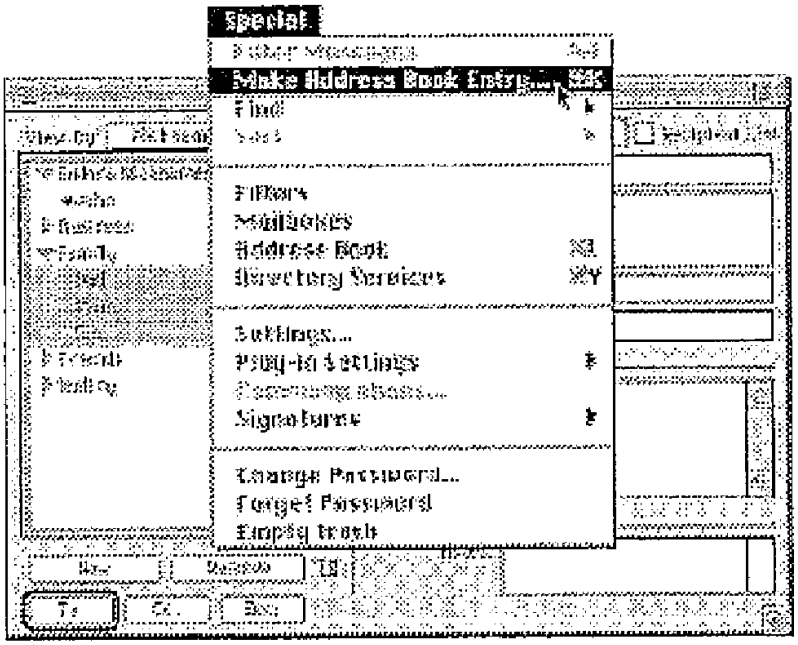
	To save your changes to the Address Book, select **Save** from the **File** menu. **Changing, Moving, Copying, and Deleting Entries** To make changes to an entry (including changing the name of the entry, the nickname for it, and any information in the Address, Notes, or Info tabs), select the entry from the list and edit the fields as appropriate. ***Important:*** *If you change a nickname, but sure to correct any entries that reference that nickname.* To move or copy an entry to a file, right-click on it and select the **Move To** or **Copy To** command. The **Choose a Nickname File** dialog is displayed so that you can select the file you want to move the entry to. In the 32-bit version, you can move an entry (or entries) to a different file by dragging it, or copy it by holding down the Shift or Ctrl key then dragging it. *Note: You cannot move an entry into the file it is already in, but you can* copy *an entry into its file (a "Copy of Entry" is created).* To delete an entry or an address file, select it from the list and click on the **Del** button or the **Delete** key. You cannot remove the Eudora Nicknames file. *Note: If a nickname file is set to read-only (you do not have permission to write to it), you cannot move or copy entries into it., or delete an entry from it.* To save your changes, select **Save** from the **File** menu." *Eudora Windows Manual* at 84-86. "**The 'Make Address Book Entry' Command** The **Make Address Book Entry…** command is used to create entries in your Address Book, and is especially helpful for making group entries. You can use this command from anywhere in Eudora, including the Address Book, mailboxes, open messages, and the Directory Services window. From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of

Exhibit K

the new entry. The new entry's **Address(es)** field will include all of the addresses that you selected.

Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.

In the Address Book, highlight several different entries (hold down the shift key to select multiple entries in sequence, or the command key to make disjoint selections), the select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message(s) you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message

	and are all put in the new entry. *Note: The **Make Address Book Entry…** command uses the Replying Settings. If the **Reply to all by Default** setting is turned on (or you hold down the option key), the new entry will include all of the recipients of the messages plus the sender. Or, if the **Include yourself** setting is turned off, your address is not included in the new entry.* In an open message window, select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. See the note above about the Replying Settings. In the Directory Services window, finish a Ph query and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry. **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings. **Using Nicknames that were Not Created by Eudora** To use a nickname file that was not created in Eudora, put the file in the Nicknames Folder in your Eudora Folder (located in your System Folder), and be sure the format is correct: One nickname on each line with the word 'alias,' a space, the nickname, a space, and the real addresses separated by commas. For example alias joe joe@wow.com alias group joe@wow.com,lisa@wow.com,bill@wow.com

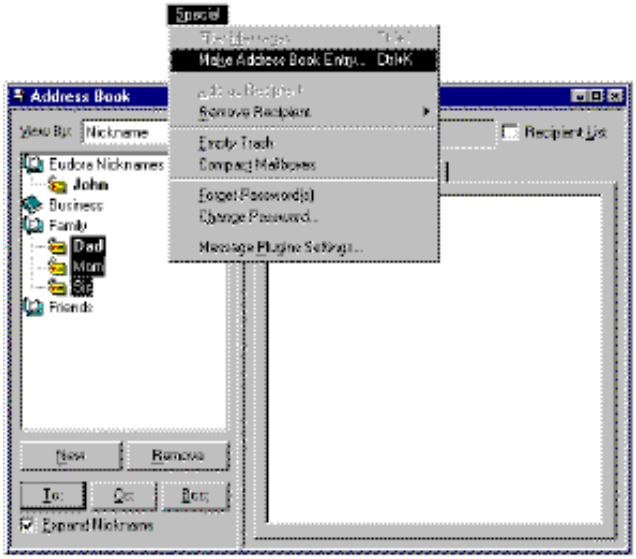
Exhibit K

You will need to quit and restart Eudora to see your new entries in the Address Book."
Eudora Mac Manual at 99-101.

"The "Make Address Book Entry" Command

The Make Address Book entry command is used to create entries in your Address Book, and is especially helpful for making group entries.

In the Address Book, highlight several different entries (hold down the Shift key to select multiple entries in sequence, or the Ctrl key to make disjoint selections), then select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message summaries you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry.

	*Note: The **Make Address Book Entry** command uses the Reply Options. If the **Include yourself** option is on, your address is included in the new entry.* In the Directory Services window, finish a Ph query, select the items that you want to include in the entry (or do not select anything to use all of the items), and select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry. **The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options. **Using Central Address Book Files on a Server** You can set up central Address Book files on a server and configure Eudora clients so that they refer to the central files. First, be sure the files are plain text, have a .**txt** extension, and are formatted as follows: One nickname on each line with the real addresses separated by commas, and one line for notes and info with the **Notes** text following the **Info** data. For example: alias Wow joe@wow.com,lisa@wow.com,chris@wow.com note Wow <fax: 222.2223><phone: 222.2222><address:1234 Street> <name:Wow Inc.>My favorite company Then, for each client application, add a **ExtraNicknameDirs** entry to the [Settings] section of the EUDORA.INI file. This entry should be followed by the list of directories that contain Address Book files, separated by semicolons (;). Any Address Book files located in those directories are added to the Address Book. Users will need to exit and

Exhibit K

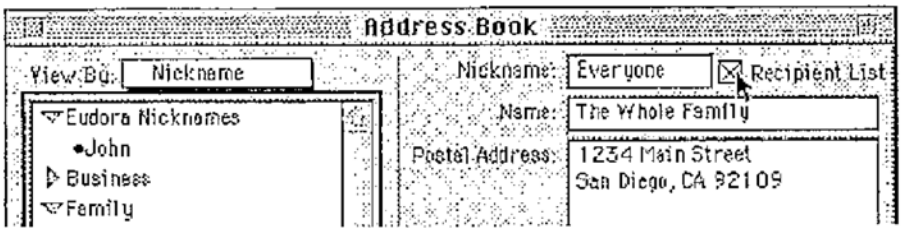
re-open Eudora to see the new entries.

Using Address Book Files Not Created by Eudora

To use a Address Book file that was not created in Eudora, put the file in the Nickname directory (in your Eudora directory), and be sure the format is as shown in the section "Using Central Address Book Files on a Server." You will need to exit and reopen Eudora to see your new entries in the Address Book."
Eudora Windows Manual at 87-90.

"**Using the Quick Recipient List**

The Quick Recipient List is your list of often-used nicknames. If you have checked the **Recipient List** option in an Address Book entry, the entry's nickname is included in the list.



The Recipient List option

To open a new message address to someone on your Quick Recipient List, select **New Message To, Forward to**, or **Redirect To** from the **Message** menu, and select the nickname from the displayed list.

To insert a nickname into a message that you have already opened, put the cursor where you want the nickname and select **Insert Recipient** from the **Edit** menu.

To insert the real address(es), instead of the nickname, hold down the option key and select **Insert & Expand Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail Settings."
Eudora Mac Manual at 102.

"Using the Quick Recipient List

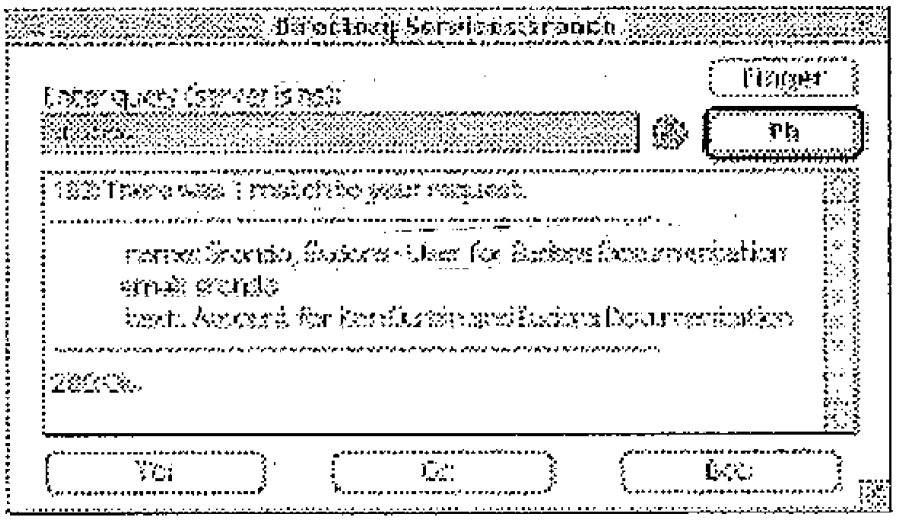
The Quick Recipient List is your list of recipients to whom you often send mail.

Exhibit K

	To add a nickname to the Quick Recipient List, check the **Recipient List** option in its Address Book entry. To add an e-mail address to the Quick Recipient list, select the text that makes up the full address. Then, select **Add As Recipient** from the **Special** menu. To remove an entry from the list, uncheck the Recipient List option in the Address Book entry, or select the nickname from the **Remove Recipient** submenu from the **Special** menu. To open a new message addressed to someone on your Quick Recipient List, select **New Message To, Forward To**, or **Redirect To** from the **Message** menu, and select the nickname from the displayed list. To insert a recipient into a message that you have already opened, put the cursor where you want the recipient and select **Insert Recipient** from the **Edit** menu. To insert the real address(es), instead of a nickname, hold down the Shift key and select **Insert Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options. More than one recipient from the Quick Recipient List can be added to the **To**, **Cc**, and **Bcc** fields of any message. If you use the Insert Recipient command, commas are added where necessary." *Eudora Windows Manual* at 90. "**Using Directory Services** Opening Directory Services Eudora can access two different online directory services, **Ph** and **Finger**. To use these services, you must put the name of the host machines for the Ph and Finger servers in the Hosts Settings. To use the directory Services, select **Directory Services** from the **Special** menu. The active Ph or Finger server (defined in your Hosts Settings) is displayed above the query field. Or, you can select a string of text (someone's name or e-mail address, for example), hold down the shift key, and select **Directory Services** from the **Special** menu. This opens the window and inserts the string of text into the query field. Using Ph

Exhibit K

To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.



A Ph query and its response

Note: You can type any Ph command in the query field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

If the **'Live' Ph queries** option is on in your Hosts Settings, the connection with your server is kept open and Ph queries are automatically sent to the server when you finish typing the query text.

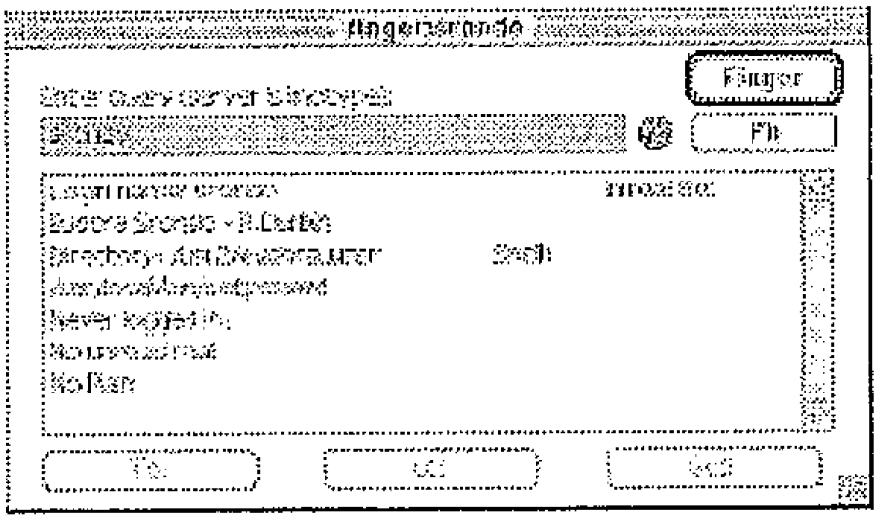
*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

Finding Ph Servers

Some Ph servers keep a list of other Ph servers that are available on the Internet. This is not always a comprehensive list of every Ph server out there, but it can be helpful.

To get the list of servers from the active server (the active server is displayed above the query field), click on the globe in the Directory Services window (new to the Ph button). A list of servers is displayed in the results area. To go to one of those servers and do a query, hold down

Exhibit K

	the command key and click on the server's URL, or just double-click on the URL. Using Finger To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."  *A Finger query and its response* Addressing a Message from the Directory Services Window You can create and address a message with the query results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-05.

"**Using Directory Services**

Opening Directory Services

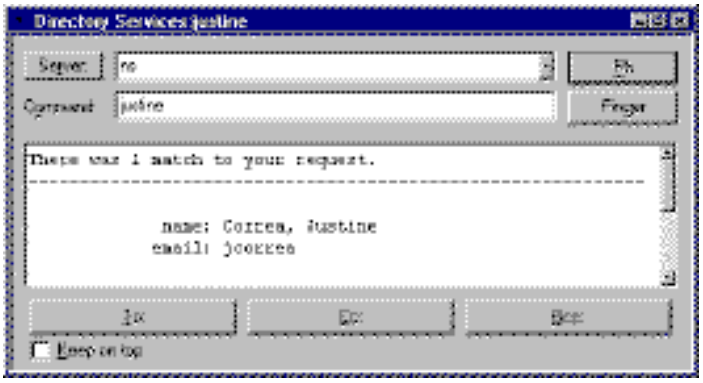
Eudora can access two different online directory services, **Ph** and **Finger**. To use these services, you must put the name of the host machines for the Ph and Finger servers in the Hosts Options. The active Ph or finger server (defined in your Host Options) is displayed in the Server field, which includes a popup list of the last 10 servers you have accessed.

To use the directory Services, select **Directory Services** from the **Tools** menu.

Using Ph

To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window.

If there is more than one match, you can use the Tab key to move down to the next one, or hold down the Shift key and press Tab to move up to the previous one.



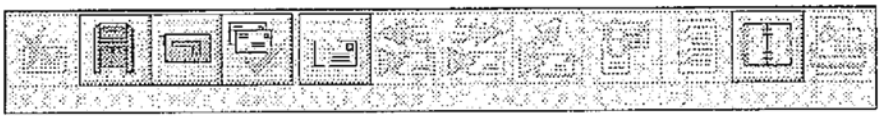
A Ph command and its response

Note: You can type any Ph command in the command field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

Exhibit K

	Finding Ph Servers Some Ph servers keep a list of other Ph servers that are available on the Internet. This is not always a comprehensive list of every Ph server out there, but it can be helpful. To get the list of servers that the active server knows about, click on the Server button in the Directory Services window. A list of servers is displayed in the results area. To go to one of those servers and do a query, double-click on the server's URL. **Using Finger** To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window. **Addressing a Message from the Directory Services Window** You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." *Eudora Windows Manual* at 91-92. "**Reference** Customizing the Main Toolbar The Main Toolbar is a group of buttons giving you easy access to your frequently used Eudora functions.

Exhibit K

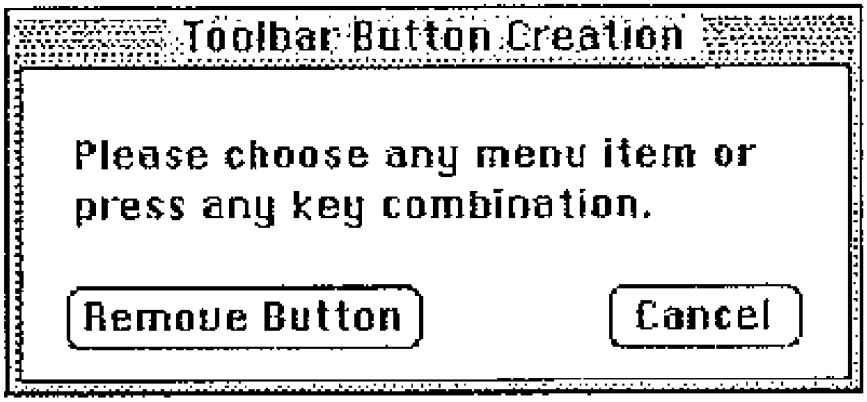


The Main Toolbar

You can set the buttons to correspond to your function keys (the **F** keys on an extended keyboard), and you can create new buttons for most Eudora commands.

Buttons on the main toolbar are context-sensitive: they are highlighted if they can be used for the current task, and are dimmed if they cannot. For example: if an incoming message is currently open or selected in a mailbox, the Reply toolbar button is highlighted because you can reply to the message. If no incoming message is open or selected, the Reply button is dimmed because there's no message to reply to. For toolbar buttons that correspond to menu commands, the state of the button—highlighted or dimmed—matches the state of its associated menu command.

To add a new button to the main toolbar, hold down the command key and put the cursor between two buttons or at either end of the toolbar, depending on where you want the new button to go; when the arrow changes to a splitter cursor [], click on it. The **Toolbar Button Creation** dialog is displayed prompting you to choose a menu item or enter a key combination. This can be almost anything you would do with Eudora, including using modifier keys with the command. When you are done, the button is added to the toolbar, and named appropriately. If you change your mind, click **Remove Button** or **Cancel**.



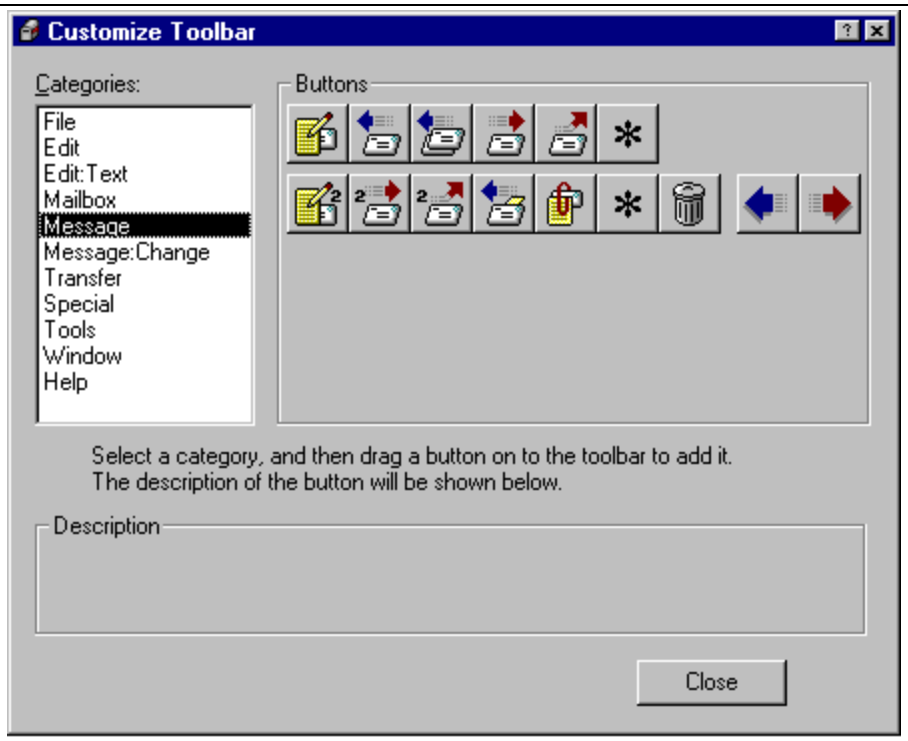
The Toolbar Button Creation dialog

You can also drag files from the Finder to the toolbar to add them as

Exhibit K

	buttons, and you can drag mailboxes from the Mailboxes window to the toolbar to add them as mailbox buttons. (You cannot drag mail folders to the toolbar.) To change what a button does, hold down the command key and click on the button, then select the function as you would when creating a button, or click **Cancel** if you change your mind." Eudora Mac Manual at 107-08. **Reference** The Main Toolbar is a group of buttons that give you easy access to your frequently used Eudora commands.  *The Main Toolbar* The 32-bit toolbar can be moved to wherever you want it on the screen. Just hold down the left mouse button on the bar itself (not on a button) and drag it around until you find a place you like. You can dock it to the top or bottom of the Eudora window or put it anywhere on your desktop. To add buttons to the 32-bit toolbar, right click on the toolbar, and select **Customize**. The **Customize Toolbar** dialog is displayed.

Exhibit K



The Customize Toolbar dialog

Select a Eudora menu from the list on the left, then drag buttons from the list on the right down to the toolbar.

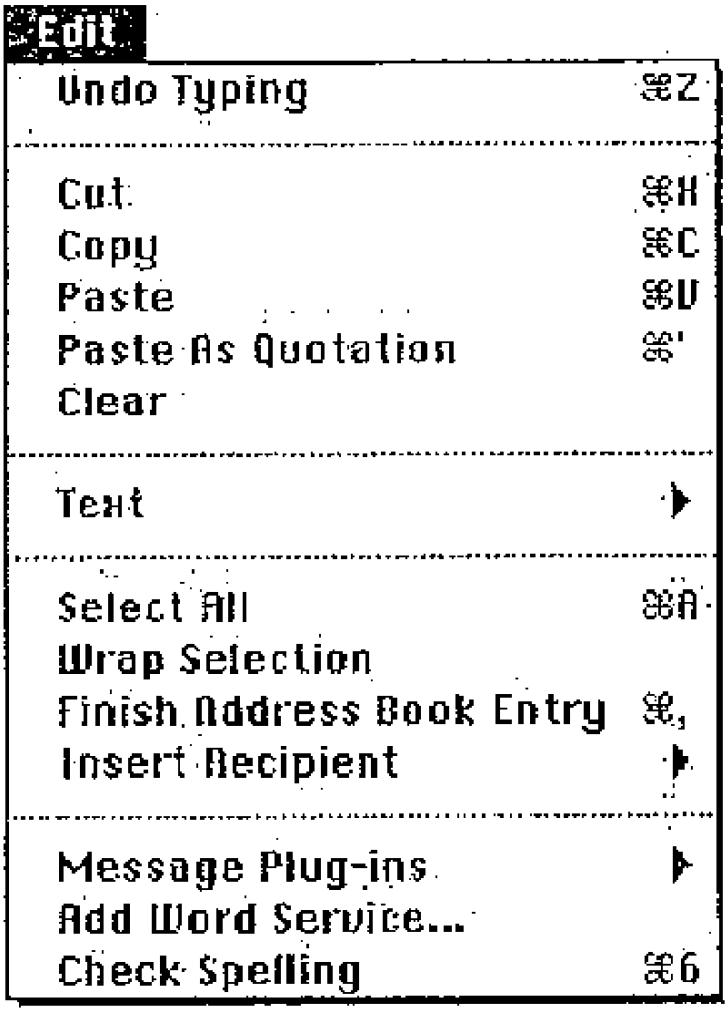
To change the placement of a button on the toolbar, hold down the **Alt** key and drag the button to where you want it. To remove a button, hold down the Alt key and drag it off of the toolbar.

The 16-bit toolbar cannot be moved or changed."
Eudora Windows Manual at 93-94.

"**Edit**

This menu provides text editing tools.

Exhibit K



* * *

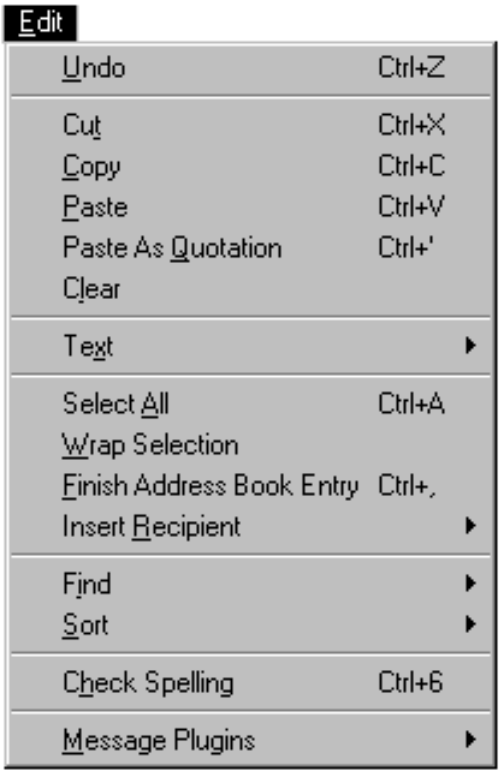
Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-50.

"Edit

This menu provides text editing tools.

