

EXHIBIT 1

(12) **United States Patent**
Hedløy

(10) **Patent No.:** **US 7,917,843 B2**

(45) **Date of Patent:** ***Mar. 29, 2011**

(54) **METHOD, SYSTEM AND COMPUTER READABLE MEDIUM FOR ADDRESSING HANDLING FROM A COMPUTER PROGRAM**

FOREIGN PATENT DOCUMENTS

EP 0 093 250 A2 9/1983

(Continued)

(75) Inventor: **Atle Hedløy**, Madrid (ES)

(73) Assignee: **Arendi S.A.R.L.**, Luxembourg (LU)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 394 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/182,048**

(22) Filed: **Jul. 29, 2008**

(65) **Prior Publication Data**

US 2008/0313159 A1 Dec. 18, 2008

Related U.S. Application Data

(63) Continuation of application No. 09/923,134, filed on Aug. 6, 2001, now Pat. No. 7,496,854, which is a continuation of application No. 09/189,626, filed on Nov. 10, 1998, now Pat. No. 6,323,853.

(30) **Foreign Application Priority Data**

Sep. 3, 1998 (NO) 984066

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/230; 715/234; 715/711; 715/752; 715/825; 715/853**

(58) **Field of Classification Search** **715/230**
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,350,695 A 10/1967 Kaufman et al. 340/172.5

(Continued)

OTHER PUBLICATIONS

Person et al., "Special Edition Using Microsoft Word 97" by Que Publisher Pub. date Dec. 16, 1996, pp. 475-514.*

(Continued)

Primary Examiner — Doug Hutton

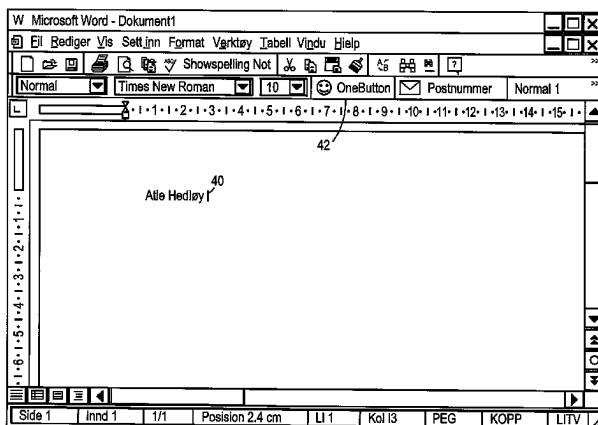
Assistant Examiner — Quoc A Tran

(74) *Attorney, Agent, or Firm* — Sunstein Kann Murphy & Timbers LLP

(57) **ABSTRACT**

A method, system and computer readable medium for providing for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

44 Claims, 14 Drawing Sheets



US 7,917,843 B2

Page 2

U.S. PATENT DOCUMENTS

4,270,182	A	5/1981	Asija	364/900	6,085,226	A	7/2000	Horvitz	709/203
4,674,065	A	6/1987	Lange et al.	364/900	6,092,074	A	7/2000	Rodkin et al.	707/102
5,226,117	A	7/1993	Miklos	395/157	6,108,686	A	8/2000	Williams	709/202
5,267,155	A	11/1993	Buchanan et al.	364/419.14	6,151,584	A	11/2000	Papierniak et al.	705/10
5,331,555	A	7/1994	Hashimoto et al.	364/419.07	6,178,411	B1	1/2001	Reiter	705/408
5,355,472	A	10/1994	Lewis	395/600	6,182,133	B1	1/2001	Horvitz	709/223
5,375,200	A	12/1994	Dugan et al.	395/159	6,192,380	B1	2/2001	Light et al.	707/505
5,388,150	A	2/1995	Schneyer et al.	379/67	6,208,975	B1	3/2001	Bull et al.	705/14
5,392,386	A	2/1995	Chalas	395/155	6,233,570	B1	5/2001	Horvitz et al.	706/11
5,416,901	A	5/1995	Torres	395/159	6,253,188	B1	6/2001	Witek et al.	705/14
5,491,783	A	2/1996	Douglas et al.	395/159	6,260,035	B1	7/2001	Horvitz et al.	706/60
5,491,784	A	2/1996	Douglas et al.	395/159	6,262,730	B1	7/2001	Horvitz et al.	345/337
5,491,785	A	2/1996	Robson et al.	395/162	6,285,987	B1	9/2001	Roth et al.	705/27
5,495,565	A	2/1996	Millard et al.	395/146	6,304,881	B1	10/2001	Halim et al.	707/201
5,500,859	A	3/1996	Sharma et al.	370/81	6,317,761	B1	11/2001	Landsman et al.	707/513
5,530,853	A	6/1996	Schell et al.	395/600	6,323,853	B1	11/2001	Hedloy	345/339
5,546,447	A	8/1996	Skarbo et al.	379/142	6,338,957	B2	1/2002	Onishi et al.	435/108
5,566,330	A	10/1996	Sheffield	395/600	6,373,940	B2	4/2002	Shaffer et al.	379/355.02
5,576,955	A	11/1996	Newbold et al.	395/795	6,377,965	B1	4/2002	Hachamovitch et al.	707/534
5,579,467	A	11/1996	Capps	395/768	6,385,592	B1	5/2002	Angles et al.	705/14
5,587,911	A	12/1996	Asano et al.	364/444.2	6,421,693	B1	7/2002	Nishiyama et al.	707/507
5,606,712	A	2/1997	Hidaka	395/800	6,499,021	B1	12/2002	Abu-Hakima	706/10
5,634,121	A	5/1997	Tracz et al.	395/602	6,651,217	B1	11/2003	Kennedy et al.	715/507
5,640,565	A	6/1997	Dickinson	395/683	6,738,615	B1	5/2004	Chow et al.	455/415
5,666,502	A	9/1997	Capps	345/352	6,826,407	B1	11/2004	Helferich	455/466
5,708,804	A	1/1998	Goodwin et al.	395/603	7,039,599	B2	5/2006	Merriman et al.	705/14
5,724,597	A	3/1998	Cuthbertson et al.	395/793	7,051,019	B1	5/2006	Land et al.	707/4
5,732,229	A	3/1998	Dickinson	395/334	7,136,853	B1	11/2006	Kohda et al.	707/6
5,732,395	A	3/1998	Alexander Silverman	704/260	7,155,451	B1	12/2006	Torres	707/104.1
5,734,871	A	3/1998	Kleinerman et al.	395/500	7,216,292	B1	5/2007	Snapper et al.	715/507
5,742,905	A	4/1998	Pepe et al.	455/461	7,272,604	B1	9/2007	Hedloy	707/10
5,754,636	A	5/1998	Bayless et al.	379/142	7,496,854	B2	2/2009	Hedloy	715/780
5,761,656	A	6/1998	Ben-Shachar	707/4	7,505,974	B2	3/2009	Gropper	707/10
5,761,663	A	6/1998	Lagarde et al.	707/10	7,509,349	B2	3/2009	Celik	707/200
5,774,887	A	6/1998	Wolff et al.	707/1	2002/0054092	A1	5/2002	Hedloy	345/748
5,778,367	A	7/1998	Wesinger, Jr. et al.	707/10	2002/0188683	A1	12/2002	Lytte et al.	709/206
5,781,189	A	7/1998	Holleran et al.	345/335	2003/