Exhibit K



* * *

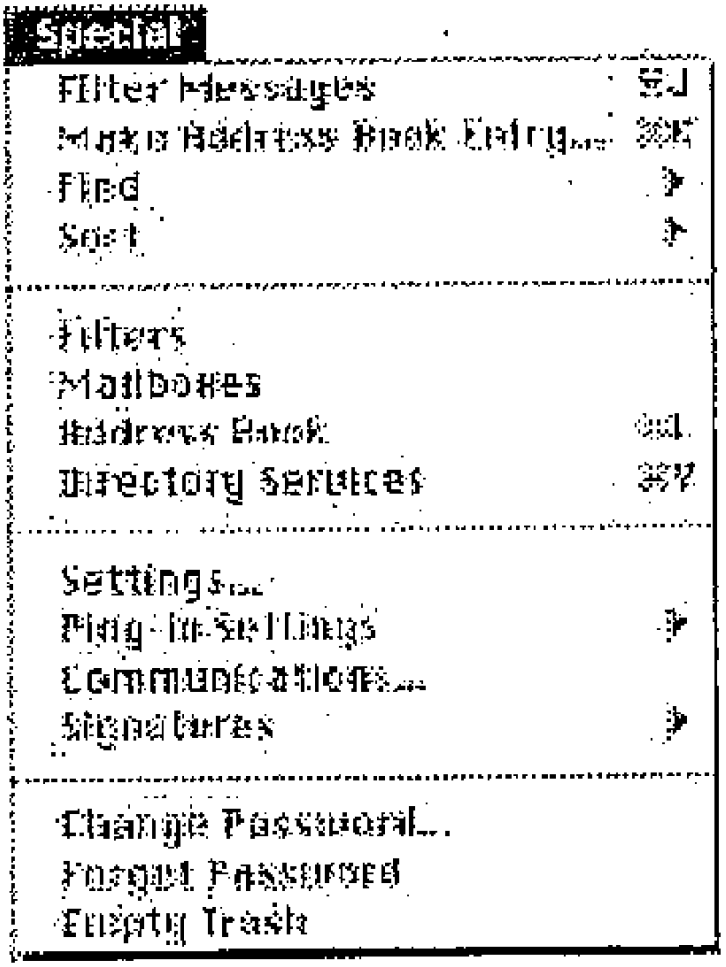
Finish Address Book Entry
Complete the partial text of a nickname.

Insert Recipient
Insert the chosen recipient"
Eudora Windows Manual at 137.

"**Special**

This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).

Make Address Book Entry…
[shift] Make Address Book Entry From Selection…
Create an Address Book entry (nickname) from the current message.
Create an entry from the selected addresses.

Find
Search for the designated character string within a message, messages, mailboxes, or mail folders.

Sort [option] Sort Descending
Sort the message summaries in a mailbox in ascending order.
Sort them in descending order.

Filters

Exhibit K

Display the Filters window.

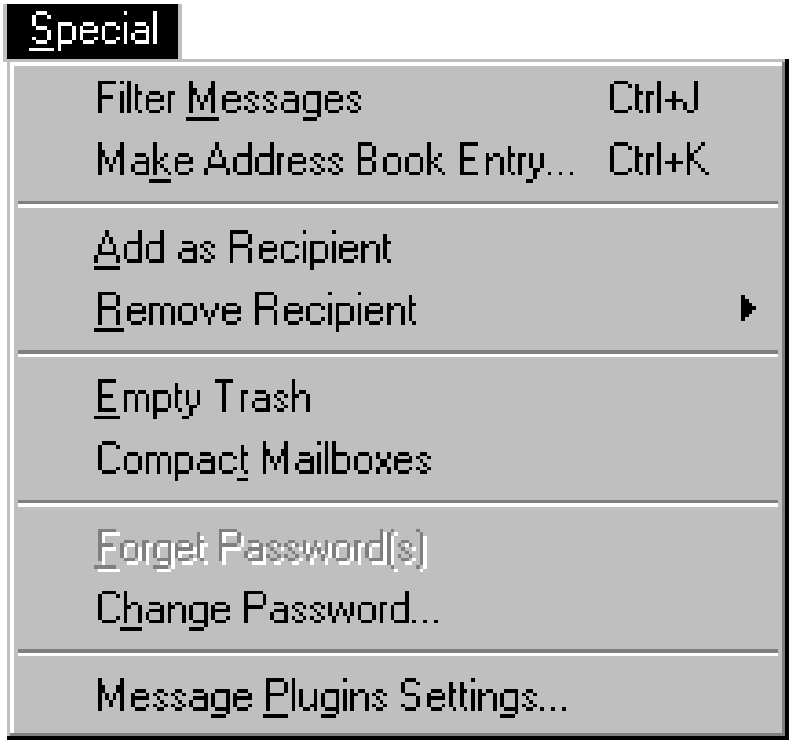
Mailboxes
Display the Mailboxes window.

Address Book
Display the Address Book window.

Directory Services
Display the Directory Services window.
Eudora Mac Manual at 155.

"Special

This menu lets you use additional Eudora functions.



Filter Messages
Run the manual filters for the current message(s).

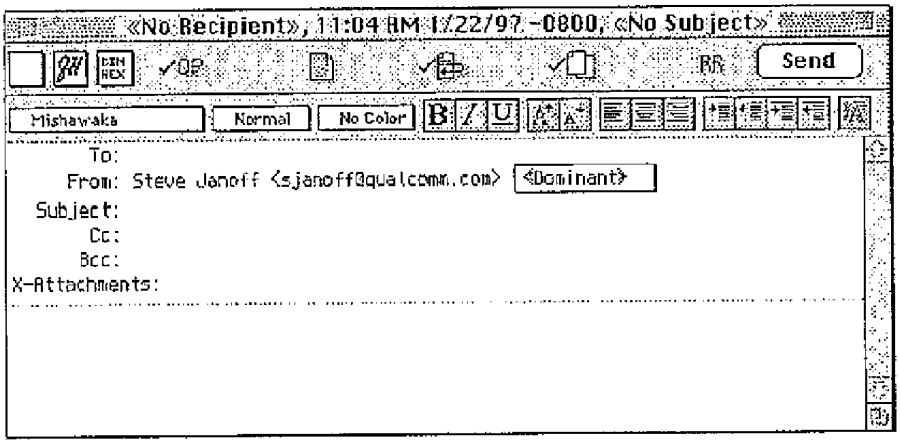
Make Address Book Entry…
Create an Address Book entry from the current message.

Add As Recipient
Add selected text to the Quick Recipient list.

Exhibit K

	Remove Recipient Select a recipient from this menu and the recipient is removed from the Quick Recipient list. **Empty Trash** Delete all messages from the Trash mailbox. **Compact Mailboxes** Reclaim unused space in all mailboxes. **Forget Password(s)** Make Eudora forget your passwords so mail can't be checked. **Change Password...** Change a POP account password on the POP server. **Message Plugins Settings...** Open the Message Plugins Settings." *Eudora Windows Manual* at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 1, 9, and 18.
displaying the document electronically using the first computer program;	Eudora discloses this element. See, e.g.: Disclosures for Claim 1. "**Using the Composition Window** The composition window consists of the title bar, the icon bar, the formatting toolbar, the message header, and the message body. The formatting toolbar can be turned on or off."

Exhibit K



The composition window

Eudora Mac Manual at 23.

"Using the Composition Window

The composition window consists of the title bar, the Toolbar, the message header, and the message body."

The composition window

Eudora Windows Manual at 19.

"**Message Header**

Exhibit K

	Outgoing mail headers consist of six fields: **To, From, Subject, Cc, Bcc,** and **X-Attachments**. Each field is described below. The **To, Subject, Cc,** and **Bcc** fields can be directly edited. To move the cursor from field to field, press the tab key or click in the desired field with the mouse. ****** **Message body** After filling in the header fields, move the insertion point to the space below the message header. Type the body of the message here. For information about formatting your message text, see the sections 'Formatting Text' and 'Formatting Toolbar.' " Eudora Mac Manual at 27-28. **"Message Header** Outgoing mail headers consist of six fields: **To, From, Subject, Cc, Bc** and **Attachments**. Each field is described below. The **To, Subject, Cc,** and **Bcc** fields can be directly edited. To move the cursor from field to field, press the tab key or click in the desired field with the mouse. ****** **Message body** After filling in the header fields, move the insertion point to the space below the message header. Type the body of the message here. For information about formatting your message text, see the section 'Formatting Text' " *Eudora Windows Manual.* at 23-24. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 1.
while the document is being displayed, analyzing, in a computer process, first information from the document to determine if the first information is at least one of a plurality of types of information that can be searched for in order to find second information related to the first information;	Eudora discloses this element. See, e.g.: Disclosure to Claim 1. "To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight

Exhibit K

	a word or a block of text to check only that text and not the rest of the message. Eudora Mac Manual at 43. **"Checking Your Spelling** Eudora includes a built-in spelling checker. It can be used to check for misspellings in the body of current message composition windows, text files, and signature files. It includes a built-in dictionary and also allows for the creation of a custom user dictionary. Additionally, it can be configured to ignore capitalized words, words with numbers, and mixed case words, to report mixed case and double (repeated) words, and to suggest alternative spellings. *Eudora Windows Manual* at 33-37. "**Using Active URLs** Any string of text that Eudora recognizes as a 'hot link,' also known as a URL (Uniform Resource Locator: http, ftp, gopher, ph, finger, etc.), is active. Active URLs are normally highlighted in blue text and underlined, and are often enclosed by less-than and greater-than signs, '<>.' You can hold down the command key and click on a URL (or just double-click on it) to open a World Wide Web location, transfer a file, do a gopher search, use the finger tool, etc. Eudora Mac Manual at 64. **"Using Active URLs** Any string of text that Eudora recognizes as a URL (Uniform Resource Locator: http, ftp, gopher, ph, finger, etc.) is active. Double-click on a URL to open a World Wide Web location, transfer a file, do a gopher search, use the finger tool, etc. URLs are highlighted and underlined to show that they are active (32-bit Eudora only). *Eudora Windows Manual* at 50. "Speaking of color, a real bonus of Eudora Pro 3.0 is its ability to acce 'Embedded URLs' in the body of a message. Any URL you enter nto th body of your mail text (designated by ,. Brackets) is automatically colored blue and underlined." Eudora Review at 3. "[I]ncoming e-mail correspondence containing URLs written in the

Exhibit K

	standard 'http://....' format are highlighted for your use, no matter what application the sender uses to generate the message." Eudora Review at 4. "**Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered. ****** Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself (this is helpful if you want to search for a header item that does not appear on the menu, such as X-Priority). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ● «Body» ● «Personality» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **BLAH BLAH BLAH BLAH** option); the «Body» option searches the message body; and the «Personality» option searches the name of the personality associated with the message. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are:

Exhibit K

contains or **does not contain**
If the specified header item contains or does not contain the text string, filter the message.

is or **is not**
If the specified header item is or is not a complete match of the text string, filter the message.

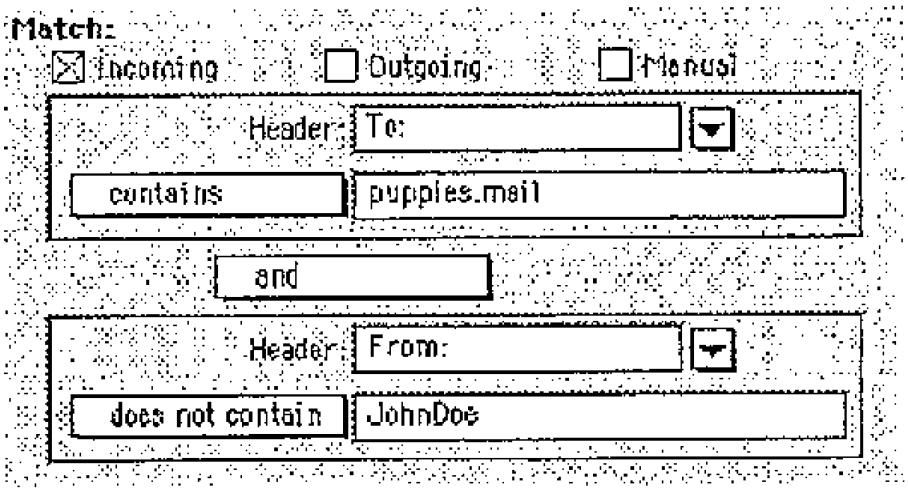
starts with or **ends with**
If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored.

appears or **does not appears**
If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain.

intersects nickname
If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message.

Use the **Text** fields to specify the text strings that the filter is searching for.

******



Sample Match Area in Filters window

Use the **Conjunction** popup to link the two terms. The conjunction

Exhibit K

	options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches the first term, filter it *unless* the message also matches the second term, in which case *do not* filter it. (This lets you exclude certain variations of the first term.) Eudora Mac Manual at 85. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered. ****** **Filter Criteria (the Match Area)** Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup. Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or

Exhibit K

	enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows: • To • Cc • From • Subject • Reply-To • «Any Recipient» • «Any Header» • «Body» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **Blah Blah Blah Blah** option, and the «Body» option searches the message body. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are: **contains** or **doesn't contain** If the specified header item contains or does not contain the text string, filter the message. **is** or **is not** If the specified header item is or is not a complete match of the text string, filter the message. **starts with** or **ends with** If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored. **appears** or **doesn't appear** If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain. **intersects nickname** If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message. Use the **Text** fields to specify the text strings that the filter is searching for.

Exhibit K

	Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match. Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches both the first and second terms, do not filter it. (This lets you exclude certain variations of the first term.) *Eudora Windows Manual* at 72-75. "In a mailbox, highlight the message(s) you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry. ****** In an open message window, select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. See the note above about the Replying Settings. ******

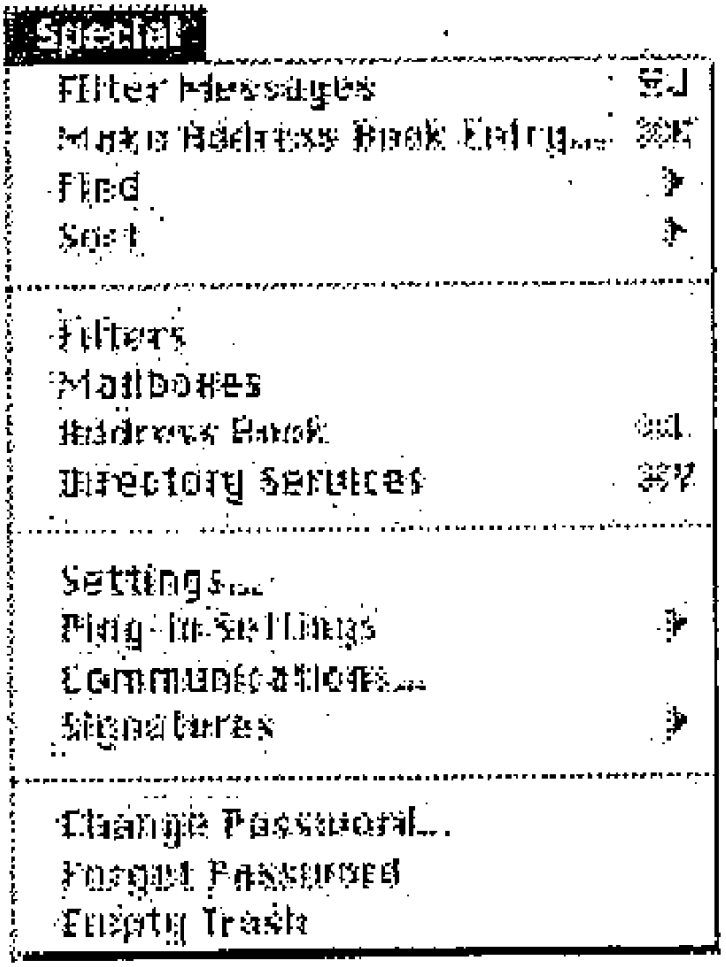
Exhibit K

	The 'Finish Address Book Entry' Command With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings." Eudora Mac Manual at 99-101. **"The "Make Address Book Entry" Command** ****** In a mailbox, highlight the message summaries you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry. ****** **The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the

Exhibit K

	Edit menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options." *Eudora Windows Manual* at 88-89. "**Edit** ****** **Finish Address Book Entry** **[option] Finish & Expand Address Book Entry** Complete the partial text of a nickname. Complete it and expand it to its real address. **Insert Recipient** **[option] Insert & Expand Recipient** Insert the chosen nickname. Insert the real address of the nickname." Eudora Mac Manual at 149-50. "**Special** This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).

Make Address Book Entry…
[shift] Make Address Book Entry From Selection…
Create an Address Book entry (nickname) from the current message.
Create an entry from the selected addresses.
Eudora Mac Manual at 155.

"Special

This menu lets you use additional Eudora functions.

Exhibit K

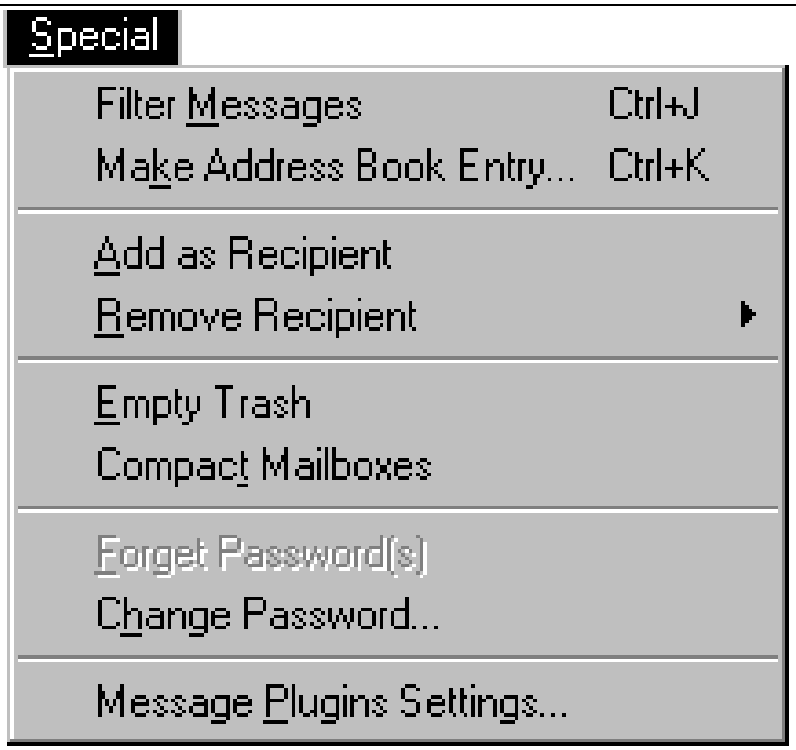
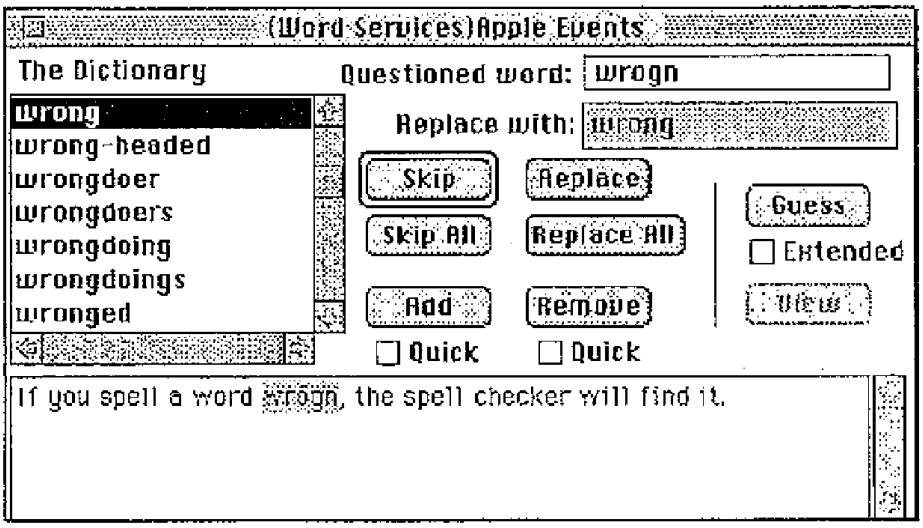
	 **Filter Messages** Run the manual filters for the current message(s). **Make Address Book Entry…** Create an Address Book entry from the current message. *Eudora Windows Manual* at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 11, 14, and 15.
retrieving the first information;	Eudora discloses this element. See, e.g.: Disclosure to Claim 1. "**Checking Your Spelling** ****** To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight

Exhibit K

a word or a block of text to check only that text and not the rest of the message.

If no misspellings are found, the spelling checker quits.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.



The Check Spelling dialog

******

Replace (All)
Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word.

******

Add
Add the questioned word to the dictionary. If the **Quick** option is on, then the questioned word is added to the dictionary immediately when you click this button. If this option is off, the **Adding word to Dictionary** dialog is displayed. This dialog provides you with options for adding the word and its various forms to the dictionary.
Eudora Mac Manual at 42-45.

"Checking Your Spelling

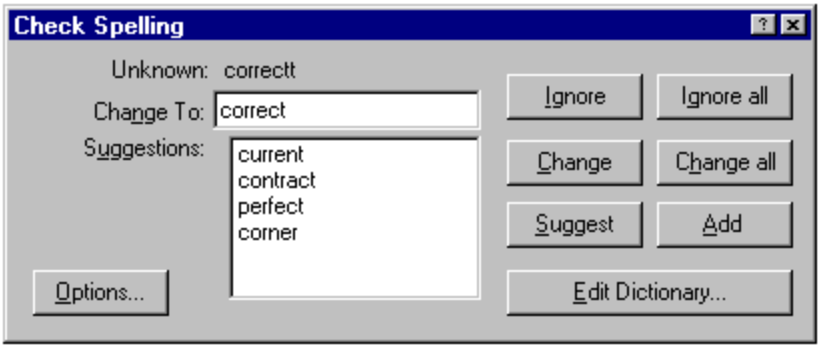
Exhibit K

******

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

******

Change all Button
This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word.

******

Add Button
This button adds the unknown word to your custom user dictionary. *Eudora Windows Manual* at 33-36.

"**Filtering Messages**

******

When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered.

Exhibit K

	****** Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself (this is helpful if you want to search for a header item that does not appear on the menu, such as X-Priority). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ● «Body» ● «Personality» ****** Filter Actions All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). Eudora Mac Manual at 82-85. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered.

Exhibit K

	****** Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ****** **Filter Actions** All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). ****** **Notify Application** Notifies the selected application when messages are received, and provides information from the message. Specify the application to use and the part of the message to be included. *Eudora Windows Manual* at 72-76. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.

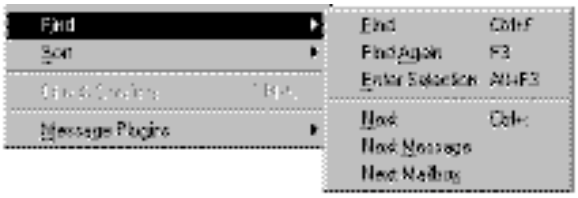
Exhibit K

Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog.

******

When the **Find** field is filled in and the appropriate options are set, you can use one of two functions in the dialog to find the text: **Find** and **Search**. Use the **Find** button to find instances of the text in just the current open message, or use any of the **Search** buttons to find instances of the text by searching mailboxes or mail folders."
Eudora Mac Manual at 89-90.

"Finding Text Within Messages
Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.



The Find submenu

Finding Text Within One Message
To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field.

Type the text you want to find in the text field. When finished entering the\ desired text, click the **Find** button.

Exhibit K

Finding text

Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed.

If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted.

To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only.

*****

Enter Selection Command
If you don't want to actually type the text in the Find dialog (for example, the text is very long or complex), highlight it in an existing message, and then select **Enter Selection** from the **Find** submenu. This automatically inserts the selected text at the insertion point in the Find dialog. Then, select the **Find** command from the **Find** submenu to start the search.
Eudora Windows Manual at 78 and 80.

"Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox."
Eudora Review at 5.

"**The 'Make Address Book Entry' Command**

******

From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of

Exhibit K

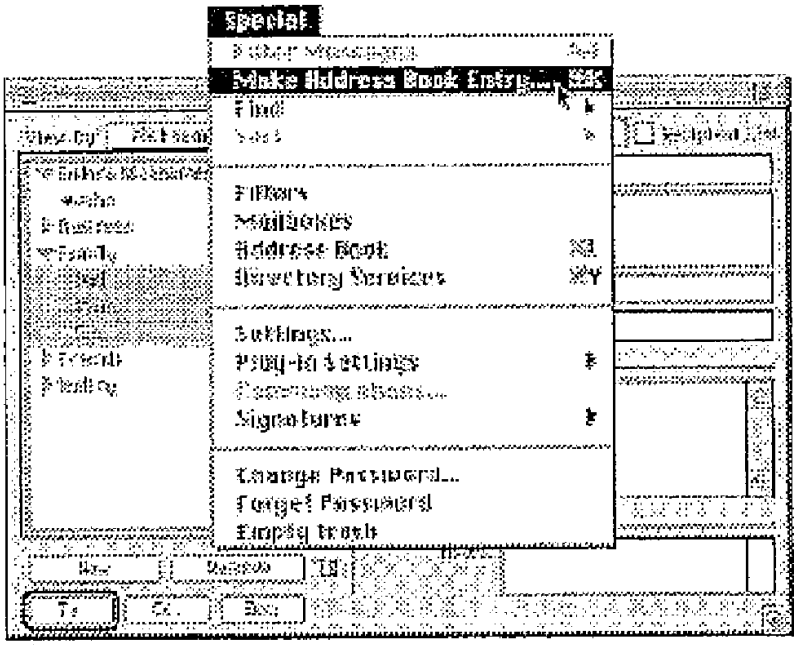
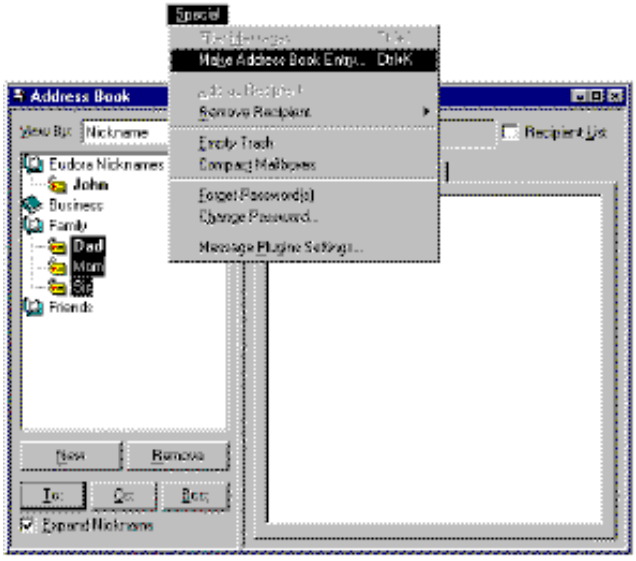
	the new entry. The new entry's **Address(es)** field will include all of the addresses that you selected. ****** In the Address Book, highlight several different entries (hold down the shift key to select multiple entries in sequence, or the command key to make disjoint selections), the select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.  *Using the "Make Address Book Entry" command from the Address Book* In a mailbox, highlight the message(s) you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry. ******

Exhibit K

	In an open message window, select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. See the note above about the Replying Settings. In the Directory Services window, finish a Ph query and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry. **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. Eudora Mac Manual at 99-101. **"The "Make Address Book Entry" Command** The Make Address Book entry command is used to create entries in your Address Book, and is especially helpful for making group entries. In the Address Book, highlight several different entries (hold down the Shift key to select multiple entries in sequence, or the Ctrl key to make disjoint selections), then select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.

Exhibit K



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message summaries you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry.

******

In the Directory Services window, finish a Ph query, select the items that you want to include in the entry (or do not select anything to use all of the items), and select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry.

The "Finish Address Book Entry" Command

With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would

Exhibit K

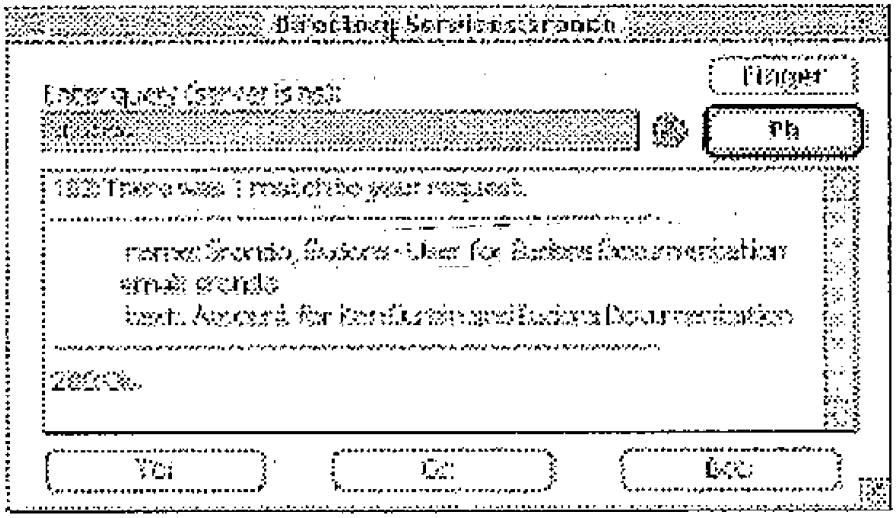
have to enter "jon" or "joh" for Eudora to complete them.
Eudora Windows Manual at 87-89.

"**Using Directory Services**

To use the directory Services, select **Directory Services** from the **Special** menu. The active Ph or Finger server (defined in your Hosts Settings) is displayed above the query field. Or, you can select a string of text (someone's name or e-mail address, for example), hold down the shift key, and select **Directory Services** from the **Special** menu. This opens the window and inserts the string of text into the query field.

Using Ph

To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.



A Ph query and its response

******

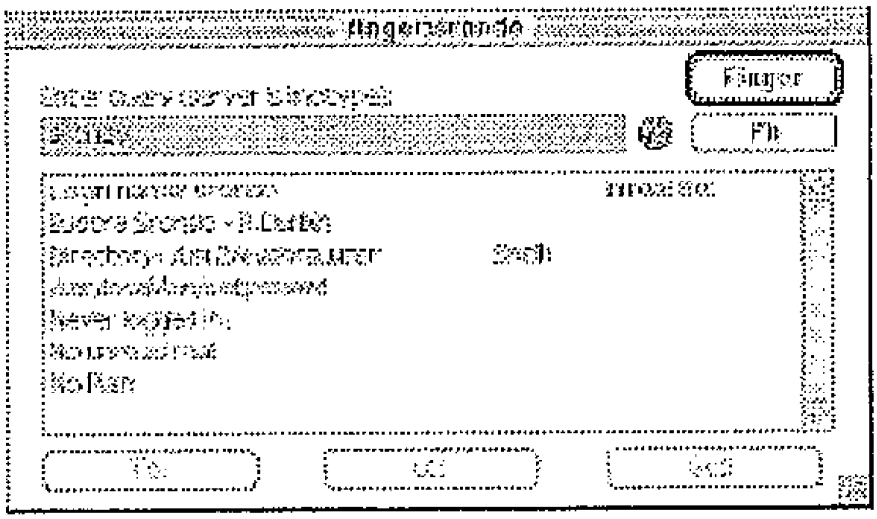
*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

******

Using Finger

Exhibit K

To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."



A Finger query and its response

Addressing a Message from the Directory Services Window

You can create and address a message with the query results in the Directory Services window.

To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results.

To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-105.

Exhibit K

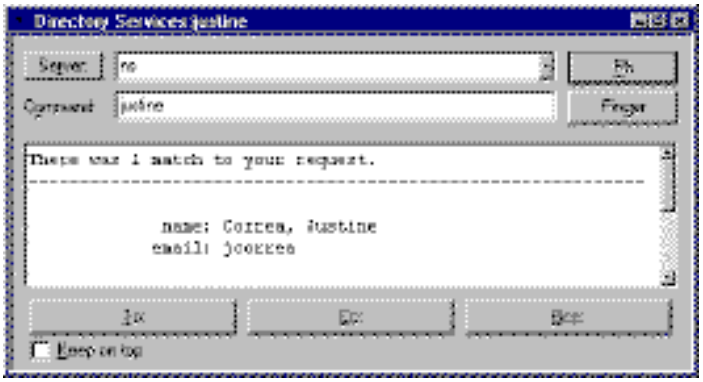
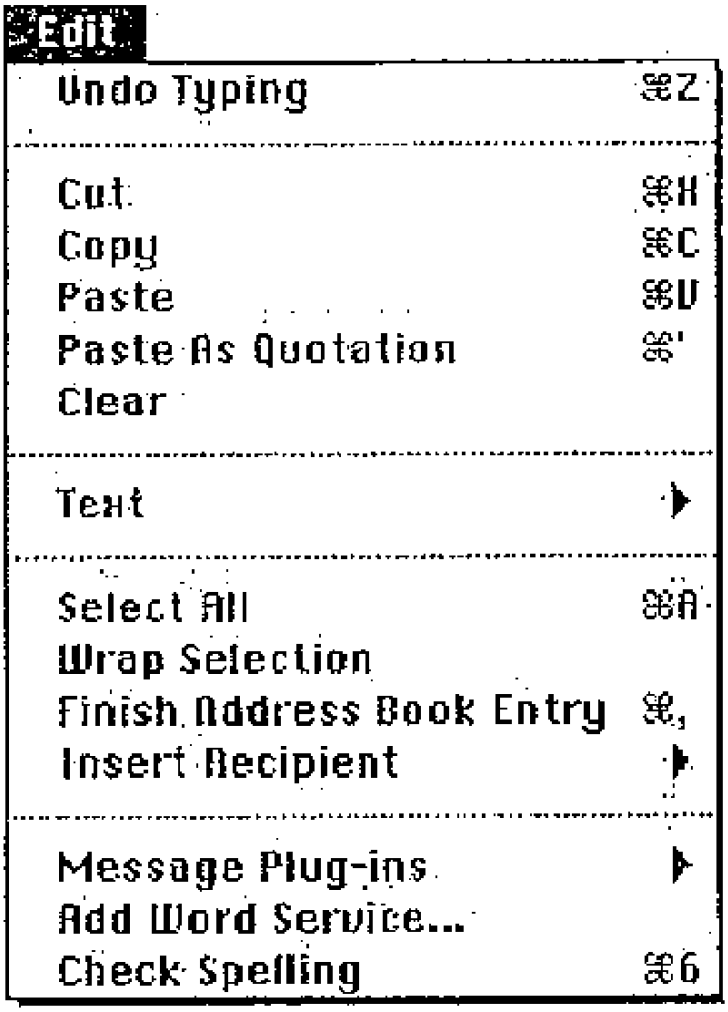
	"**Using Directory Services** **Opening Directory Services** Eudora can access two different online directory services, **Ph** and **Finger**. To use these services, you must put the name of the host machines for the Ph and Finger servers in the Hosts Options. The active Ph or finger server (defined in your Host Options) is displayed in the Server field, which includes a popup list of the last 10 servers you have accessed. ****** **Using Ph** To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window. ******  *A Ph command and its response* ****** **Using Finger** To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window.

Exhibit K

	Addressing a Message from the Directory Services Window You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." Eudora Windows Manual at 91-92. "**Edit** This menu provides text editing tools.

Exhibit K



******

Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-150.

"Edit

Exhibit K

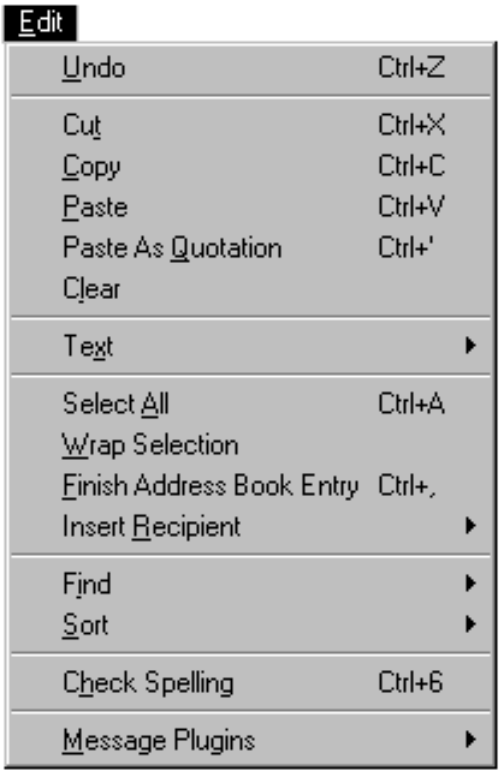
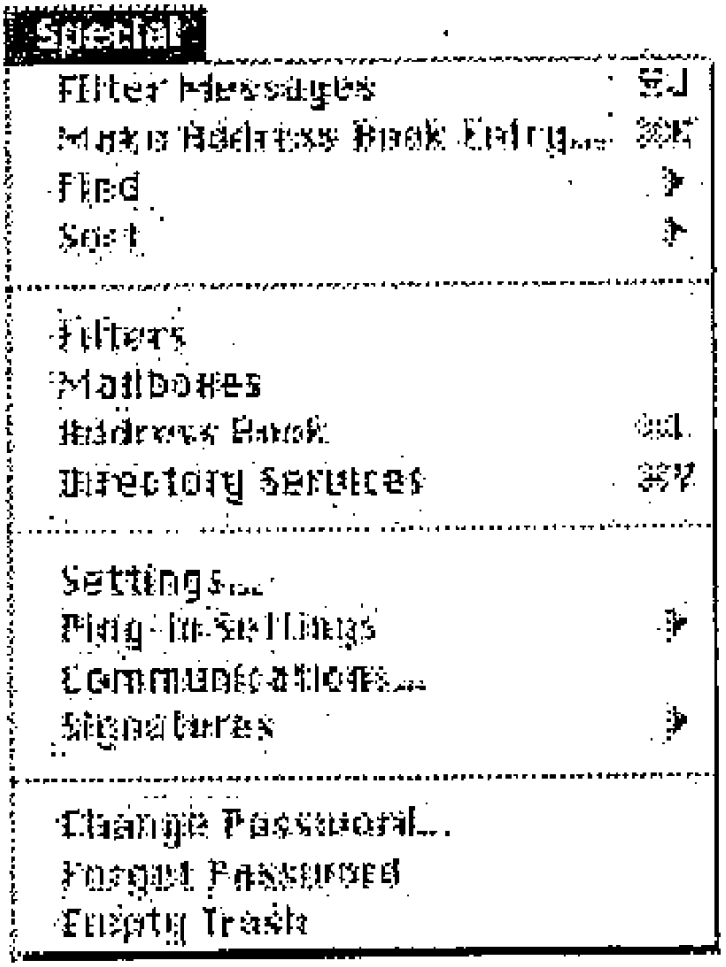
	This menu provides text editing tools.  * * * **Finish Address Book Entry** Complete the partial text of a nickname. **Insert Recipient** Insert the chosen recipient" Eudora Windows Manual at 137. "**Special** This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).

Make Address Book Entry…
[shift] Make Address Book Entry From Selection…
Create an Address Book entry (nickname) from the current message.
Create an entry from the selected addresses.

Find
Search for the designated character string within a message, messages, mailboxes, or mail folders.
Eudora Mac Manual at 155.

“Special

This menu lets you use additional Eudora functions.

Exhibit K

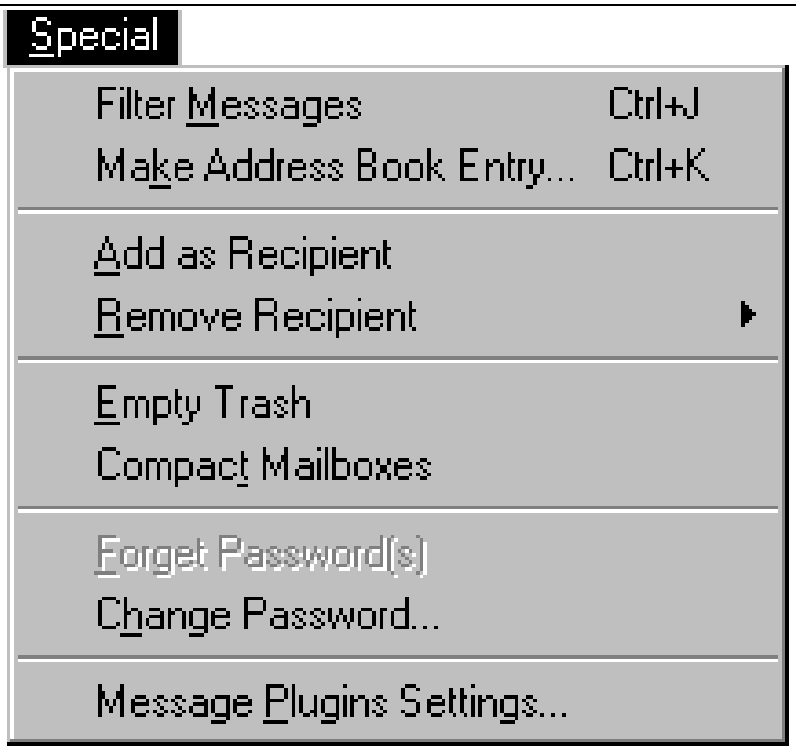
	 **Filter Messages** Run the manual filters for the current message(s). **Make Address Book Entry…** Create an Address Book entry from the current message. **Add As Recipient** Add selected text to the Quick Recipient list. *Eudora Windows Manual* at 141.
providing an input device, configured by the first computer program, that allows a user to enter a user command to initiate an operation, the operation comprising (i) performing a search using at least part of the first information as a search term in order to find the second information, of a specific type or types, associated with the search term in an information source external to the document, wherein the specific type or types of second information is dependent at least in part on the type or types	Eudora discloses this element. See, e.g.: Disclosure to Claim 1. "**Checking Your Spelling** ****** To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight a word or a block of text to check only that text and not the rest of the

Exhibit K

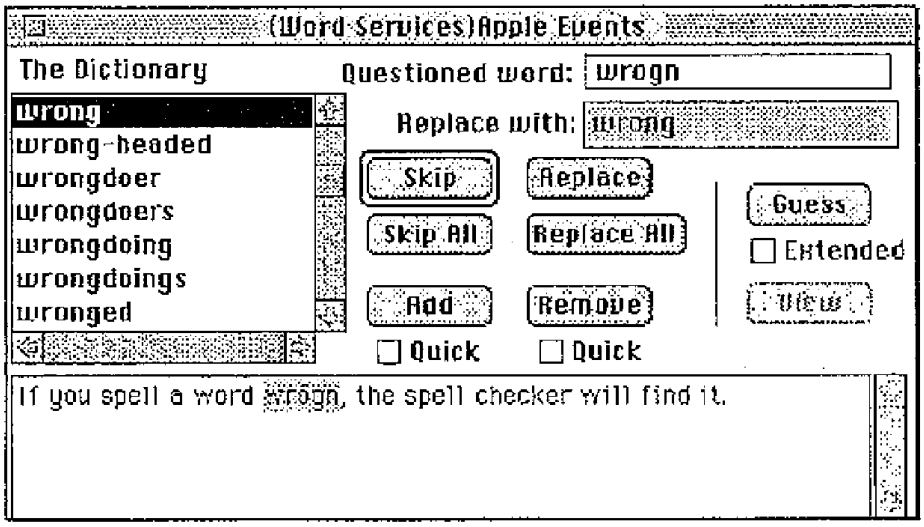
of the first information, and (ii) performing an action using at least part of the second information;	message. If no misspellings are found, the spelling checker quits. If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.  *The Check Spelling dialog* The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check. ****** The Check Spelling Dialog The Check Spelling dialog allows you to skip a questioned word, replace it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below. Questioned word A word that is not found in the spelling checker dictionary. Replace with

Exhibit K

	Replace the questioned word with the word in this field. You can select a word from the Dictionary/Guesses field, or type a new one. Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. ****** **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View** Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field. ****** **View suggestions instead of dictionary first** Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word. Eudora Mac Manual at 42-47. **"Checking Your Spelling**

Exhibit K

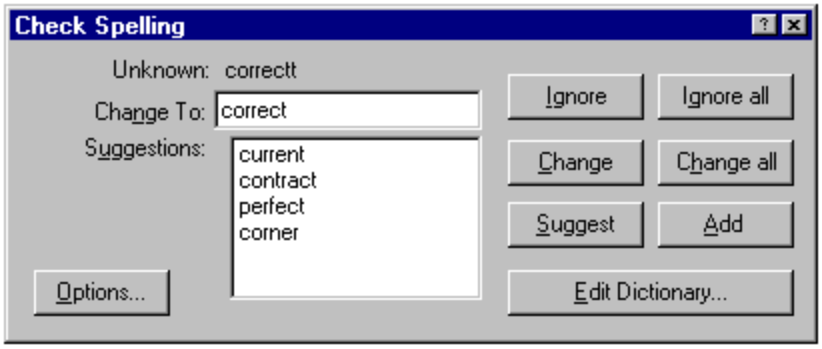
******

To automatically check spelling when you send or queue a message, turn on the **Check when message queue/send selected** option in the Spell Checking Options. If this is on, when you send or queue a message the message is checked for spelling errors. If you go through the spell checking process, the message is automatically sent or queued. If you click Cancel, or leave spelling errors in the message, a dialog is displayed asking you if you still want to send or queue the message. If you don't want that dialog to be displayed, turn on the Don't warn me anymore option (this can also be set in the Spell Checking Options).

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

To correct the misspelled word, type the correct spelling of the word in the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check.

Check Spelling Dialog
The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below.

Exhibit K

	Unknown Field An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word. **Suggest Button** This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* ****** **Make Suggestions** Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options,. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed."* Eudora Windows Manual at 33-37.

Exhibit K

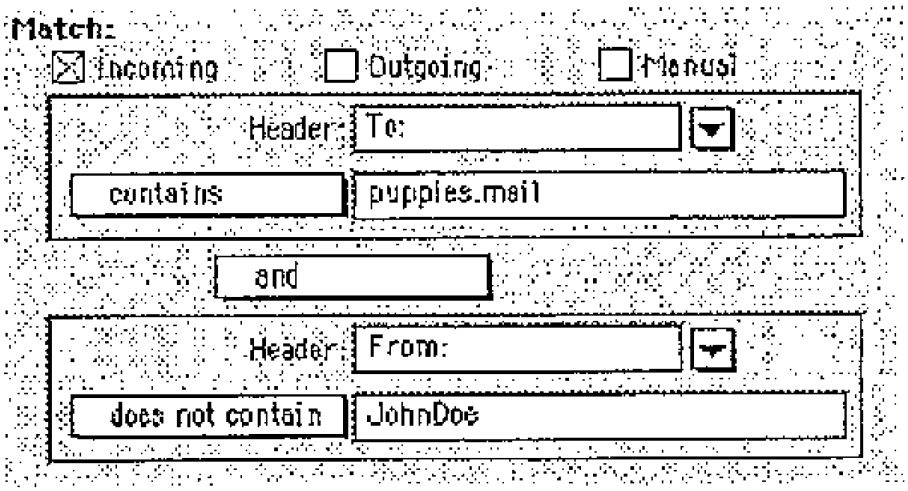
	"**Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters. When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered. *Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.* ******  *Sample Match Area in Filters window* Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the

Exhibit K

	message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches the first term, filter it *unless* the message also matches the second term, in which case *do not* filter it. (This lets you exclude certain variations of the first term.) Filter Actions All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to messages. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages. Label colors and names are set in the Macintosh Labels control panel and the Eudora Labels Settings. **Make Personality** Assigns the selected personality to messages. For outgoing messages, the message is sent from the assigned personality. For incoming messages, all your responses to the message will be from the assigned personality until you change the personality associated with the incoming message or your response. For more information, see the section 'Using an Alternate E-mail Account.'

Exhibit K

	Make Subject Assigns the new subject to messages. If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol to stand for the old subject if you want to add the new subject to the old subject. For example, entering **New Subject [was &]** results in **New Subject [was Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Settings. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. One typical use of this action is to reply to specific senders with stationery telling them that you're on vacation: "I'm out till the 10th. I'll reply to your message when I get back." For more details, see the section 'Using Stationery Files.' **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section

Exhibit K

	'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the filter list). The **Last used** field displays the date the filter was last used on a message. This helps you identify filters that are no longer useful and can be safely deleted." Eudora Mac Manual at 82-87. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters. ****** When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered. ****** **Filter Criteria (the Match Area)** Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup.

Exhibit K

	Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ● «Body» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **Blah Blah Blah Blah** option, and the «Body» option searches the message body. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are: **contains** or **doesn't contain** If the specified header item contains or does not contain the text string, filter the message. **is** or **is not** If the specified header item is or is not a complete match of the text string, filter the message. **starts with** or **ends with** If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored. **appears** or **doesn't appear** If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain. **intersects nickname** If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message.

	Use the **Text** fields to specify the text strings that the filter is searching for. *Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.* Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches both the first and second terms, do not filter it. (This lets you exclude certain variations of the first term.) **Filter Actions** All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to message summaries. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label**

Exhibit K

	Assigns the selected label to messages. **Make Subject** Assigns the new subject to messages (does not affect the subject in the message itself). If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol if you want to add the new subject to the old subject. For example, entering **New Subject:&** results in **New Subject:Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Options. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Notify Application** Notifies the selected application when messages are received, and provides information from the message. Specify the application to use and the part of the message to be included. Use the Browse button to select an application, or enter the command line yourself. The command line should include the path to the executable, any options, and the following substitution variables, all separated by blank spaces: %1 Date %2 To %3 From %4 Subject %5 Cc %6 The entire message For example, the command line to send the subject of a message to a

Exhibit K

	pager might look like this: C:\apps\pager.exe -c %4 **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the list)." *Eudora Windows Manual* at 72-77. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.

Exhibit K

Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog.

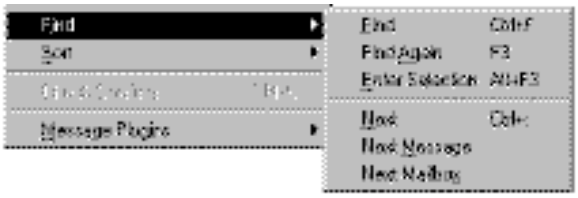
If you need to specify how the text should appear, use the **Whole word** and **Match case** options:

Whole word
If this option is on, the text is found only if it appears by itself and is not part of another word. For example, if the text is 'info' then the word 'information' will be passed over.

Match case
If this option is on, exact matches of the text are found, taking the capitalization into account.

When the **Find** field is filled in and the appropriate options are set, you can use one of two functions in the dialog to find the text: **Find** and **Search**. Use the **Find** button to find instances of the text in just the current open message, or use any of the **Search** buttons to find instances of the text by searching mailboxes or mail folders."
Eudora Mac Manual at 89-90.

"Finding Text Within Messages
Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.



The Find submenu

Exhibit K

Finding Text Within One Message
To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field.

Type the text you want to find in the text field. When finished entering the\ desired text, click the **Find** button.

Finding text

Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed.

If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted.

To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only."
Eudora Windows Manual at 78.

"Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox."
Eudora Review at 5.

"**The 'Make Address Book Entry' Command**

******

From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the

Exhibit K

	addresses that you selected. *Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.* ****** **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings." Eudora Mac Manual at 99-101. **"The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options." Eudora Windows Manual at 89. "**Using the Quick Recipient List** ****** To insert the real address(es), instead of the nickname, hold down the

Exhibit K

option key and select **Insert & Expand Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail Settings."
Eudora Mac Manual at 102.

"Using the Quick Recipient List

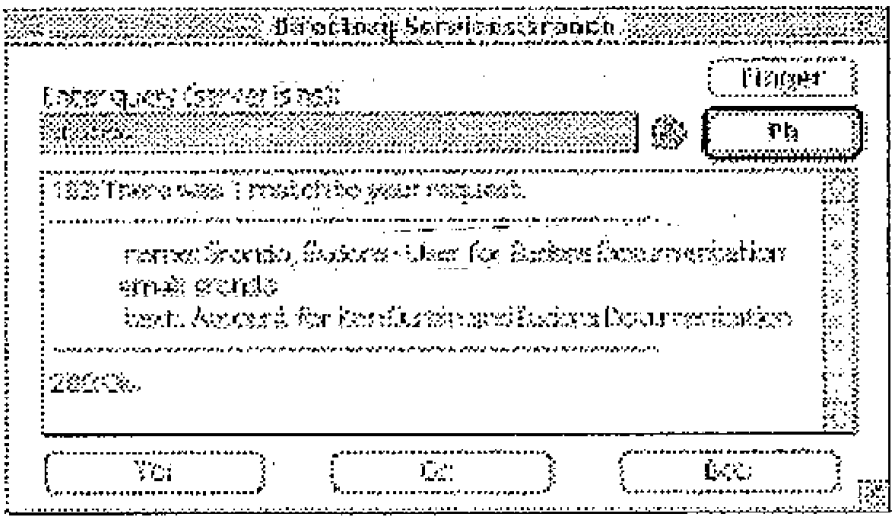
******

To insert the real address(es), instead of a nickname, hold down the Shift key and select **Insert Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options."
Eudora Windows Manual at 90.

"**Using Directory Services**

******

To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.



A Ph query and its response

Note: You can type any Ph command in the query field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

If the **'Live' Ph queries** option is on in your Hosts Settings, the

Exhibit K

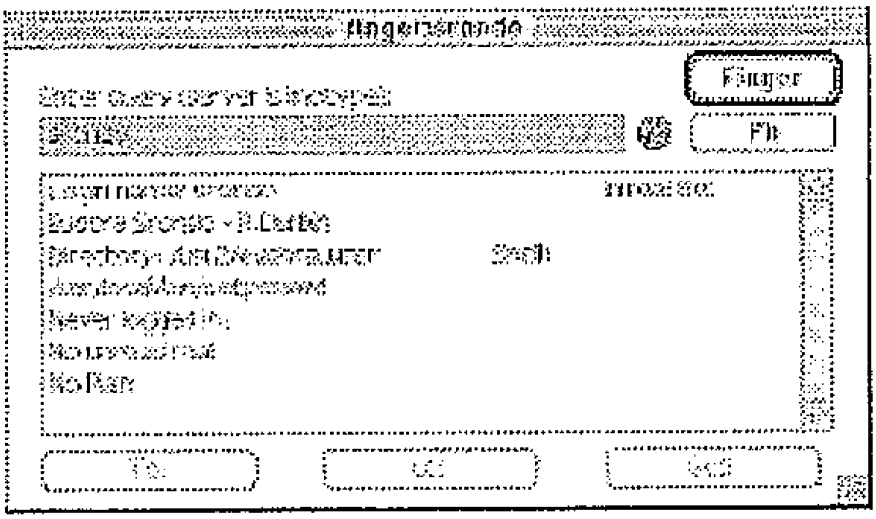
connection with your server is kept open and Ph queries are automatically sent to the server when you finish typing the query text.

*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

******

Using Finger

To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."



A Finger query and its response

Addressing a Message from the Directory Services Window

You can create and address a message with the query results in the Directory Services window.

To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results.

Exhibit K

To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-05.

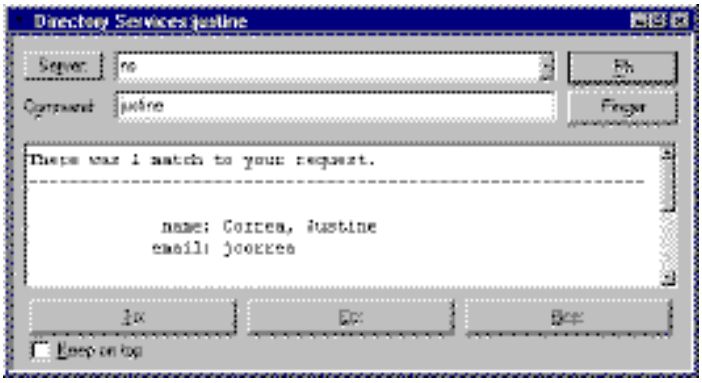
"**Using Directory Services**

******

Using Ph

To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window.

If there is more than one match, you can use the Tab key to move down to the next one, or hold down the Shift key and press Tab to move up to the previous one.



A Ph command and its response

******

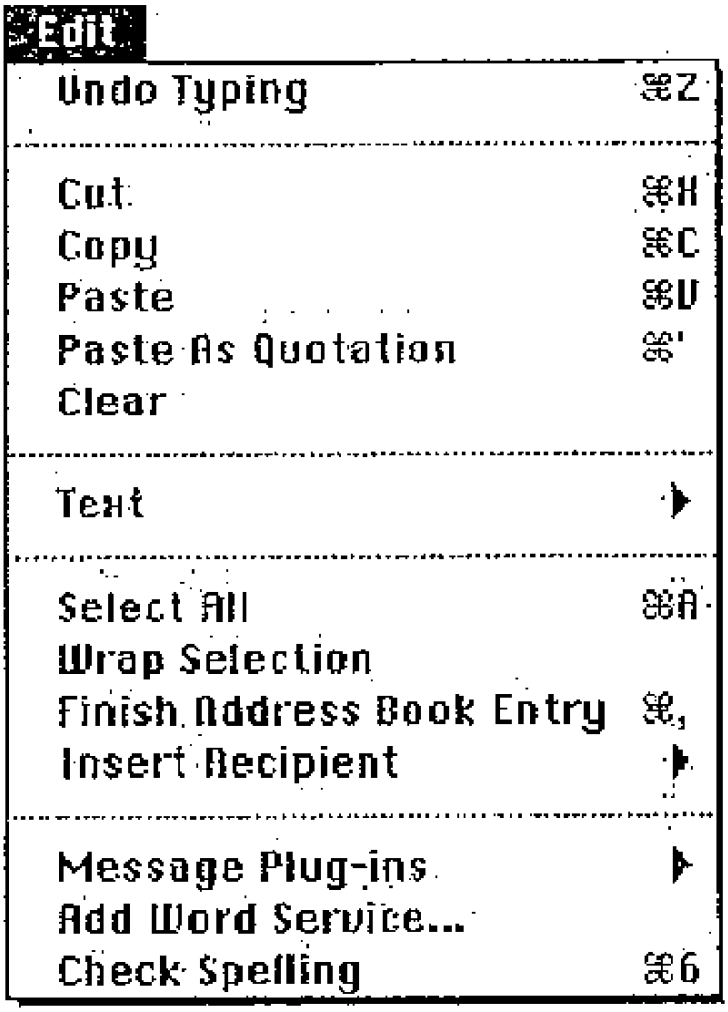
Using Finger

To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is

Exhibit K

	used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window. **Addressing a Message from the Directory Services Window** You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." Eudora Windows Manual at 91-92. "**Edit** This menu provides text editing tools.

Exhibit K



* * *

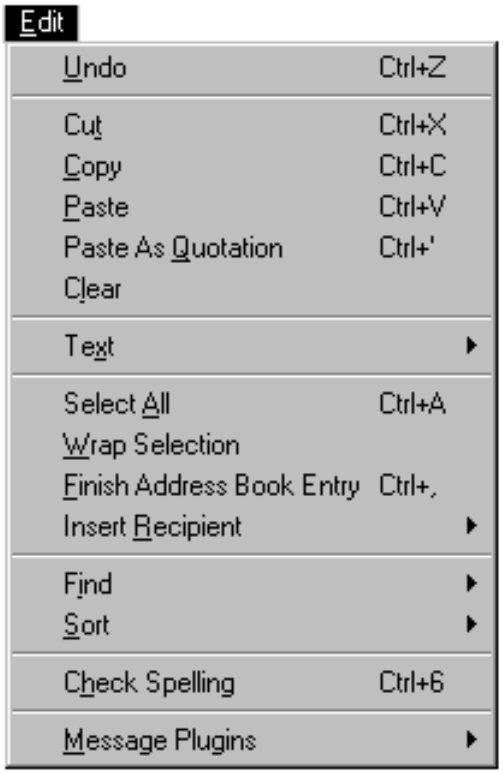
Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-50.

"Edit

This menu provides text editing tools.

Exhibit K



* * *

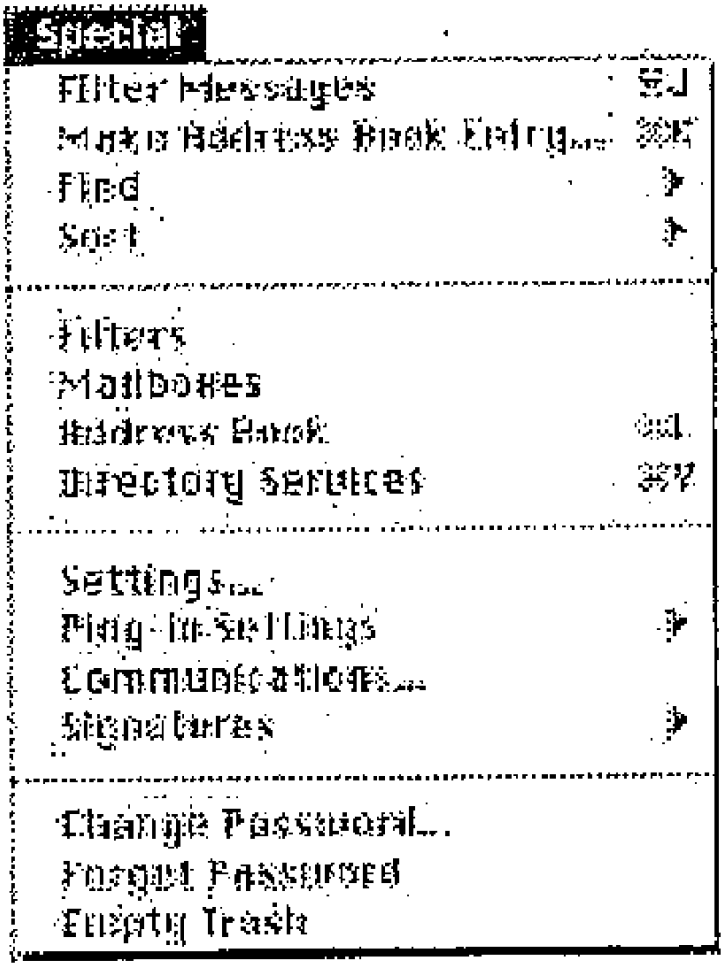
Finish Address Book Entry
Complete the partial text of a nickname.

Insert Recipient
Insert the chosen recipient"
Eudora Windows Manual at 137.

"**Special**

This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).
Eudora Mac Manual at 155.

"Special

This menu lets you use additional Eudora functions.

Exhibit K

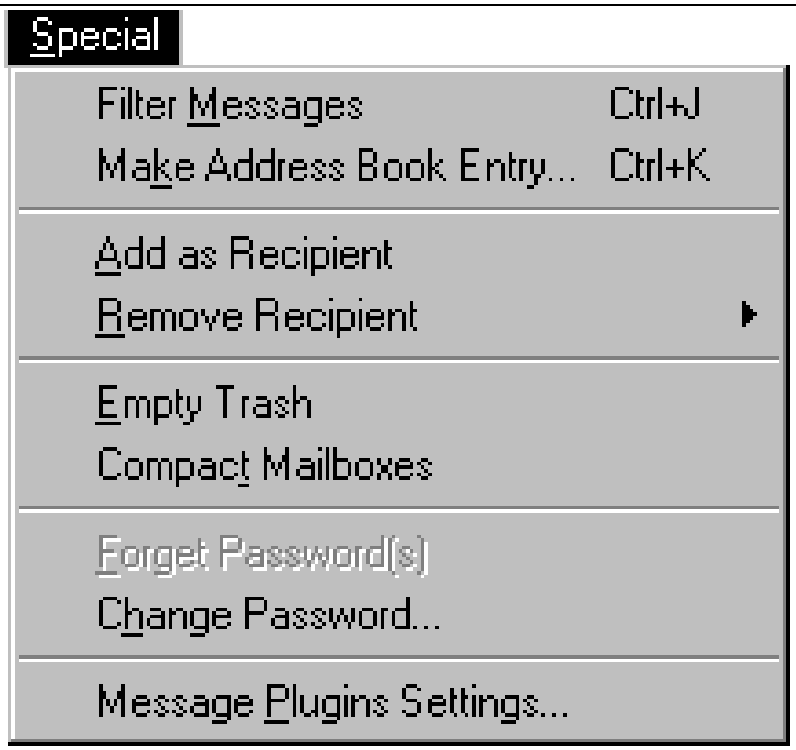
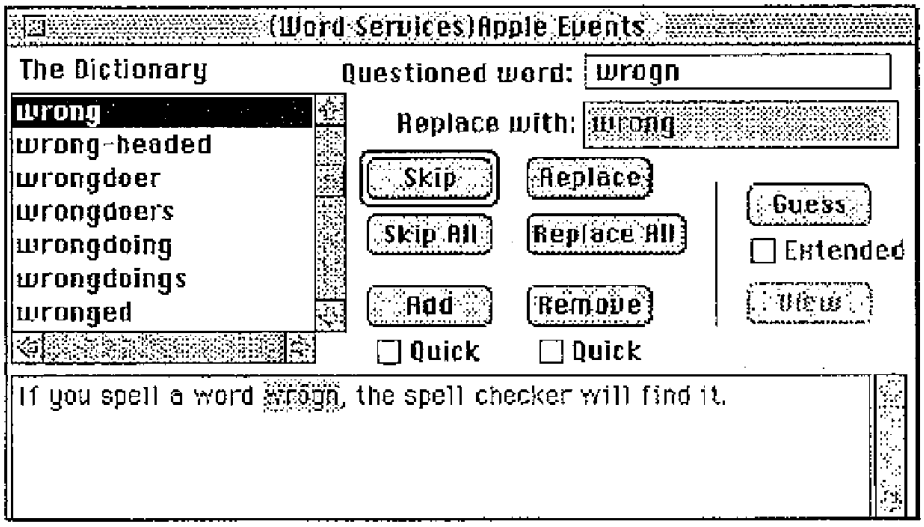
	 **Filter Messages** Run the manual filters for the current message(s). Eudora Windows Manual at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 6, 8, 9, 11, 12, 14, 19, and 20.
in consequence of receipt by the first computer program of the user command from the input device, causing a search for the search term in the information source, using a second computer program, in order to find second information related to the search term; and	Eudora discloses this element. See, e.g.: Disclosure to Claim 1. "**Checking Your Spelling** ****** To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight a word or a block of text to check only that text and not the rest of the message.

Exhibit K

If no misspellings are found, the spelling checker quits.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.



The Check Spelling dialog

The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check.

******

The Check Spelling Dialog

The Check Spelling dialog allows you to skip a questioned word, replace it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below.

Questioned word
A word that is not found in the spelling checker dictionary.

Replace with
Replace the questioned word with the word in this field. You can select a

Exhibit K

	word from the Dictionary/Guesses field, or type a new one. Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. ****** **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View** Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field. ****** **View suggestions instead of dictionary first** Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word. Eudora Mac Manual at 42-47. **"Checking Your Spelling** ******

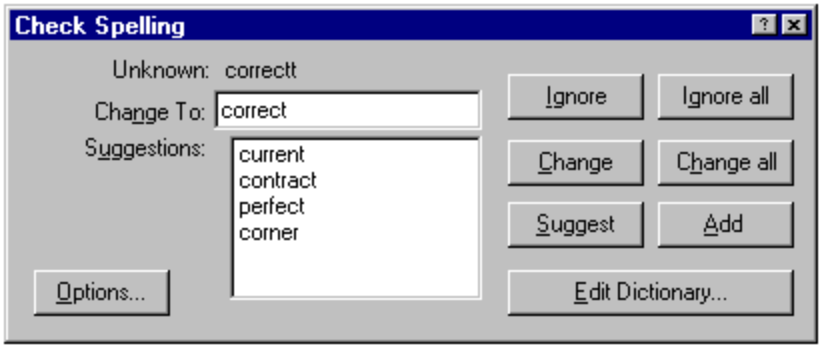
Exhibit K

To automatically check spelling when you send or queue a message, turn on the **Check when message queue/send selected** option in the Spell Checking Options. If this is on, when you send or queue a message the message is checked for spelling errors. If you go through the spell checking process, the message is automatically sent or queued. If you click Cancel, or leave spelling errors in the message, a dialog is displayed asking you if you still want to send or queue the message. If you don't want that dialog to be displayed, turn on the Don't warn me anymore option (this can also be set in the Spell Checking Options).

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

To correct the misspelled word, type the correct spelling of the word in the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check.

Check Spelling Dialog
The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below.

Exhibit K

	Unknown Field An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word. **Suggest Button** This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* ****** **Make Suggestions** Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options,. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed."* Eudora Windows Manual at 33-37. "**Filtering Messages**

Exhibit K

Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot."

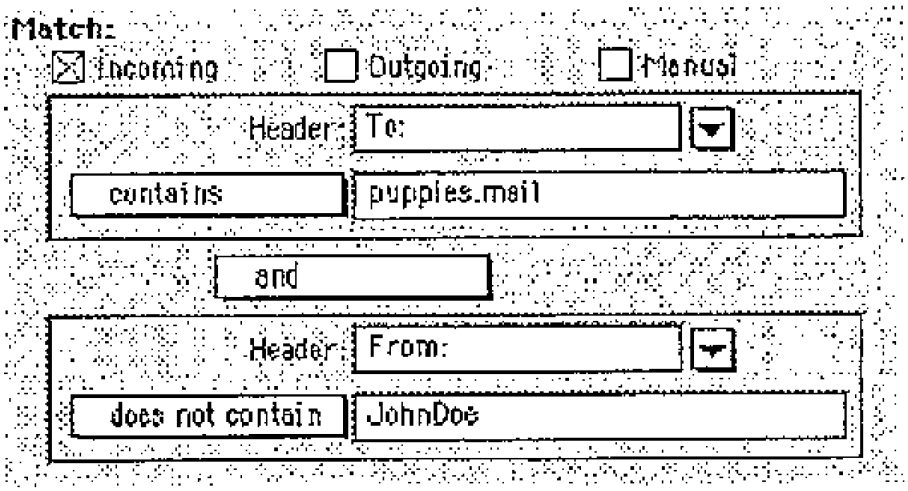
******

Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters.

When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered.

Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.

******



Sample Match Area in Filters window

Use the **Conjunction** popup to link the two terms. The conjunction options are:

ignore
Ignore the second term; if the message matches the first term, filter the message.

Exhibit K

	and If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches the first term, filter it *unless* the message also matches the second term, in which case *do not* filter it. (This lets you exclude certain variations of the first term.) Filter Actions All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to messages. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages. Label colors and names are set in the Macintosh Labels control panel and the Eudora Labels Settings. **Make Personality** Assigns the selected personality to messages. For outgoing messages, the message is sent from the assigned personality. For incoming messages, all your responses to the message will be from the assigned personality until you change the personality associated with the incoming message or your response. For more information, see the section 'Using an Alternate E-mail Account.' **Make Subject**

	Assigns the new subject to messages. If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol to stand for the old subject if you want to add the new subject to the old subject. For example, entering **New Subject [was &]** results in **New Subject [was Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Settings. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. One typical use of this action is to reply to specific senders with stationery telling them that you're on vacation: "I'm out till the 10th. I'll reply to your message when I get back." For more details, see the section 'Using Stationery Files.' **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server').

Exhibit K

	Copy To Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the filter list). The **Last used** field displays the date the filter was last used on a message. This helps you identify filters that are no longer useful and can be safely deleted." Eudora Mac Manual at 82-87. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters. ****** When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered. ****** **Filter Criteria (the Match Area)** Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup.

Exhibit K

	Use the **Header** field to specify which message header items you want the filter to search. You can select an option from the popup menu or enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows: ● To ● Cc ● From ● Subject ● Reply-To ● «Any Recipient» ● «Any Header» ● «Body» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **Blah Blah Blah Blah** option, and the «Body» option searches the message body. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are: **contains** or **doesn't contain** If the specified header item contains or does not contain the text string, filter the message. **is** or **is not** If the specified header item is or is not a complete match of the text string, filter the message. **starts with** or **ends with** If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored. **appears** or **doesn't appear** If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain. **intersects nickname** If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message. Use the **Text** fields to specify the text strings that the filter is searching

	for. *Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.* Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches both the first and second terms, do not filter it. (This lets you exclude certain variations of the first term.) **Filter Actions** All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to message summaries. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages.

	Make Subject Assigns the new subject to messages (does not affect the subject in the message itself). If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol if you want to add the new subject to the old subject. For example, entering **New Subject:&** results in **New Subject:Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Options. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Notify Application** Notifies the selected application when messages are received, and provides information from the message. Specify the application to use and the part of the message to be included. Use the Browse button to select an application, or enter the command line yourself. The command line should include the path to the executable, any options, and the following substitution variables, all separated by blank spaces: %1 Date %2 To %3 From %4 Subject %5 Cc %6 The entire message For example, the command line to send the subject of a message to a pager might look like this:

Exhibit K

	C:\apps\pager.exe -c %4 **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the list)." *Eudora Windows Manual* at 72-77. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.

Exhibit K

Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog.

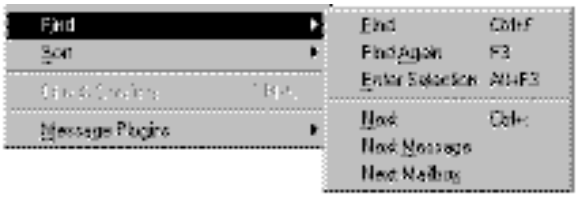
If you need to specify how the text should appear, use the **Whole word** and **Match case** options:

Whole word
If this option is on, the text is found only if it appears by itself and is not part of another word. For example, if the text is 'info' then the word 'information' will be passed over.

Match case
If this option is on, exact matches of the text are found, taking the capitalization into account.

When the **Find** field is filled in and the appropriate options are set, you can use one of two functions in the dialog to find the text: **Find** and **Search**. Use the **Find** button to find instances of the text in just the current open message, or use any of the **Search** buttons to find instances of the text by searching mailboxes or mail folders."
Eudora Mac Manual at 89-90.

"Finding Text Within Messages
Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.



The Find submenu

Exhibit K

Finding Text Within One Message
To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field.

Type the text you want to find in the text field. When finished entering the\ desired text, click the **Find** button.

Finding text

Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed.

If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted.

To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only."
Eudora Windows Manual at 78.

"Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox."
Eudora Review at 5.

"**The 'Make Address Book Entry' Command**

******

From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the

Exhibit K

	addresses that you selected. *Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.* ****** **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings." Eudora Mac Manual at 99-101. **"The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options." Eudora Windows Manual at 89. "**Using the Quick Recipient List** ****** To insert the real address(es), instead of the nickname, hold down the

Exhibit K

option key and select **Insert & Expand Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail Settings."
Eudora Mac Manual at 102.

"Using the Quick Recipient List

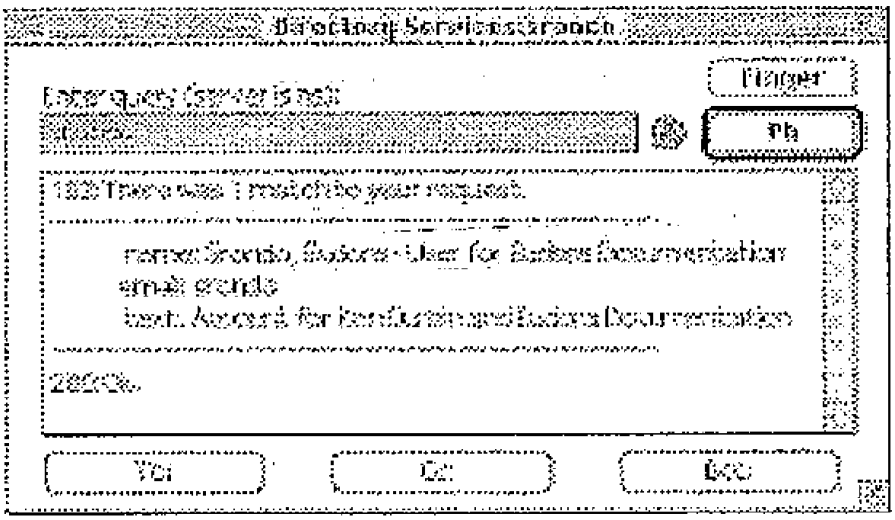
******

To insert the real address(es), instead of a nickname, hold down the Shift key and select **Insert Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options."
Eudora Windows Manual at 90.

"**Using Directory Services**

******

To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.



A Ph query and its response

Note: You can type any Ph command in the query field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

If the **'Live' Ph queries** option is on in your Hosts Settings, the

Exhibit K

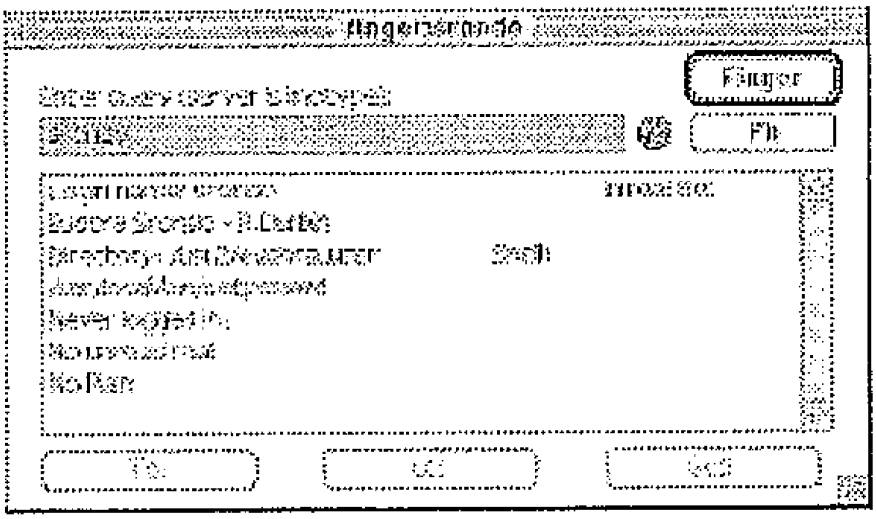
connection with your server is kept open and Ph queries are automatically sent to the server when you finish typing the query text.

*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

******

Using Finger

To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."



A Finger query and its response

Addressing a Message from the Directory Services Window

You can create and address a message with the query results in the Directory Services window.

To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results.

Exhibit K

To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-05.

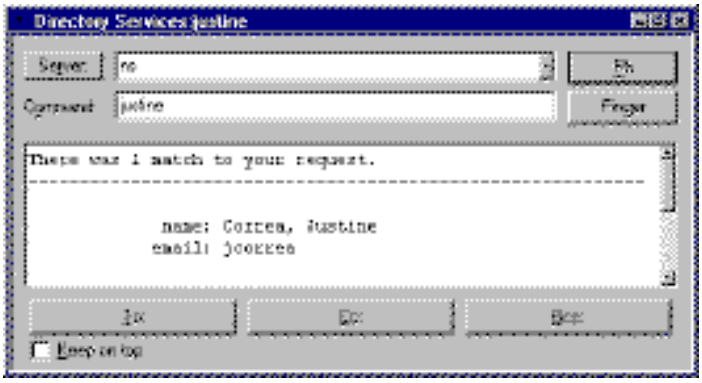
"**Using Directory Services**

******

Using Ph

To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window.

If there is more than one match, you can use the Tab key to move down to the next one, or hold down the Shift key and press Tab to move up to the previous one.



A Ph command and its response

******

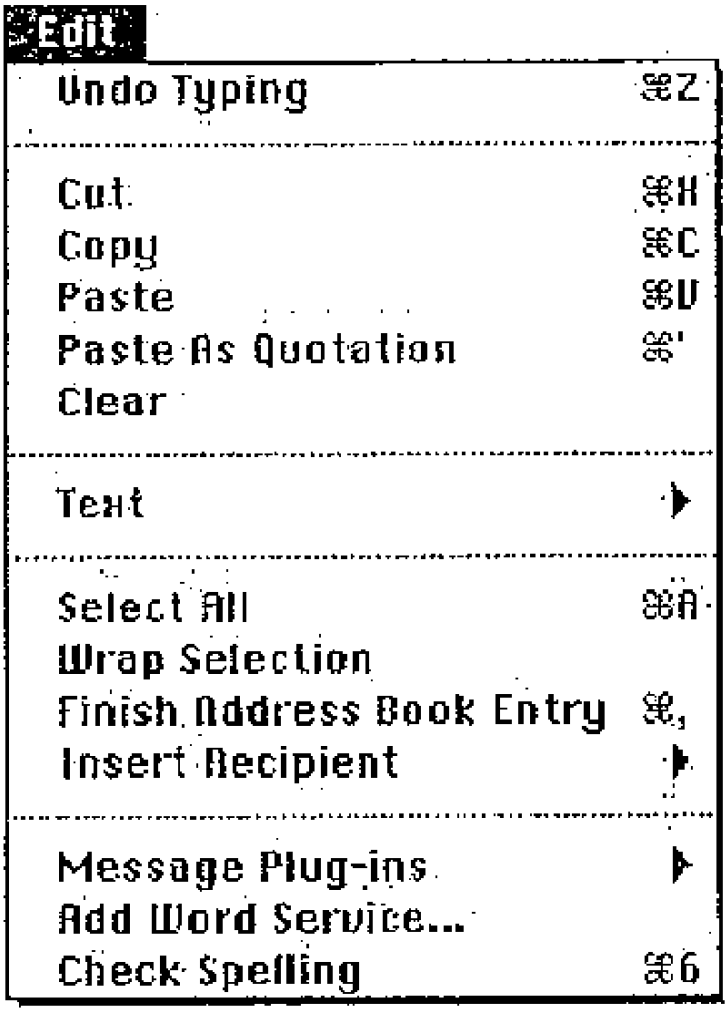
Using Finger

To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is

Exhibit K

	used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window. **Addressing a Message from the Directory Services Window** You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." Eudora Windows Manual at 91-92. "**Edit** This menu provides text editing tools.

Exhibit K



* * *

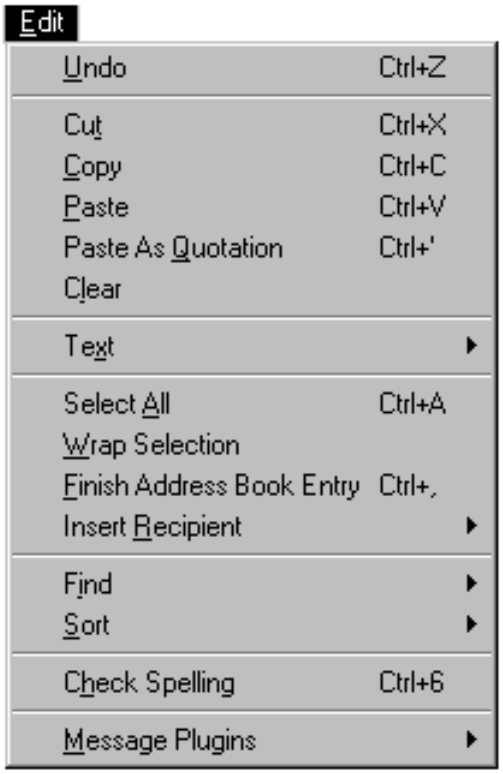
Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-50.

"Edit

This menu provides text editing tools.

Exhibit K



* * *

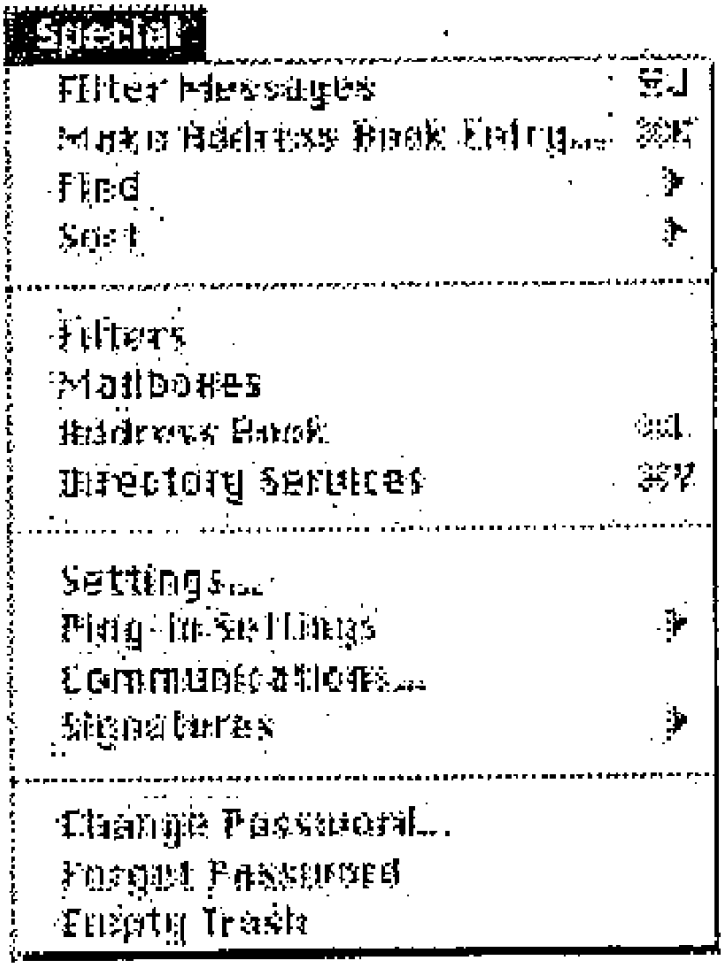
Finish Address Book Entry
Complete the partial text of a nickname.

Insert Recipient
Insert the chosen recipient"
Eudora Windows Manual at 137.

"**Special**

This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).
Eudora Mac Manual at 155.

"Special

This menu lets you use additional Eudora functions.

Exhibit K

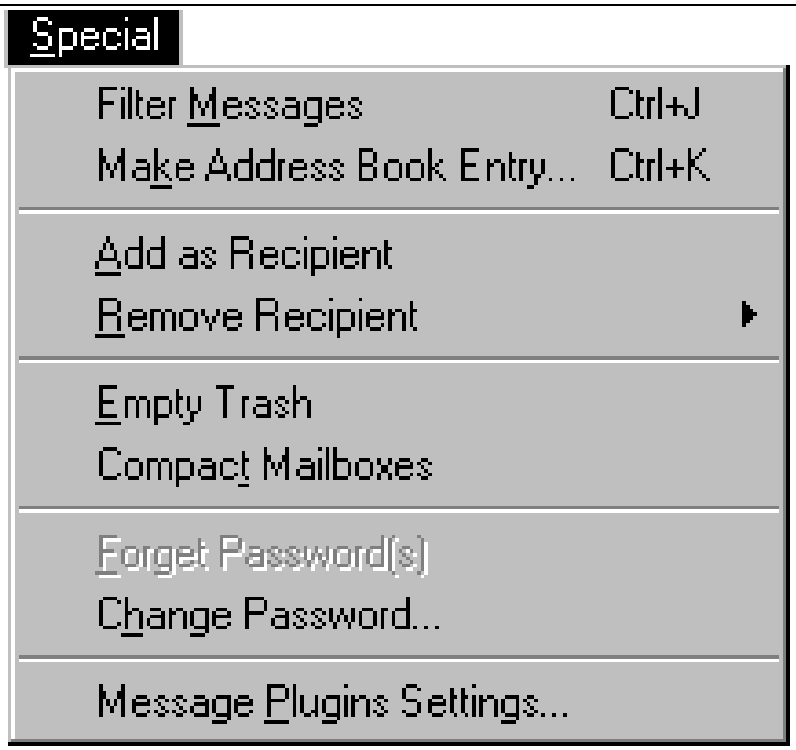
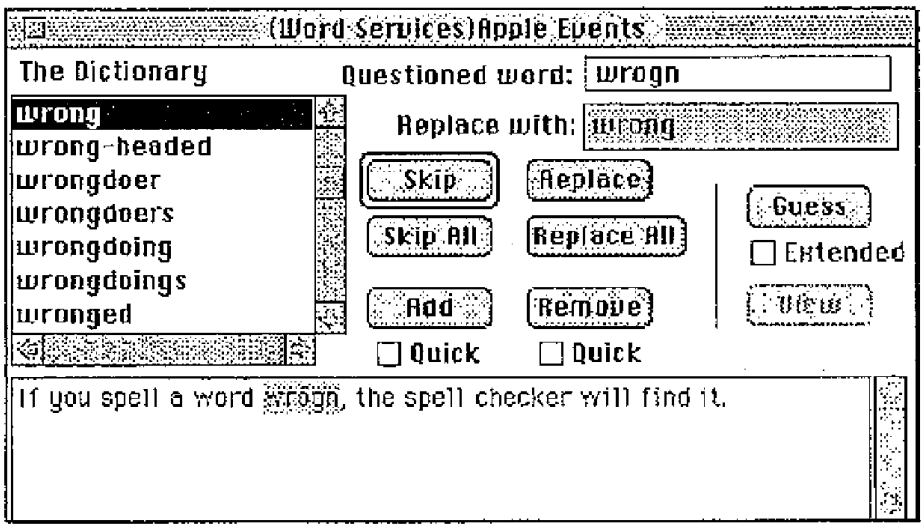
	 **Filter Messages** Run the manual filters for the current message(s). Eudora Windows Manual at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 2, 10, and 19.
if searching finds any second information related to the search term, performing the action using at least part of the second information, wherein the action is of a type depending at least in part on the type or types of the first information.	Eudora discloses this element. See, e.g.: Disclosure to Claim 1. "**Checking Your Spelling** ****** To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight a word or a block of text to check only that text and not the rest of the message.

Exhibit K

If no misspellings are found, the spelling checker quits.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.



The Check Spelling dialog

The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check.

******

The Check Spelling Dialog

The Check Spelling dialog allows you to skip a questioned word, replace it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below.

Questioned word
A word that is not found in the spelling checker dictionary.

Replace with
Replace the questioned word with the word in this field. You can select a word from the Dictionary/Guesses field, or type a new one.

Exhibit K

	Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. ****** **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View** Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field. ****** **View suggestions instead of dictionary first** Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word. Eudora Mac Manual at 42-47. **"Checking Your Spelling** ******

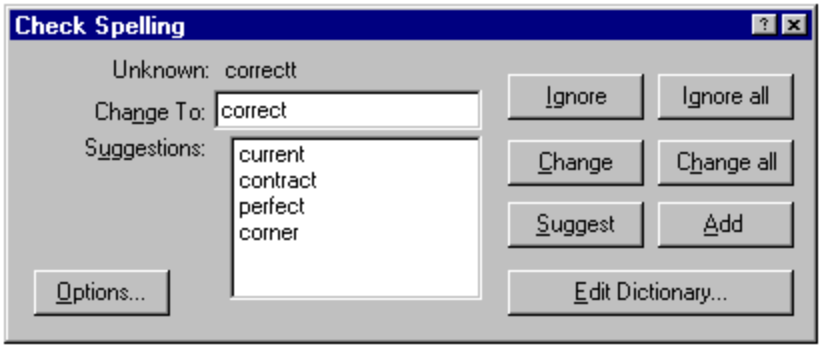
Exhibit K

To automatically check spelling when you send or queue a message, turn on the **Check when message queue/send selected** option in the Spell Checking Options. If this is on, when you send or queue a message the message is checked for spelling errors. If you go through the spell checking process, the message is automatically sent or queued. If you click Cancel, or leave spelling errors in the message, a dialog is displayed asking you if you still want to send or queue the message. If you don't want that dialog to be displayed, turn on the Don't warn me anymore option (this can also be set in the Spell Checking Options).

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

To correct the misspelled word, type the correct spelling of the word in the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check.

Check Spelling Dialog
The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below.

Unknown Field

Exhibit K

	An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word. **Suggest Button** This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* ****** **Make Suggestions** Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options,. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed."* Eudora Windows Manual at 33-37. "**Filtering Messages**

Exhibit K

Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot."

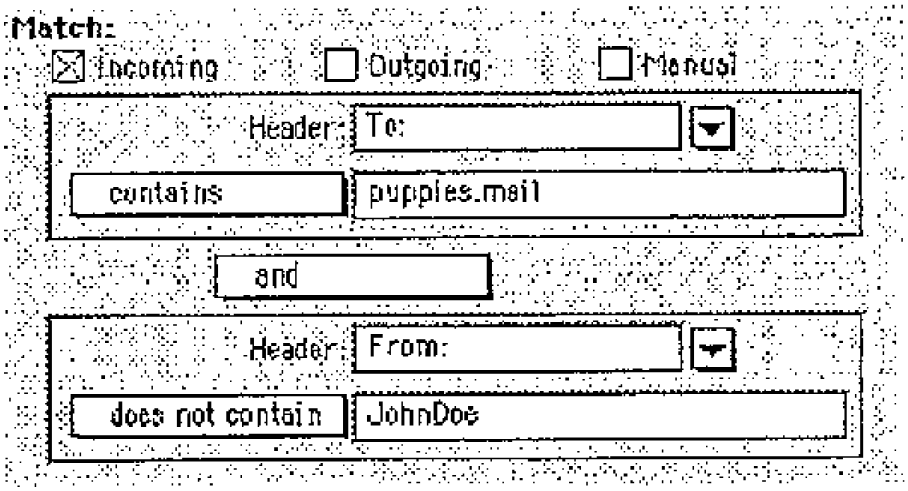
******

Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters.

When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered.

Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match.

******



Sample Match Area in Filters window

Use the **Conjunction** popup to link the two terms. The conjunction options are:

ignore
Ignore the second term; if the message matches the first term, filter the message.

Exhibit K

	and If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches the first term, filter it *unless* the message also matches the second term, in which case *do not* filter it. (This lets you exclude certain variations of the first term.) Filter Actions All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to messages. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages. Label colors and names are set in the Macintosh Labels control panel and the Eudora Labels Settings. **Make Personality** Assigns the selected personality to messages. For outgoing messages, the message is sent from the assigned personality. For incoming messages, all your responses to the message will be from the assigned personality until you change the personality associated with the incoming message or your response. For more information, see the section 'Using an Alternate E-mail Account.' **Make Subject** Assigns the new subject to messages. If you choose this option, the entire

	subject of the message is replaced with the new subject. Use the '&' symbol to stand for the old subject if you want to add the new subject to the old subject. For example, entering **New Subject [was &]** results in **New Subject [was Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Settings. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. One typical use of this action is to reply to specific senders with stationery telling them that you're on vacation: "I'm out till the 10th. I'll reply to your message when I get back." For more details, see the section 'Using Stationery Files.' **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server').

Exhibit K

	Copy To Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the filter list). The **Last used** field displays the date the filter was last used on a message. This helps you identify filters that are no longer useful and can be safely deleted." Eudora Mac Manual at 82-87. **"Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot." ****** Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters. ****** When the filters are invoked (automatically or manually), each message is matched against each filter in order from top to bottom. If the message meets a filter's criteria, the actions are done as specified until there are no more actions, then the message is matched against the next filter. If at any point a **Skip rest** action is done, nothing else is done with that message, and the next message is filtered. ****** **Filter Criteria (the Match Area)** Each filter can use one or two 'terms' as its criteria, connecting them as appropriate with the conjunction popup. Use the **Header** field to specify which message header items you want the

Exhibit K

	filter to search. You can select an option from the popup menu or enter one yourself. This is helpful if you want to search for a header item that does not appear on the menu, such as X-Persona (for an alternate personality). The selections are as follows: • To • Cc • From • Subject • Reply-To • «Any Recipient» • «Any Header» • «Body» The «Any Recipient» option searches all possible recipient items (To, Cc, Bcc); the «Any Header» option searches all message headers (including hidden headers that are shown with the **Blah Blah Blah Blah** option, and the «Body» option searches the message body. Use the **Match Type popup** to control how the header item is matched with the text string in the text field. The match options are: **contains** or **doesn't contain** If the specified header item contains or does not contain the text string, filter the message. **is** or **is not** If the specified header item is or is not a complete match of the text string, filter the message. **starts with** or **ends with** If the specified header item starts with or ends with the text string, filter the message. The **starts with** item refers to the first non-whitespace character after the colon, so any spaces after the colon are ignored. **appears** or **doesn't appear** If the header item appears or does not appear in the message, filter the message (the text field is ignored). This is useful for filtering messages based only on the types of fields they contain. **intersects nickname** If the text string is included in a nickname (whether it is a full address or a nickname within the nickname), filter the message. Use the **Text** fields to specify the text strings that the filter is searching for.

	Note: It is recommended that the contents of this field be kept as specific and brief as possible. The greater the complexity, the less the likelihood of a match. Use the **Conjunction** popup to link the two terms. The conjunction options are: **ignore** Ignore the second term; if the message matches the first term, filter the message. **and** If the message matches both the first and second terms, filter it. **or** If the message matches either term, filter it. **unless** If the message matches both the first and second terms, do not filter it. (This lets you exclude certain variations of the first term.) **Filter Actions** All messages that match the filter criteria are acted on as specified with the **Actions** popups. Each filter can do up to five things to a message that matches the criteria. You can use the same action twice if it does not directly affect the original message (for example, **Copy To** can be used twice, but not **Transfer To**). The Actions options are as follows: **None** No action **Make Status** Assigns the selected status to message summaries. **Make Priority** Assigns the selected priority level to messages. If you select a set level, messages are set to that priority. If you select **Raise** or **Lower**, messages are raised or lowered one priority level based on their pre-filter level. **Make Label** Assigns the selected label to messages.

	Make Subject Assigns the new subject to messages (does not affect the subject in the message itself). If you choose this option, the entire subject of the message is replaced with the new subject. Use the '&' symbol if you want to add the new subject to the old subject. For example, entering **New Subject:&** results in **New Subject:Old Subject]**. **Play Sound** Plays the selected sound when messages are received. **Open** Opens the **Mailbox** and/or **Message** when a message is received. If you set a previous action to filter messages into a mailbox, then that mailbox is opened. **Print** Prints one copy of each messages. **Notify User** Notifies you **As normal** and/or **In report** when messages are received. The **As normal** option notifies you based on the options you have selected in the Getting Attention Options. The **In report** option notifies you by displaying a filter report that details what filter actions have been done. **Notify Application** Notifies the selected application when messages are received, and provides information from the message. Specify the application to use and the part of the message to be included. Use the Browse button to select an application, or enter the command line yourself. The command line should include the path to the executable, any options, and the following substitution variables, all separated by blank spaces: %1 Date %2 To %3 From %4 Subject %5 Cc %6 The entire message For example, the command line to send the subject of a message to a pager might look like this:

Exhibit K

	C:\apps\pager.exe -c %4 **Forward To** Forwards messages to the e-mail address given. Forwarded messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Redirect To** Redirects messages to the e-mail address given. Redirected messages are placed in the queue in the Out mailbox, and sent the next time you send queued messages. **Reply With** Replies to messages with the selected stationery message. Replies are placed in the queue in the Out mailbox and sent the next time you send queued messages. **Server Options** Sets the message's server action to **Fetch** and/or **Delete** (see the section 'Managing Your Mail on the POP Server'). **Copy To** Copies messages to the selected mailbox. **Transfer To** Transfers messages to the selected mailbox. **Skip Rest** Stops filtering for the message (the message is not matched to the rest of the filters in the list)." *Eudora Windows Manual* at 72-77. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.

Exhibit K

Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog.

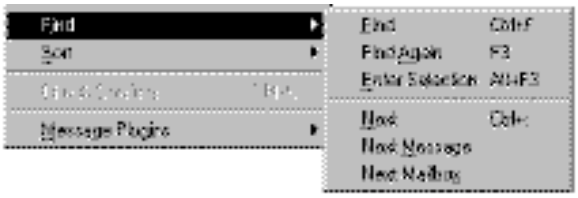
If you need to specify how the text should appear, use the **Whole word** and **Match case** options:

Whole word
If this option is on, the text is found only if it appears by itself and is not part of another word. For example, if the text is 'info' then the word 'information' will be passed over.

Match case
If this option is on, exact matches of the text are found, taking the capitalization into account.

When the **Find** field is filled in and the appropriate options are set, you can use one of two functions in the dialog to find the text: **Find** and **Search**. Use the **Find** button to find instances of the text in just the current open message, or use any of the **Search** buttons to find instances of the text by searching mailboxes or mail folders."
Eudora Mac Manual at 89-90.

"Finding Text Within Messages
Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.



The Find submenu

Exhibit K

Finding Text Within One Message
To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field.

Type the text you want to find in the text field. When finished entering the\ desired text, click the **Find** button.

Finding text

Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed.

If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted.

To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only."
Eudora Windows Manual at 78.

"Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox."
Eudora Review at 5.

"**The 'Make Address Book Entry' Command**

******

From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the

	addresses that you selected. *Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.* ****** **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings." Eudora Mac Manual at 99-101. **"The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options." Eudora Windows Manual at 89. "**Using the Quick Recipient List** ****** To insert the real address(es), instead of the nickname, hold down the

Exhibit K

option key and select **Insert & Expand Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail Settings."
Eudora Mac Manual at 102.

"Using the Quick Recipient List

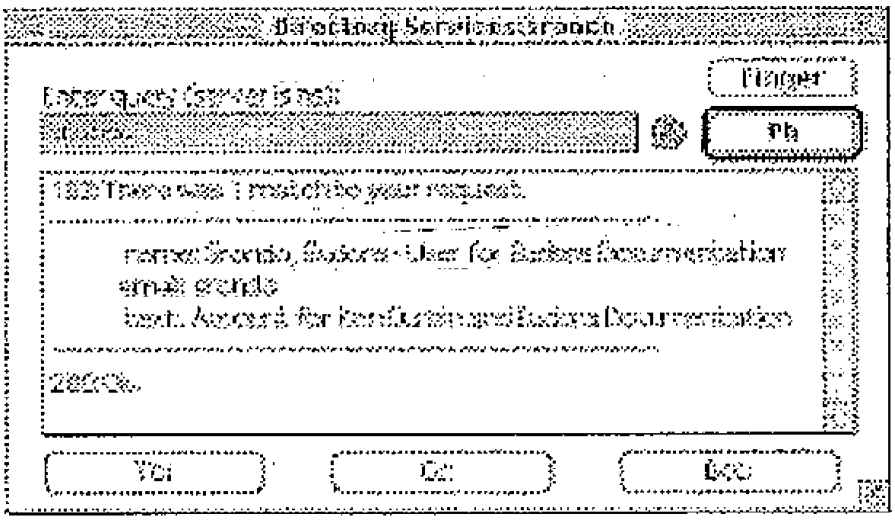
******

To insert the real address(es), instead of a nickname, hold down the Shift key and select **Insert Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options."
Eudora Windows Manual at 90.

"**Using Directory Services**

******

To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.



A Ph query and its response

Note: You can type any Ph command in the query field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

If the **'Live' Ph queries** option is on in your Hosts Settings, the

Exhibit K

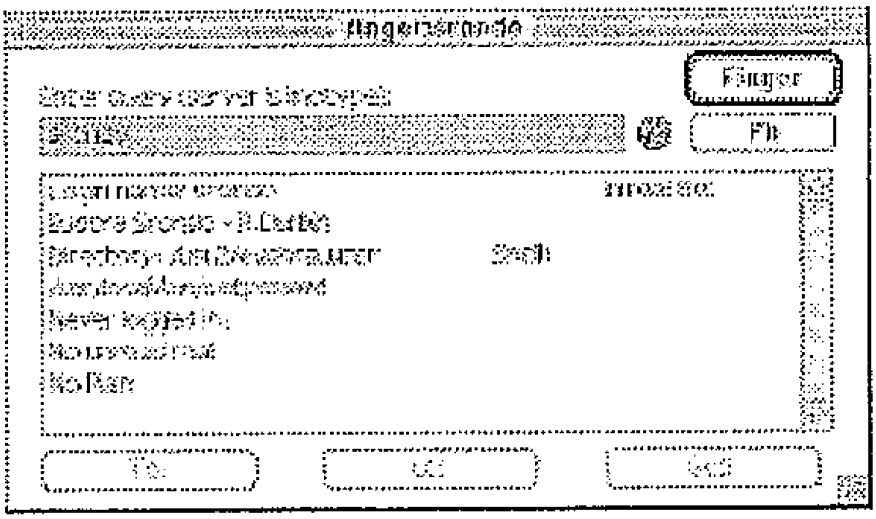
connection with your server is kept open and Ph queries are automatically sent to the server when you finish typing the query text.

*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

******

Using Finger

To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."



A Finger query and its response

Addressing a Message from the Directory Services Window

You can create and address a message with the query results in the Directory Services window.

To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results.

Exhibit K

To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-05.

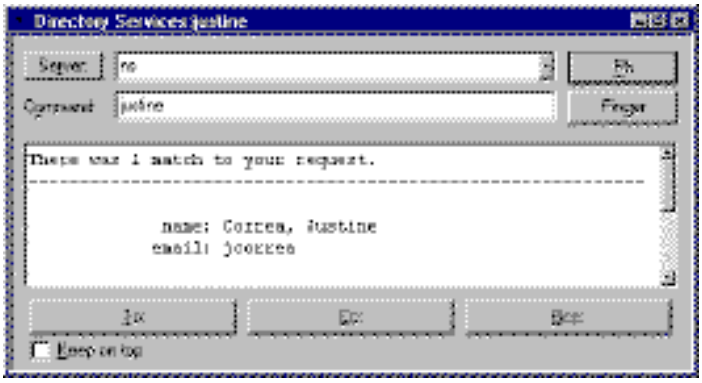
"**Using Directory Services**

******

Using Ph

To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window.

If there is more than one match, you can use the Tab key to move down to the next one, or hold down the Shift key and press Tab to move up to the previous one.



A Ph command and its response

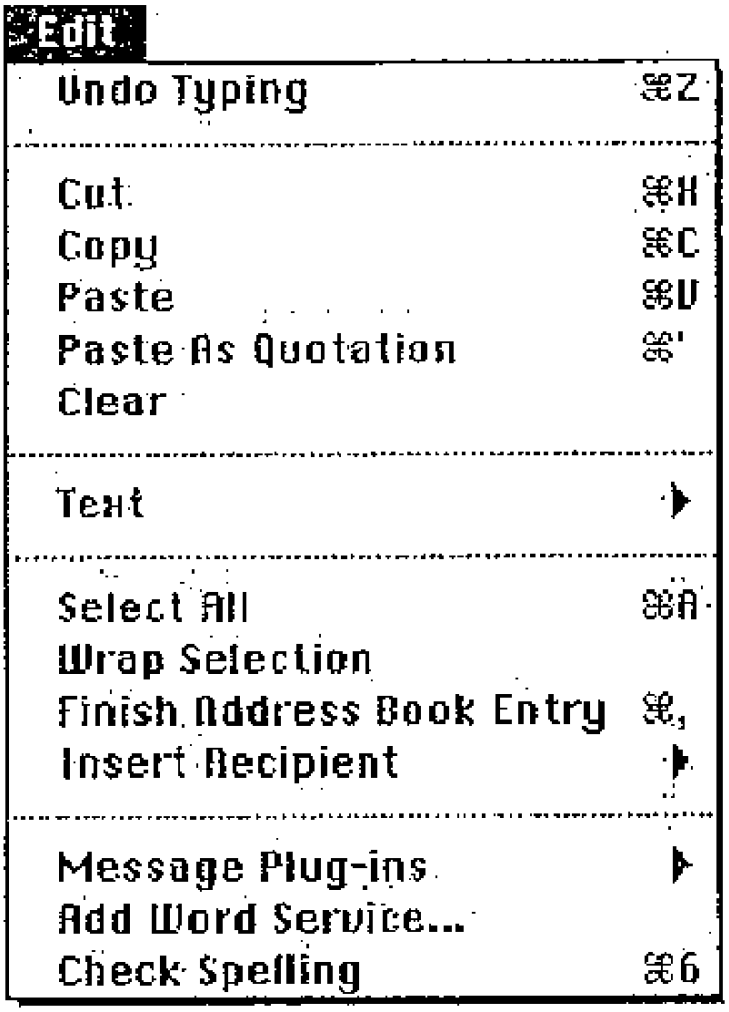
******

Using Finger

To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is

	used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window. **Addressing a Message from the Directory Services Window** You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." Eudora Windows Manual at 91-92. "**Edit** This menu provides text editing tools.

Exhibit K



* * *

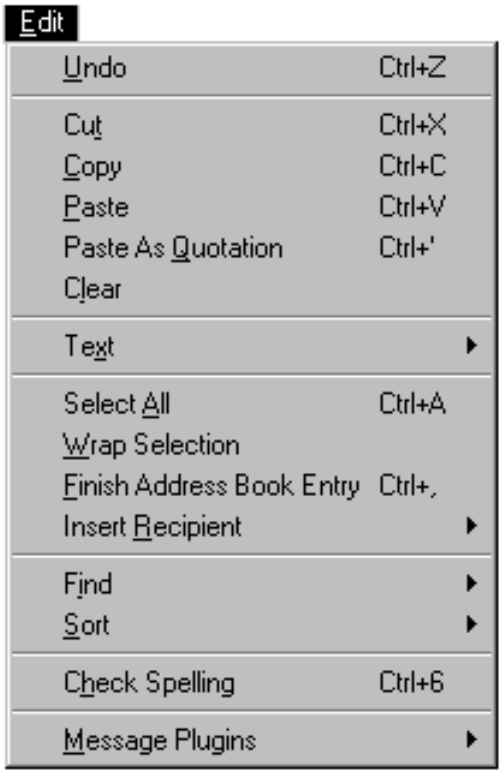
Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-50.

"Edit

This menu provides text editing tools.

Exhibit K



* * *

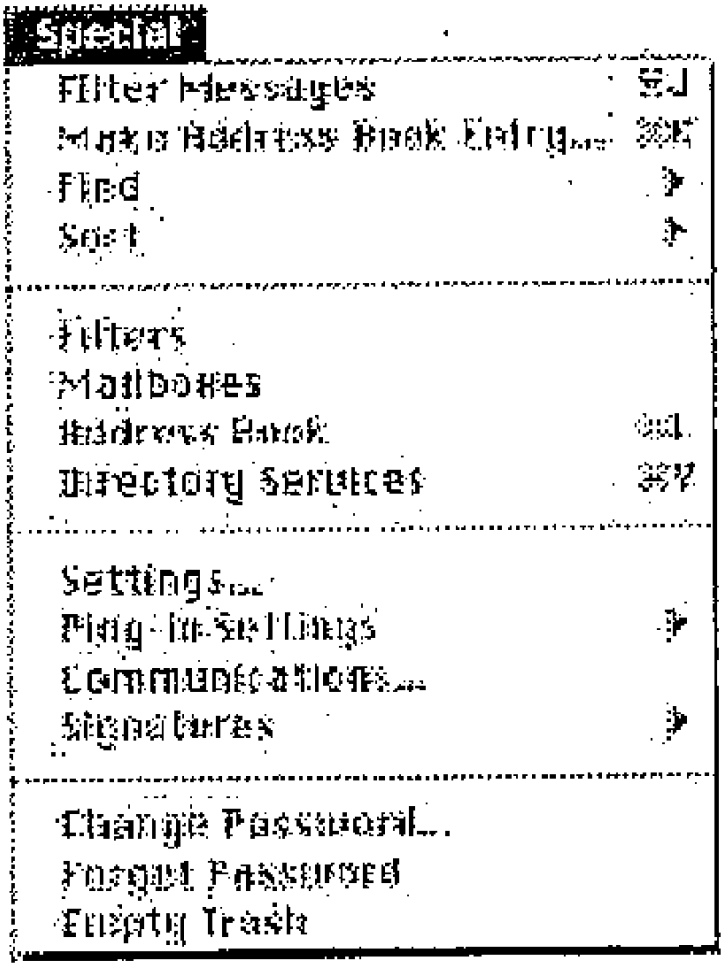
Finish Address Book Entry
Complete the partial text of a nickname.

Insert Recipient
Insert the chosen recipient"
Eudora Windows Manual at 137.

"**Special**

This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).
Eudora Mac Manual at 155.

"Special

This menu lets you use additional Eudora functions.

Exhibit K

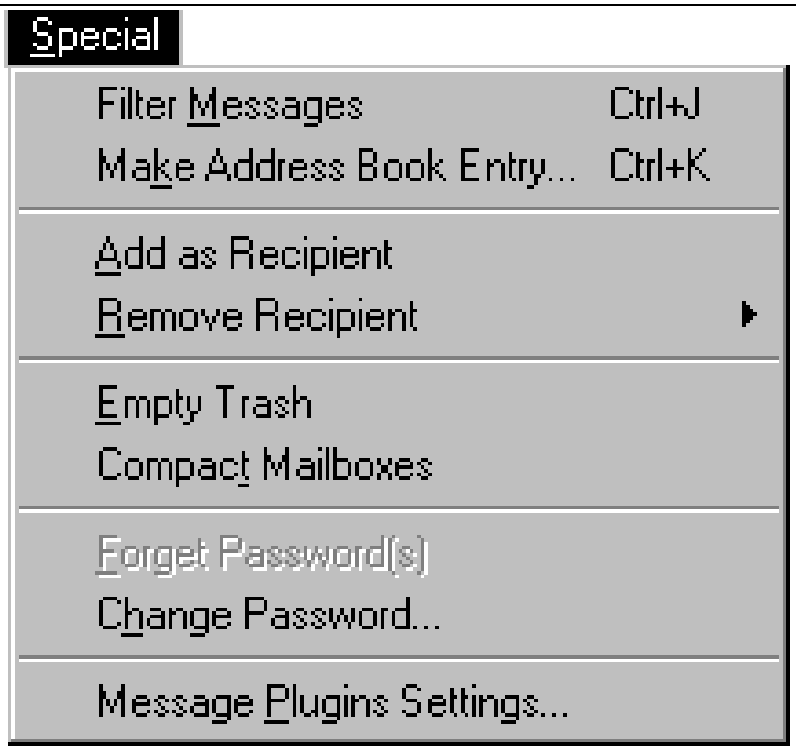
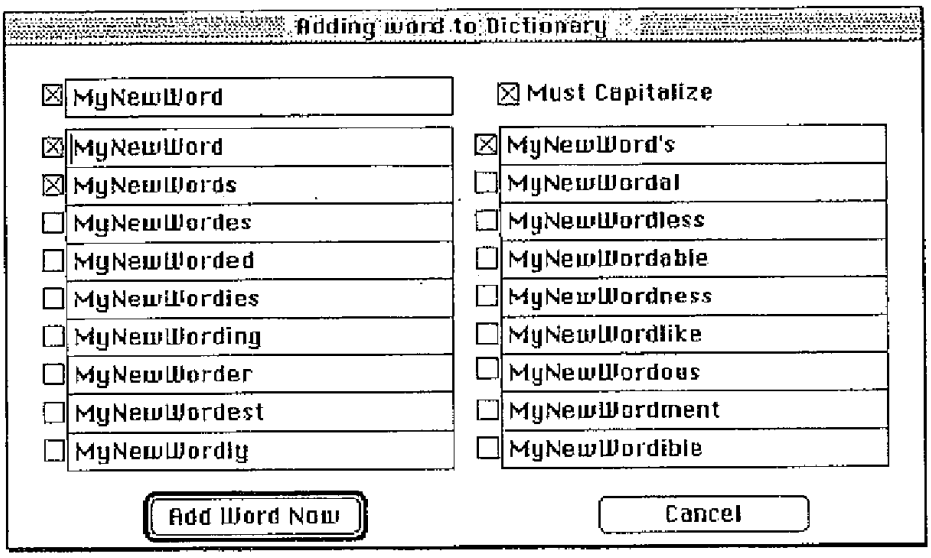
	 **Filter Messages** Run the manual filters for the current message(s). Eudora Windows Manual at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 12 and 17.
Claim 8	
A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.	Eudora discloses claim 1. *See* claim 1 above. Eudora further discloses this element. See, e.g.: Disclosure to Claim 1. "**Add** Add the questioned word to the dictionary. If the **Quick** option is on, then the questioned word is added to the dictionary immediately when you click this button. If this option is off, the **Adding word to Dictionary** dialog is displayed. This dialog provides you with options for adding the word and its various forms to the dictionary.

Exhibit K

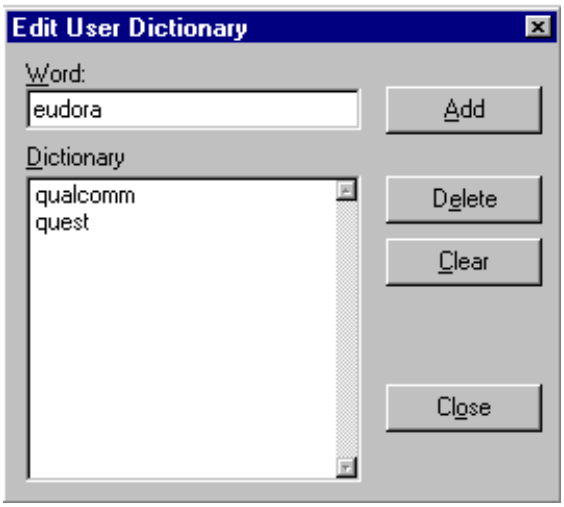


The Adding word to Dictionary dialog

Eudora Mac Manual at 45.

"**Add Button**
This button adds the unknown word to your custom user dictionary.

Edit Dictionary Button
This button displays the Edit User Dictionary dialog.



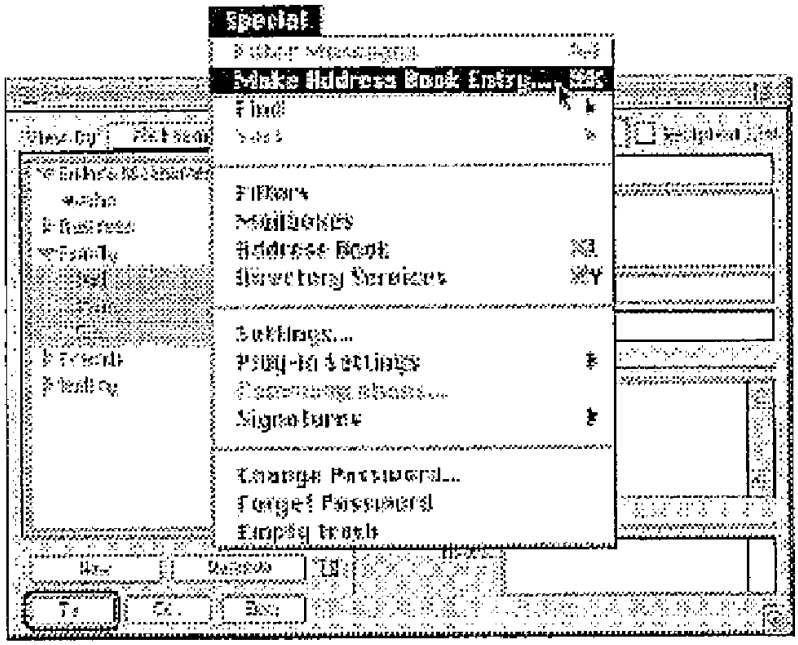
The Edit User Dictionary dialog

The Edit User Dictionary dialog lists all of the words in your user dictionary in the Dictionary field. It also allows you to add words to or delete words from your personal user dictionary, or even clear the entire

Exhibit K

	dictionary. *Note: Words in the user dictionary are saved in all lower case.* To add a word to the dictionary using this dialog, type the correct spelling of the word in the Word field and click the **Add** button. The word is then added to the dictionary and displayed in the Dictionary field. *Note: The Add button in this dialog works the same as the Add button in the Check Spelling dialog.*" Eudora Windows Manual at 36. "**The 'Make Address Book Entry' Command** The **Make Address Book Entry…** command is used to create entries in your Address Book, and is especially helpful for making group entries. You can use this command from anywhere in Eudora, including the Address Book, mailboxes, open messages, and the Directory Services window. From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the addresses that you selected. *Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.* In the Address Book, highlight several different entries (hold down the shift key to select multiple entries in sequence, or the command key to make disjoint selections), the select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.

Exhibit K



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message(s) you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry.

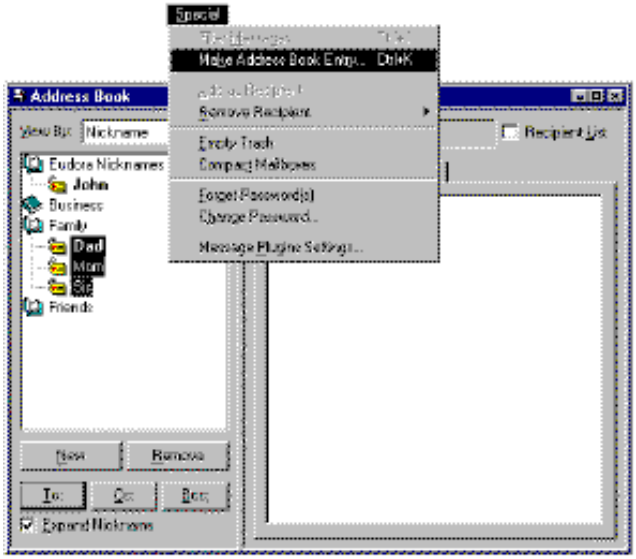
*Note: The **Make Address Book Entry…** command uses the Replying Settings. If the **Reply to all by Default** setting is turned on (or you hold down the option key), the new entry will include all of the recipients of the messages plus the sender. Or, if the **Include yourself** setting is turned off, your address is not included in the new entry.*

In an open message window, select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. See the note above about the Replying Settings.

Exhibit K

	In the Directory Services window, finish a Ph query and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry. **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings." Eudora Mac Manual at 99-101. **"The "Make Address Book Entry" Command** The Make Address Book entry command is used to create entries in your Address Book, and is especially helpful for making group entries. In the Address Book, highlight several different entries (hold down the Shift key to select multiple entries in sequence, or the Ctrl key to make disjoint selections), then select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.

Exhibit K



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message summaries you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry.

*Note: The **Make Address Book Entry** command uses the Reply Options. If the **Include yourself** option is on, your address is included in the new entry.*

In the Directory Services window, finish a Ph query, select the items that you want to include in the entry (or do not select anything to use all of the items), and select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry.

The "Finish Address Book Entry" Command

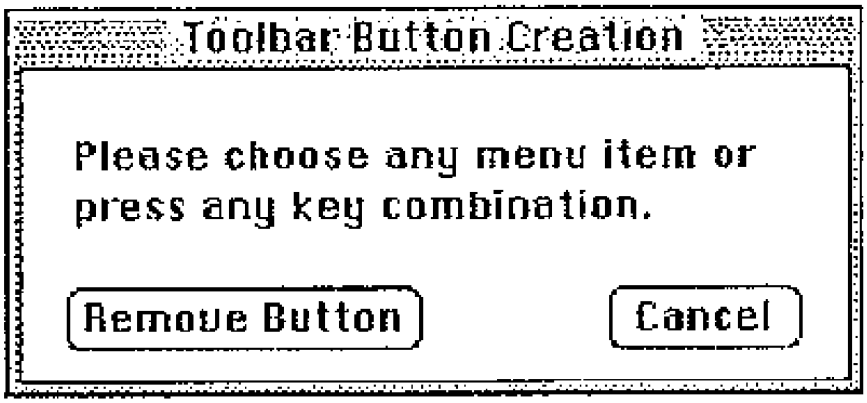
With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the

Exhibit K

nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them.

To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options."
Eudora Windows Manual at 87-89.

"To add a new button to the main toolbar, hold down the command key and put the cursor between two buttons or at either end of the toolbar, depending on where you want the new button to go; when the arrow changes to a splitter cursor [], click on it. The **Toolbar Button Creation** dialog is displayed prompting you to choose a menu item or enter a key combination. This can be almost anything you would do with Eudora, including using modifier keys with the command. When you are done, the button is added to the toolbar, and named appropriately. If you change your mind, click **Remove Button** or **Cancel**.



The Toolbar Button Creation dialog

You can also drag files from the Finder to the toolbar to add them as buttons, and you can drag mailboxes from the Mailboxes window to the toolbar to add them as mailbox buttons. (You cannot drag mail folders to the toolbar.)

To change what a button does, hold down the command key and click on the button, then select the function as you would when creating a button, or click **Cancel** if you change your mind."
Eudora Mac Manual at 107-08.

Exhibit K

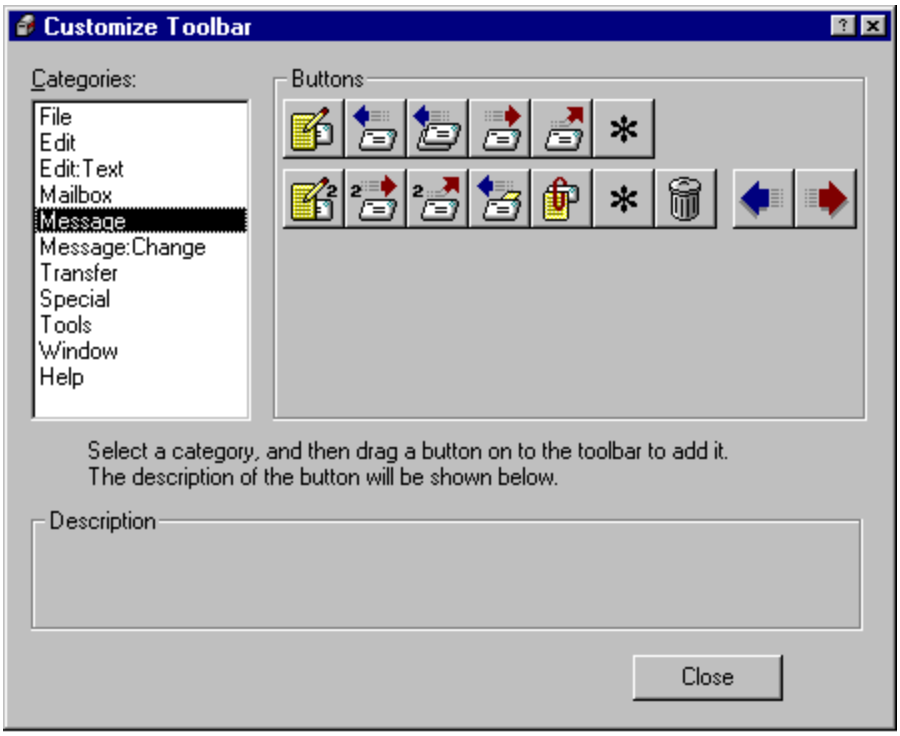
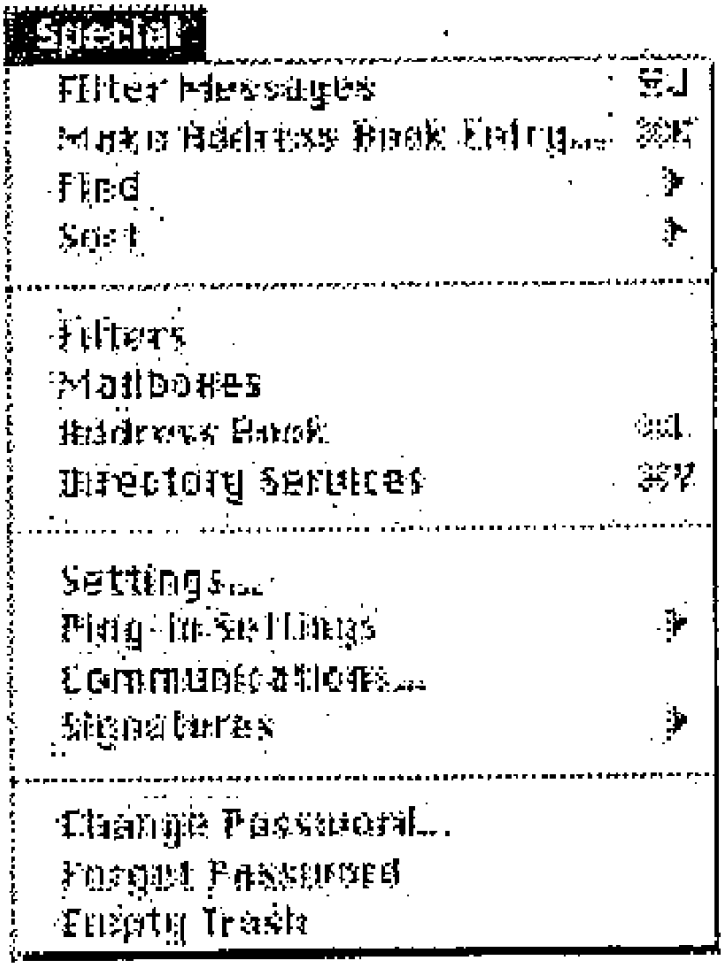
	"To add buttons to the 32-bit toolbar, right click on the toolbar, and select **Customize**. The **Customize Toolbar** dialog is displayed.  *The Customize Toolbar dialog* Select a Eudora menu from the list on the left, then drag buttons from the list on the right down to the toolbar. To change the placement of a button on the toolbar, hold down the **Alt** key and drag the button to where you want it. To remove a button, hold down the Alt key and drag it off of the toolbar. The 16-bit toolbar cannot be moved or changed." Eudora Windows Manual at 93-94. "**Special** This menu lets you use additional Eudora functions.

Exhibit K



Filter Messages
Run the manual filters for the current message(s).

Make Address Book Entry…
[shift] Make Address Book Entry From Selection…
Create an Address Book entry (nickname) from the current message.
Create an entry from the selected addresses.
Eudora Mac Manual at 155.

“Special

This menu lets you use additional Eudora functions.

Exhibit K

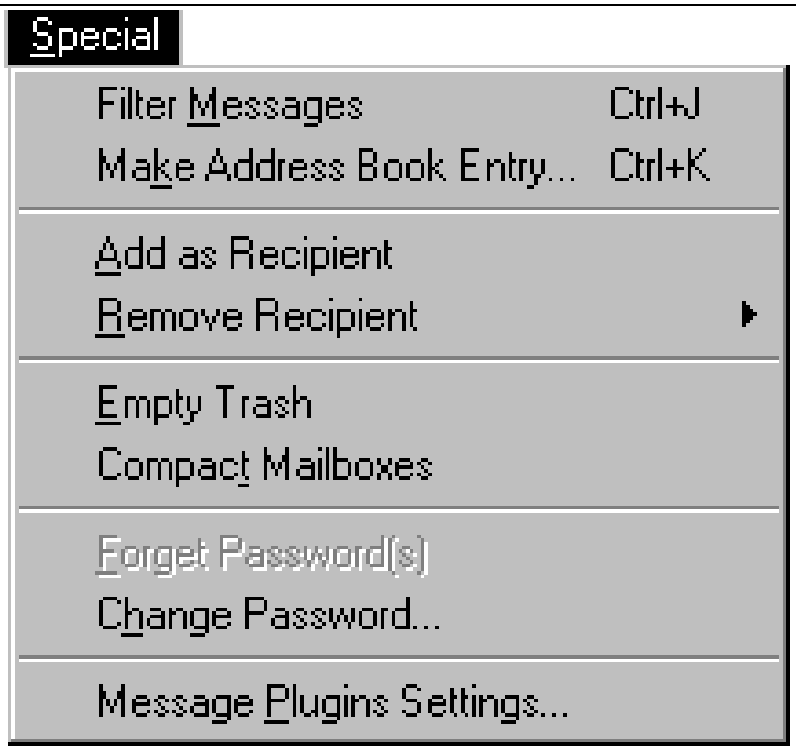
	 **Filter Messages** Run the manual filters for the current message(s). **Make Address Book Entry…** Create an Address Book entry from the current message. **Add As Recipient** Add selected text to the Quick Recipient list. Eudora Windows Manual at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Tables 4, 5, and 17.
Claim 13	
A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.	Eudora discloses claim 1. *See* claim 1 above. Eudora further discloses this element. See, e.g.: Disclosure to Claim 1. "**Checking Your Spelling**

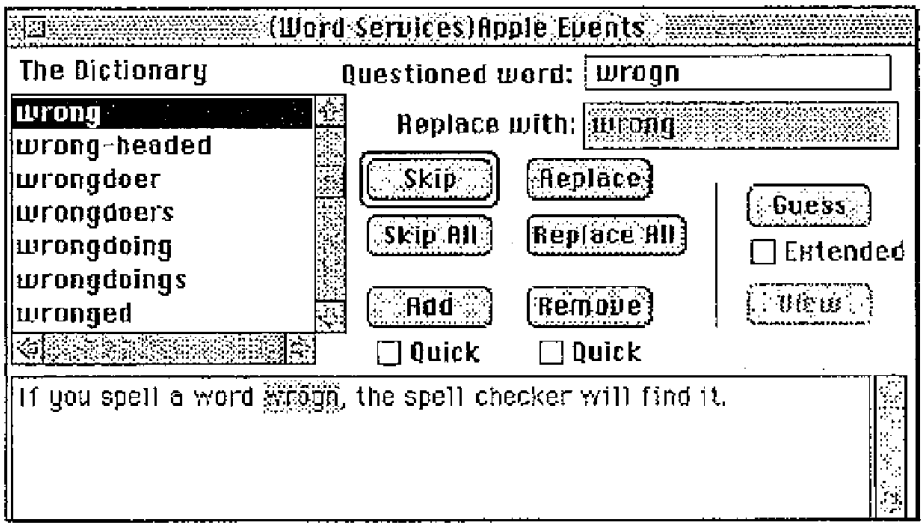
Exhibit K

******

To check your spelling in Eudora, select **Check Spelling** from the **Edit** menu. The spelling checker starts at the beginning of the document. The subject of the message and the message body are checked, ignoring the parts of the body that are identified as quoted text. You can also highlight a word or a block of text to check only that text and not the rest of the message.

If no misspellings are found, the spelling checker quits.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.



The Check Spelling dialog

The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check.

******

The Check Spelling Dialog

The Check Spelling dialog allows you to skip a questioned word, replace

	it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below. Questioned word A word that is not found in the spelling checker dictionary. Replace with Replace the questioned word with the word in this field. You can select a word from the Dictionary/Guesses field, or type a new one. Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. **Skip (All)** Ignore this occurrence of the questioned word. If you use **Skip All**, you ignore this and all subsequent occurrences of the questioned word. **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View** Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field.

Exhibit K

******

Spell Checking Options

******

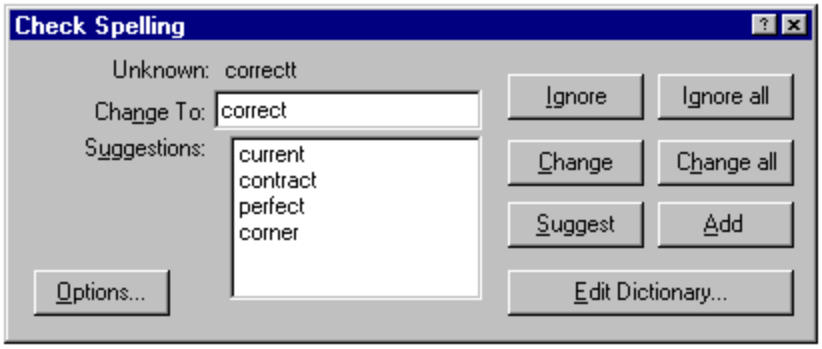
View suggestions instead of dictionary first
Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word.
Eudora Mac Manual at 42-47.

"To automatically check spelling when you send or queue a message, turn on the **Check when message queue/send selected** option in the Spell Checking Options. If this is on, when you send or queue a message the message is checked for spelling errors. If you go through the spell checking process, the message is automatically sent or queued. If you click Cancel, or leave spelling errors in the message, a dialog is displayed asking you if you still want to send or queue the message. If you don't want that dialog to be displayed, turn on the Don't warn me anymore option (this can also be set in the Spell Checking Options).

To check the spelling of a current composition window, text file, or signature file, click on the **Check Spelling** button in the main window toolbar or select **Check Spelling** from the **Edit** menu. If there are no misspellings, the No misspellings found alert is displayed.

Note: If text is selected, Eudora only checks the spelling of the selected text. Otherwise, it starts the spelling check from the beginning of the message body or text file and checks the entire text.

If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.



The Check Spelling dialog

To correct the misspelled word, type the correct spelling of the word in

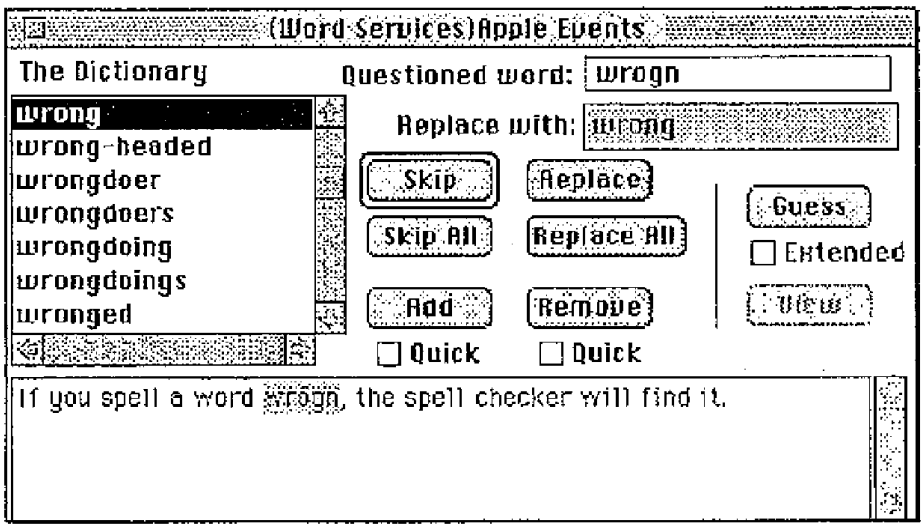
Exhibit K

	the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check. **Check Spelling Dialog** The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below. **Unknown Field** An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions. **Ignore Button** This button causes the spelling checker to ignore this occurrence of the unknown word. **Ignore all Button** This button causes the spelling checker to ignore this occurrence and all subsequent occurrences of the unknown word. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word.

Exhibit K

	Suggest Button This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* ****** **Make Suggestions** Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options,. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.*" Eudora Windows Manual at 34-37. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 2.
Claim 15	
A method according to claim 1, further comprising, if searching results in a plurality of distinct instances of second information, displaying such instances to enable user selection of one of them for use in performing the action.	Eudora discloses claim 1. *See* claim 1 above. Eudora further discloses this element. See, e.g.: Disclosure to Claim 1. "If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the **Questioned word** field. The word is also highlighted in context at the bottom of the window.

Exhibit K



The Check Spelling dialog

The **Replace with** field displays the dictionary entry alphabetically closest to the questioned word. If this suggestion is not acceptable, you can change it by clicking on a word from the list. Or, you can type the correct spelling of the word directly in the **Replace with** field. Once the **Replace with** field contains the correct entry, click the **Replace** button. The word in the document is replaced with the word in the **Replace with** field. The spelling checker then proceeds with the check.

Note: If you select **Check Spelling** from the **Edit** menu and an error dialog appears telling you that the application (Spellswell cannot be found, it means that the link between Eudora and Spellswell has somehow been broken and must be reestablished. This is rare, but easy to fix. Take these steps: Select **Add Word Service…** from the **Edit** menu. Browse through the dialog and find the Spellswell application on your hard drive. Double-click on the application, or click once on it to highlight it and then click **Open**. Now Spellswell is available again: you can select **Check Spelling** from the **Edit** menu to spell-check your messages.

The Check Spelling Dialog

The Check Spelling dialog allows you to skip a questioned word, replace it, guess the correct spelling, and add or delete the word to or from your user dictionary. Each of the fields and buttons is described below.

Questioned word
A word that is not found in the spelling checker dictionary.

Replace with

Exhibit K

	Replace the questioned word with the word in this field. You can select a word from the Dictionary/Guesses field, or type a new one. Dictionary/Guesses This field is labeled **Dictionary** when the **View suggestions instead of dictionary first** option is off (the default), and **Guesses** when it is on. (See the section "Spell Checking Options.") **Dictionary** lists all words that are alphabetically similar to the questioned word. To display the spelling checker's suggestions for the correct spelling, click on the **Guess** button. **Guesses** automatically lists all suggestions for the correct spelling. **Skip (All)** Ignore this occurrence of the questioned word. If you use **Skip All**, you ignore this and all subsequent occurrences of the questioned word. **Replace (All)** Replace this occurrence of the questioned word with the word in the **Replace with** field. If you use **Replace All**, you replace this and all subsequent occurrences of the questioned word. **Guess** Display the spelling checker's suggestions for the correct spelling of the questioned word. If the **Extended** option is checked, the spelling checker displays more possible choices for the questioned word (an extended guess takes longer than a regular guess). *Note: You can make a wild card guess if you type some letters followed by ? in the **Replace with** field and then press the **Guess** button. The more specific you are, the faster the search will be.* **View** Display the dictionary list of words that are alphabetically similar to the questioned word. The **View** button is actively only when Guesses are being displayed in the **Guesses** field. ****** **View suggestions instead of dictionary first** Displays in the **Guesses** field the spelling checker's suggestions for the correct spelling of the questioned word. **HTML checking - ignore between brackets: '<>'"**

Exhibit K

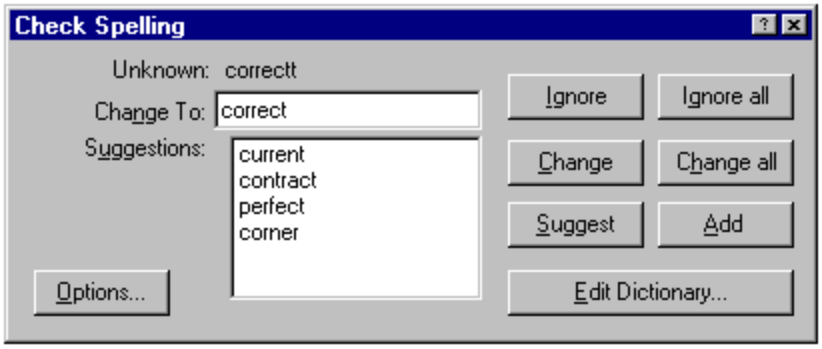
	Eudora Mac Manual at 43-47. "If a misspelled, unknown, or repeated word is found, the Check Spelling dialog is displayed with the word listed in the Unknown field.  *The Check Spelling dialog* To correct the misspelled word, type the correct spelling of the word in the Change To field, select it from Suggestions list and click the **Change** button, or double-click it in the Suggestions list. The spelling checker then proceeds with the check. **Check Spelling Dialog** The Check Spelling dialog allows you to ignore an unknown word, change it, suggest the correct spelling, add the word to your user dictionary, edit your dictionary, or change the spell checking preferences via the Options button. Each of the fields and buttons is described below. **Unknown Field** An unknown word is one that is not found in Eudora's built-in dictionary or your own custom dictionary. You can act on an unknown word using the Ignore, Ignore all, Change, Change all, or Add buttons, as described below. **Change To Field** This field works in conjunction with the Change and Change all buttons. It allows you to modify the unknown word by typing its correct spelling in this field, or selecting a suggested alternative spelling from the Suggestions field, and then clicking the Change or Change all buttons, as described below. **Suggestions Field** This field lists Eudora's suggestions for the correct spelling of the unknown word. If the Always Suggest option is turned on, all suggestions are listed here by default. If this option is turned off, click the Suggest button to display Eudora's suggestions.

Exhibit K

	Ignore Button This button causes the spelling checker to ignore this occurrence of the unknown word. **Ignore all Button** This button causes the spelling checker to ignore this occurrence and all subsequent occurrences of the unknown word. **Change Button** This button substitutes to contents of the Change To field for the unknown word. **Change all Button** This button substitutes to contents of the Change To field for the unknown word, and all subsequent occurrences of the unknown word. **Suggest Button** This button displays Eudora's suggestions for the correct spelling of the unknown word in the Suggestions field. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed.* **Add Button** This button adds the unknown word to your custom user dictionary. ****** **Make Suggestions** Displays Eudora's suggestions for the correct spelling of an unknown word. You can select any combination of the suggestion options,. *Note: If Eudora doesn't have suggestions in its dictionary, then none are listed."* Eudora Windows Manual at 34-37. "**The 'Make Address Book Entry' Command** The **Make Address Book Entry…** command is used to create entries in your Address Book, and is especially helpful for making group entries. You can use this command from anywhere in Eudora, including the Address Book, mailboxes, open messages, and the Directory Services window.

Exhibit K

	From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make Address Book Entry From Selection…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the addresses that you selected. *Note: If the new nickname has the same name as an existing nickname, a prompt is displayed asking if you want to add the selected names to the existing nickname or replace the existing nickname with the new selection.* ****** **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. Eudora Mac Manual at 99-101. **"The "Finish Address Book Entry" Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. Eudora Windows Manual at 89 "**Finish Address Book Entry** **[option] Finish & Expand Address Book Entry** Complete the partial text of a nickname. Complete it and expand it to its real address. **Insert Recipient** **[option] Insert & Expand Recipient** Insert the chosen nickname. Insert the real address of the nickname." Eudora Mac Manual at 150.

Exhibit K

	"**Finish Address Book Entry** Complete the partial text of a nickname. **Insert Recipient** Insert the chosen recipient" Eudora Windows Manual at 137. "**Filter Messages** Run the manual filters for the current message(s). **Make Address Book Entry…** **[shift] Make Address Book Entry From Selection…** Create an Address Book entry (nickname) from the current message. Create an entry from the selected addresses." Eudora Mac Manual at 155. "**Filter Messages** Run the manual filters for the current message(s). **Make Address Book Entry…** Create an Address Book entry from the current message." Eudora Windows Manual at 141. For example (and without limitation to the Obviousness Statement that is incorporated into each element in this chart), this element is rendered obvious for the reasons stated in Exhibit U, Table 7, 17, and 20.
Claim 17	
A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.	Eudora discloses claim 1. *See* claim 1 above. Eudora further discloses this element. See, e.g.: Disclosure to Claim 1. "**Inserting the Contents of a Text File into a Message** The contents of a text file can be inserted directly into a message (and then edited if desired). To insert a text file into a message, put the cursor where you want the text inserted, and select Attach Document… from the Message menu. Then select the text file you want and click on the Insert button. The text from the file is inserted into your message and you can edit it as normal." Eudora Mac Manual at 42.

Exhibit K

	"**Checking Your Spelling** Eudora includes the Spellswell 7 Spelling Checker, developed by Working Software, Inc. Because it is an Apple Events Word Services Suite application, Spellswell 7 can be used with Eudora. This section describes the spelling checker's basic functions when it is used with Eudora. For more information on Spellswell 7, how it functions with other applications, specialized dictionaries, etc., see the Spellswell 7 User Manual. It is located in the Documentation folder within the Eudora Pro Application Folder. The spelling checker includes a customizable 93,000+ word dictionary. It can be used to check for spelling mistakes and typographical errors in message composition windows, text files, and signature files. Eudora Mac Manual at 42-43. **"Checking Your Spelling** Eudora includes a built-in spelling checker. It can be used to check for misspellings in the body of current message composition windows, text files, and signature files. It includes a built-in dictionary and also allows for the creation of a custom user dictionary. Additionally, it can be configured to ignore capitalized words, words with numbers, and mixed case words, to report mixed case and double (repeated) words, and to suggest alternative spellings. Eudora Windows Manual at 33. "Included with Eudora Pro[TM] 3.0 is the spell check application, **Spellswell** from **Working Software**, which performs simple checks for grammar and spelling errors. The 93,000+ word dictionary is considerably robust for its size, has an intuitive interface, and is user-expandable. In a world where spelling errors are an embarrassment, this feature is a welcome addition." Eudora Review at 4. "**Filtering Messages** Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as "Hot."

Exhibit K

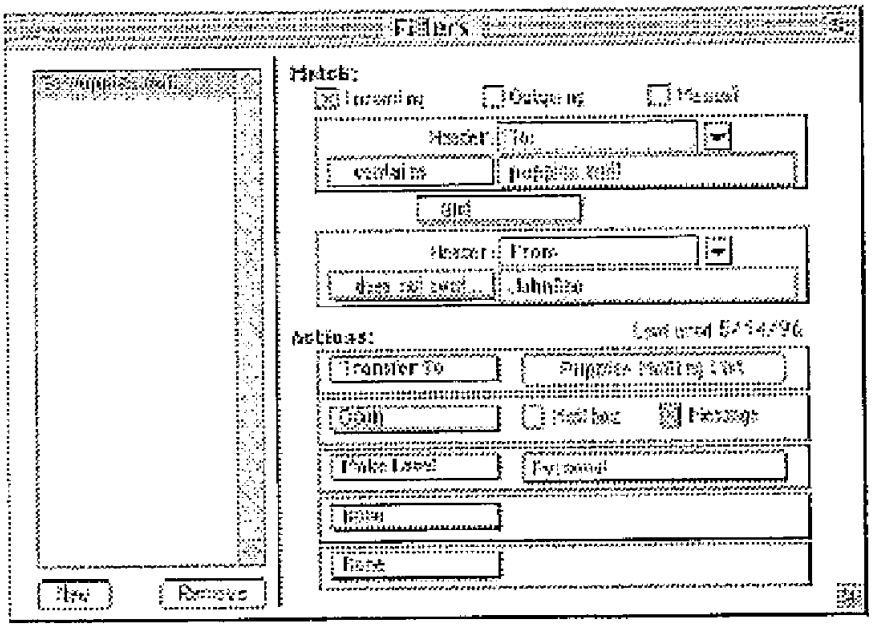
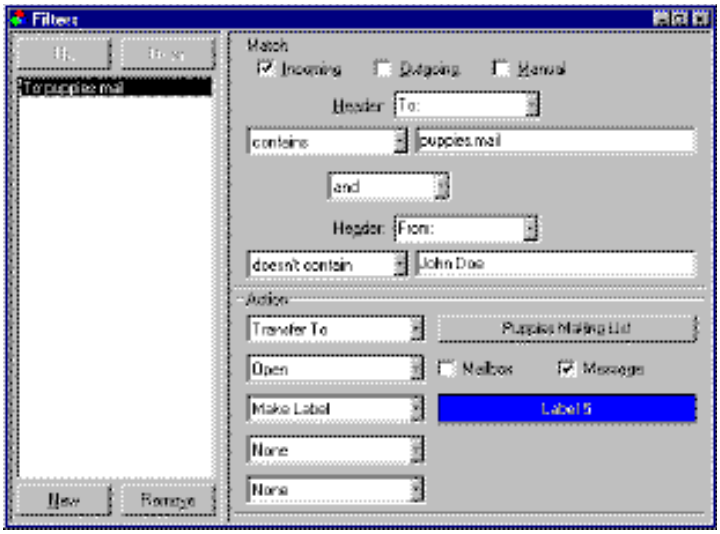
	To open the Filters window, select **Filters** from the **Special** menu. The Filters window is displayed, and any filters you have created are listed on the left.  *The Filters window with a sample filter* To create or modify a filter, first click on the **New** button or select an existing filter. Second, select the options for how you want the filter to be used: as an automatic filter to be invoked on any **Incoming** and/or **Outgoing** mail, and as a **Manual** filter that can be invoked when you select **Filter Messages** from the **Special** menu. Any combination of these options works. Third, define the criteria for the filter: use the header item popups and the text fields to specify which header items should include a particular string of text. You can define two related terms for the criteria so that your filter is as specific as possible (see the section 'Filter Criteria'). Fourth, define the action to be taken on messages that fit the criteria (see the section 'Filter Actions') and save the filters. When the filters are invoked (automatically or manually), each message is matched against each filter, and messages that meet a filter's criteria are acted on as specified until a **Transfer To** or **Skip Rest** action is done. At that point, the next message is filtered. Eudora Mac Manual at 82-83.

Exhibit K

“Filtering Messages

Many of the e-mail management functions in Eudora can be done automatically using filters. For example, you can automatically reply to a request for information, transfer all the messages from your children into a Personal mailbox, and label all the messages from your customers as “Hot.”

To open the Filters window, select **Filters** from the **Tools** menu. The Filters window is displayed, and any filters you have created are listed on the left.



The Filters window with an example filter

To create or modify a filter, first click on the **New** button or select an existing filter.

Second, select the options for how you want the filter to be used: as an automatic filter to be invoked on any **Incoming** and/or **Outgoing** mail, and as a **Manual** filter that can be invoked when you select **Filter Messages** from the **Special** menu. Any combination of these options works.

Third, define the criteria for the filter: use the header item popups and the text fields to specify which header items should include a particular string of text. You can define two related terms for the criteria so that your filter is as specific as possible (see the section ‘Filter Criteria’).

Fourth, define the action to be taken on messages that fit the criteria (see

Exhibit K

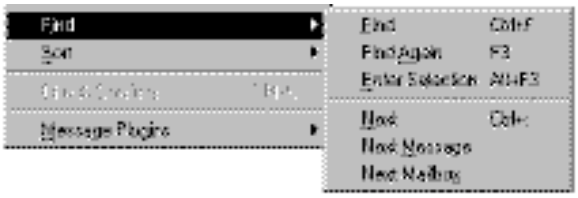
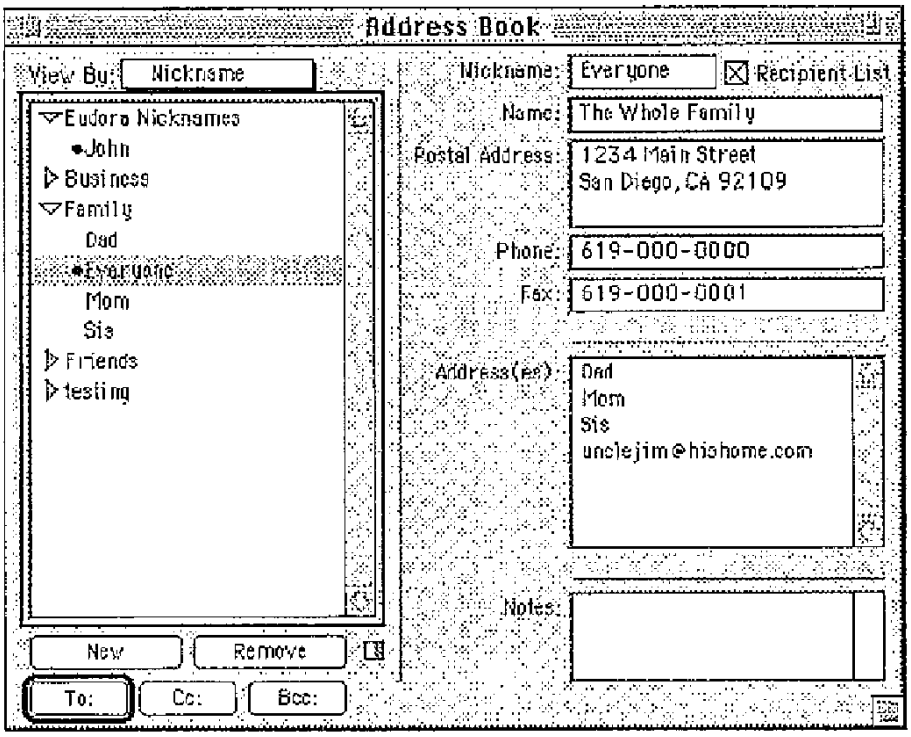
	the section 'Filter Actions') and save the filters. Eudora Windows Manual at 72-73. "**Finding Text Within Messages** You can find a word or a string of text anywhere in your Eudora messages, your Address Book, or your Filter. To do this, select **Find** from the **Special** menu, and **Find…** from the submenu. The **Find** dialog is displayed.  Enter the word or string of text that you want to find in the **Find** field. Or, if you don't want to type in the text, you can highlight the text in an existing message, then select **Enter Selection** from the **Find** submenu. The selected text is automatically inserted in the **Find** field of the **Find** dialog. Eudora Mac Manual at 89-90. **"Finding Text Within Messages** Eudora incorporates a Find function that searches for specific text within a single message, multiple messages, or even multiple mailboxes. To display the Find submenu of commands, select **Find** from the **Edit** menu.  *The Find submenu* **Finding Text Within One Message** To search for text within a single message, open the message and make sure it is current. Then, select **Find** from the **Edit** menu and select the **Find** command from the submenu. The Find dialog is displayed, with the blinking insertion point located in the text field. Type the text you want to find in the text field. When finished entering

Exhibit K

	the\ desired text, click the **Find** button.  *Finding text* Starting at where the cursor is in the message, Eudora searches the current message for the specified text. If no match is found, the not found alert is displayed. If the search is successful, the message is scrolled to the first point where the match is found and the matching text is highlighted. To continue searching in the same message for the next occurrence of the text, click the **Find** button in the Find dialog, or select the **Find Again** command from the **Find** submenu. These commands are equivalent and limit the search to the same message. Repeating these commands cycles through the matches in the open message only. ***** **Enter Selection Command** If you don't want to actually type the text in the Find dialog (for example, the text is very long or complex), highlight it in an existing message, and then select **Enter Selection** from the **Find** submenu. This automatically inserts the selected text at the insertion point in the Find dialog. Then, select the **Find** command from the **Find** submenu to start the search. Eudora Windows Manual at 78 and 80. "Eudora Pro[TM] 3//0 has a powerful 'Quick Search Engine' which allows you to use a VCR-like interface to find any particular text string in any message or mailbox." Eudora Review at 5. "**Using the Address Book** The Address Book is where you keep information about individuals or groups that you correspond with. Each entry in the Address Book includes a nickname for a person or group, their full e-mail addresses, a real name, any contact information, and any notes. You can also use the Address Book to put nicknames on the Quick Recipient List, and to address a new message.

Exhibit K

To open your Address Book, select **Address Book** from the **Special** menu.



The Address Book with example entries

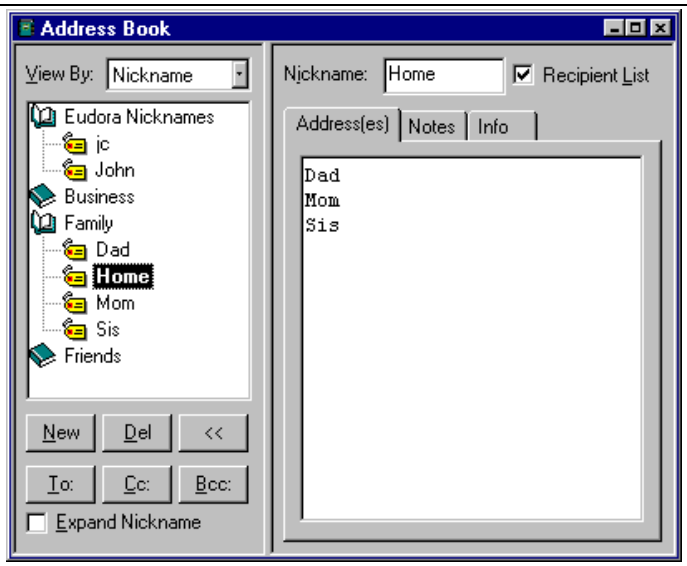
All of the Address Book entries are kept in files, so you can group your entries by putting them in different files. The example above shows files for Business, Family, and Friends (Eudora Nicknames is the default file). To show the entries in a file, click on the arrow to the left of the file. When the arrow points down, all the entries for that file are displayed." Eudora Mac Manual at 95-96.

"Using the Address Book

The Address Book is where you keep information about individuals or groups that you correspond with. Each entry in the Address Book includes a nickname for a person or group, their full e-mail addresses, a real name, any contact information, and any notes. You can also use the Address Book to put nicknames on the Quick Recipient List, and to address a new message.

To open your Address Book, select **Address Book** from the **Tools** menu."

Exhibit K



The 32-bit Address Book with example entries

All of the Address Book entries are kept in files. The example above shows files for Business, Family, and Friends (Eudora Nicknames is the default file). In the 32-bit Address Book, you can show or hide the entries in a file by double-clicking on the file. The icon shows an open or closed book, depending on whether the file is open or closed. In the 16-bit Address Book, files are flush to the left, and their entries are listed under them.
Eudora Windows Manual at 83-84.

"An adjunct to the 'Filter' system is a more powerful 'Address Book,' which stores addressee information. E-mail addresses, street address, phone numbers and incidental notations concerning your correspondents are easily stored and accessed. You may also initiate e-mail messages directly from the Address Book, either to individuals or groups."
Eudora Review at 6.

"**The 'Make Address Book Entry' Command**

The **Make Address Book Entry…** command is used to create entries in your Address Book, and is especially helpful for making group entries. You can use this command from anywhere in Eudora, including the Address Book, mailboxes, open messages, and the Directory Services window.

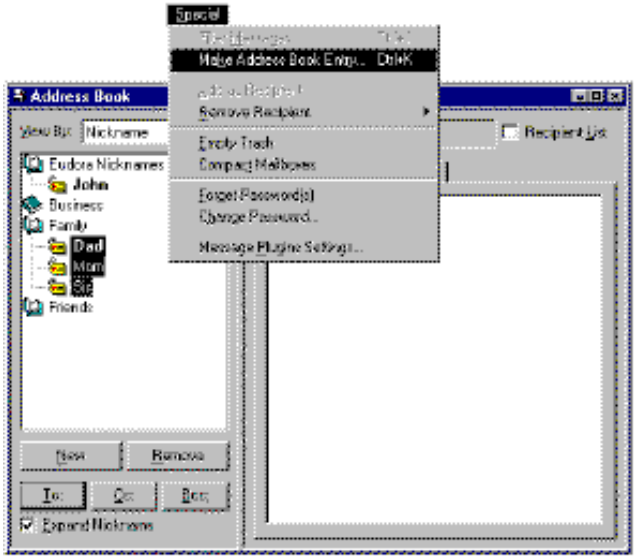
From anywhere in Eudora, including open messages, you can highlight the addresses you want, then hold down the shift key and select **Make**

Exhibit K

	Address Book Entry From Selection… from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. The new entry's **Address(es)** field will include all of the addresses that you selected. ****** **The 'Finish Address Book Entry' Command** With the **Finish Address Book Entry** command, you can enter a unique portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, joan and john, you would have to enter 'joa' or 'joh' for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the option key and select **Finish & Expand Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail settings. **Using Nicknames that were Not Created by Eudora** To use a nickname file that was not created in Eudora, put the file in the Nicknames Folder in your Eudora Folder (located in your System Folder), and be sure the format is correct: One nickname on each line with the word 'alias,' a space, the nickname, a space, and the real addresses separated by commas. For example alias joe joe@wow.com alias group joe@wow.com,lisa@wow.com,bill@wow.com You will need to quit and restart Eudora to see your new entries in the Address Book." Eudora Mac Manual at 99-101. **"The "Make Address Book Entry" Command** The Make Address Book entry command is used to create entries in your Address Book, and is especially helpful for making group entries. In the Address Book, highlight several different entries (hold down the Shift key to select multiple entries in sequence, or the Ctrl key to make disjoint selections), then select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed prompting you for

Exhibit K

the nickname of the new entry. The **Address(es)** field of the new entry will include the nicknames for the entries you selected, not the real addresses.



Using the "Make Address Book Entry" command from the Address Book

In a mailbox, highlight the message summaries you want and select **Make Address Book Entry…** from the **Special** menu. The New Nickname dialog is displayed prompting you for the nickname of the new entry. Follow the instructions for creating a new entry. If the current message is an outgoing message, the new entry will include all of the addresses in the **To**, **Cc**, and **Bcc** fields. If the current message is an incoming message, the new entry will include the address in the **From** field. If multiple messages are current (i.e., you have several message summaries selected in a mailbox window), addresses are taken from each message and are all put in the new entry.

*Note: The **Make Address Book Entry** command uses the Reply Options. If the **Include yourself** option is on, your address is included in the new entry.*

In the Directory Services window, finish a Ph query, select the items that you want to include in the entry (or do not select anything to use all of the items), and select **Make Address Book Entry...** from the **Special** menu. The New Nickname dialog is displayed so that you can name the nickname. The real name and e-mail address are included in the new entry.

The "Finish Address Book Entry" Command

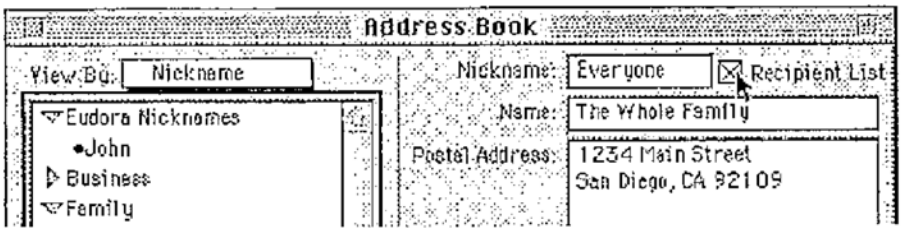
With the **Finish Address Book Entry** command, you can enter a unique

Exhibit K

	portion of a nickname in the **To**, **Cc**, or **Bcc** fields of a message, then select **Finish Address Book Entry** from the **Edit** menu, and the nickname will be completed for you. You must enter the characters in the nickname that make it unique, or Eudora will not know which nickname to use. For example, if you have two nicknames, jon and john, you would have to enter "jon" or "joh" for Eudora to complete them. To insert the real addresses for the entry, instead of the nickname, hold down the Shift key and select **Finish Address Book Entry** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options. **Using Central Address Book Files on a Server** You can set up central Address Book files on a server and configure Eudora clients so that they refer to the central files. First, be sure the files are plain text, have a .**txt** extension, and are formatted as follows: One nickname on each line with the real addresses separated by commas, and one line for notes and info with the **Notes** text following the **Info** data. For example: alias Wow joe@wow.com,lisa@wow.com,chris@wow.com note Wow <fax: 222.2223><phone: 222.2222><address:1234 Street> <name:Wow Inc.>My favorite company Then, for each client application, add a **ExtraNicknameDirs** entry to the [Settings] section of the EUDORA.INI file. This entry should be followed by the list of directories that contain Address Book files, separated by semicolons (;). Any Address Book files located in those directories are added to the Address Book. Users will need to exit and re-open Eudora to see the new entries. **Using Address Book Files Not Created by Eudora** To use a Address Book file that was not created in Eudora, put the file in the Nickname directory (in your Eudora directory), and be sure the format is as shown in the section "Using Central Address Book Files on a Server." You will need to exit and reopen Eudora to see your new entries in the Address Book." Eudora Windows Manual at 87-90. "**Using the Quick Recipient List** The Quick Recipient List is your list of often-used nicknames. If you have

Exhibit K

checked the **Recipient List** option in an Address Book entry, the entry's nickname is included in the list.



The Recipient List option

To open a new message address to someone on your Quick Recipient List, select **New Message To, Forward to**, or **Redirect To** from the **Message** menu, and select the nickname from the displayed list.

To insert a nickname into a message that you have already opened, put the cursor where you want the nickname and select **Insert Recipient** from the **Edit** menu.

To insert the real address(es), instead of the nickname, hold down the option key and select **Insert & Expand Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Expand nicknames immediately** option in the Sending Mail Settings."
Eudora Mac Manual at 102.

"Using the Quick Recipient List

The Quick Recipient List is your list of recipients to whom you often send mail.

To add a nickname to the Quick Recipient List, check the **Recipient List** option in its Address Book entry.

To add an e-mail address to the Quick Recipient list, select the text that makes up the full address. Then, select **Add As Recipient** from the **Special** menu.

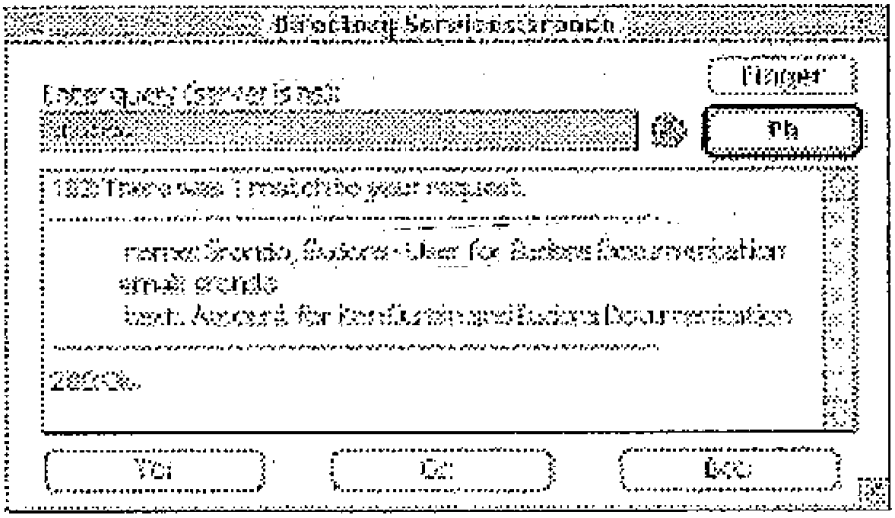
To remove an entry from the list, uncheck the Recipient List option in the Address Book entry, or select the nickname from the **Remove Recipient** submenu from the **Special** menu.

To open a new message addressed to someone on your Quick Recipient List, select **New Message To, Forward To**, or **Redirect To** from the **Message** menu, and select the nickname from the displayed list.

Exhibit K

	To insert a recipient into a message that you have already opened, put the cursor where you want the recipient and select **Insert Recipient** from the **Edit** menu. To insert the real address(es), instead of a nickname, hold down the Shift key and select **Insert Recipient** from the **Edit** menu. To set this to happen all the time, turn on the **Automatically Expand Nicknames** option in the Miscellaneous Options. More than one recipient from the Quick Recipient List can be added to the **To**, **Cc**, and **Bcc** fields of any message. If you use the Insert Recipient command, commas are added where necessary." *Eudora Windows Manual* at 90. "**Using Directory Services** Opening Directory Services Eudora can access two different online directory services, **Ph** and **Finger**. To use these services, you must put the name of the host machines for the Ph and Finger servers in the Hosts Settings. To use the directory Services, select **Directory Services** from the **Special** menu. The active Ph or Finger server (defined in your Hosts Settings) is displayed above the query field. Or, you can select a string of text (someone's name or e-mail address, for example), hold down the shift key, and select **Directory Services** from the **Special** menu. This opens the window and inserts the string of text into the query field. Using Ph To look someone up using Ph, enter your query and click on **Ph**. The query is sent to your Ph server, and the response is displayed in the lower section of the window.

Exhibit K



A Ph query and its response

Note: You can type any Ph command in the query field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

If the **'Live' Ph queries** option is on in your Hosts Settings, the connection with your server is kept open and Ph queries are automatically sent to the server when you finish typing the query text.

*Note: To add the results of your Ph query to your Address Book, select **Make Address Book Entry…** from the **Special** menu (for details on how to use this command, see the section 'The "Make Address Book Entry" Command'). This may not work if your Ph server is not set up for it.*

Finding Ph Servers

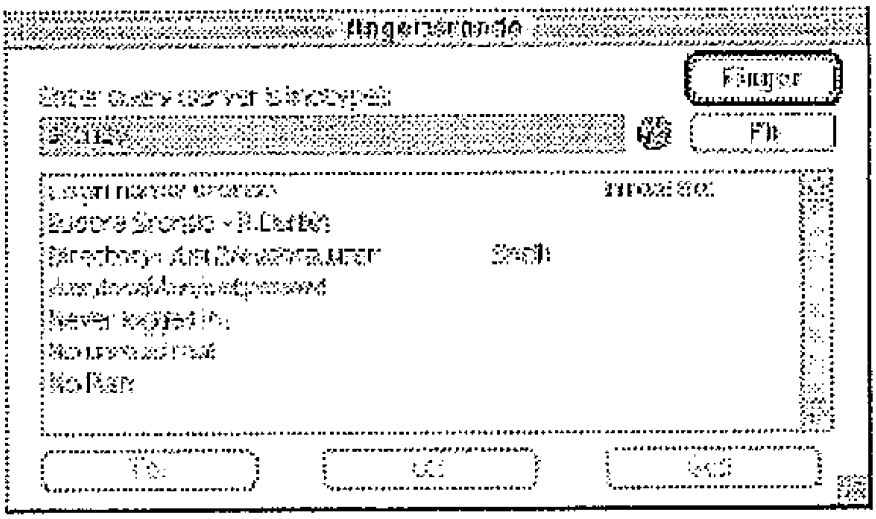
Some Ph servers keep a list of other Ph servers that are available on the Internet. This is not always a comprehensive list of every Ph server out there, but it can be helpful.

To get the list of servers from the active server (the active server is displayed above the query field), click on the globe in the Directory Services window (new to the Ph button). A list of servers is displayed in the results area. To go to one of those servers and do a query, hold down the command key and click on the server's URL, or just double-click on the URL.

Using Finger

Exhibit K

To use the Finger protocol, enter your query and click **Finger**. The query should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the query field is used (this is the SMTP host from your Hosts Settings). The Finger query is sent to the Finger server, and the response is displayed in the lower section of the window."



A Finger query and its response

Addressing a Message from the Directory Services Window

You can create and address a message with the query results in the Directory Services window.

To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results.

To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finder query, then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message.

Or, you can select the e-mail address from the results and drag it into the appropriate field of the outgoing message.
Eudora Mac Manual at 103-05.

"**Using Directory Services**

Exhibit K

Opening Directory Services

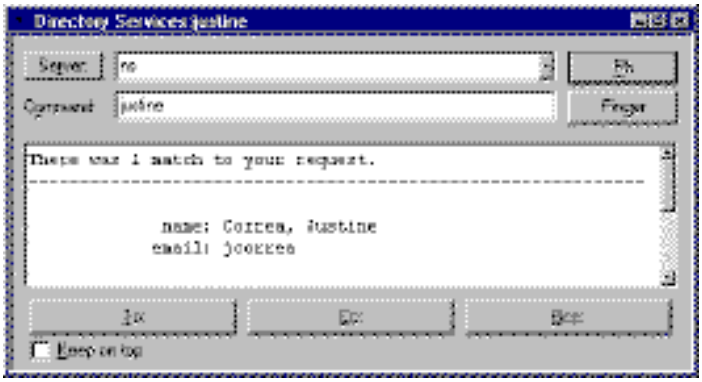
Eudora can access two different online directory services, **Ph** and **Finger**. To use these services, you must put the name of the host machines for the Ph and Finger servers in the Hosts Options. The active Ph or finger server (defined in your Host Options) is displayed in the Server field, which includes a popup list of the last 10 servers you have accessed.

To use the directory Services, select **Directory Services** from the **Tools** menu.

Using Ph

To look someone up using Ph, enter the information (usually someone's name) into the command field and click on **Ph**. The command is sent to your Ph server, and the response is displayed in the lower section of the window.

If there is more than one match, you can use the Tab key to move down to the next one, or hold down the Shift key and press Tab to move up to the previous one.



A Ph command and its response

Note: You can type any Ph command in the command field, except login commands or commands requiring login. For information about the Ph server source code, see Appendix A.

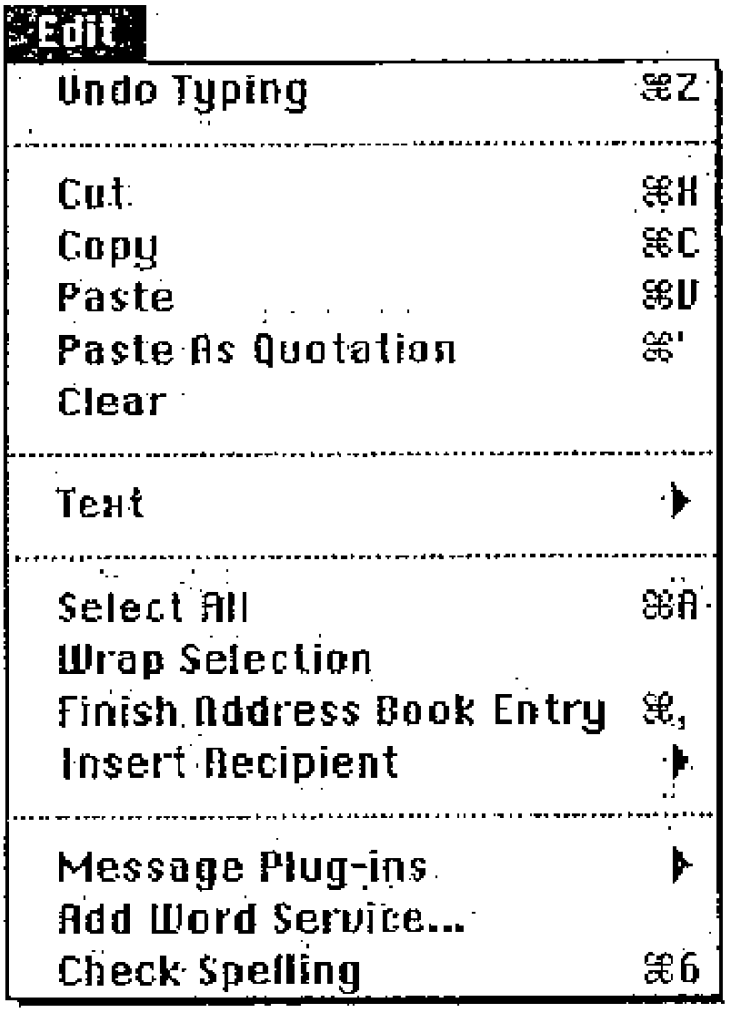
Finding Ph Servers

Some Ph servers keep a list of other Ph servers that are available on the Internet. This is not always a comprehensive list of every Ph server out there, but it can be helpful.

Exhibit K

	To get the list of servers that the active server knows about, click on the Server button in the Directory Services window. A list of servers is displayed in the results area. To go to one of those servers and do a query, double-click on the server's URL. **Using Finger** To use the **Finger** protocol, enter your query and click **Finger**. The command should be in the form 'name@domain.' If you omit the '@domain' segment, the host name displayed above the Server Field is used. The finger command is sent to the finger server, and the response is displayed in the lower section of the window. **Addressing a Message from the Directory Services Window** You can create and address a message with the command results in the Directory Services window. To create a new message, be sure there are no outgoing messages already open, do the Ph or Finder command, and use the Tab key to select the right address (if there is more than one). Then click on the **To**, **Cc** or **Bcc** button. A new message is created, and addressed appropriately with the query results. To add an address to an existing message, make sure the message you want to address is active, open the Directory Services window, do the Ph or Finger command, and use the Tab key to select the right address. Then click on the **To**, **Cc** or **Bcc** button. The address from the query result is added to the appropriate field of the current message." Eudora Windows Manual at 91-92. "**Edit** This menu provides text editing tools.

Exhibit K



* * *

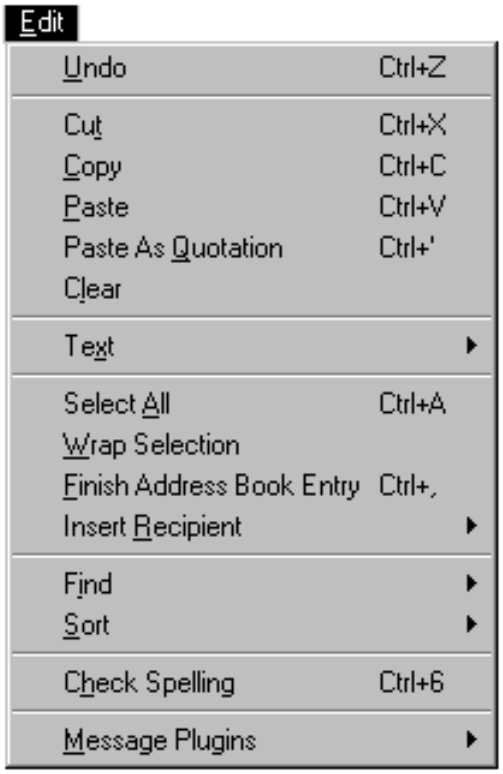
Finish Address Book Entry
[option] Finish & Expand Address Book Entry
Complete the partial text of a nickname.
Complete it and expand it to its real address.

Insert Recipient
[option] Insert & Expand Recipient
Insert the chosen nickname.
Insert the real address of the nickname."
Eudora Mac Manual at 149-50.

"Edit

This menu provides text editing tools.

Exhibit K



* * *

Finish Address Book Entry
Complete the partial text of a nickname.

Insert Recipient
Insert the chosen recipient"
Eudora Windows Manual at 137.

"**Special**

This menu lets you use additional Eudora functions.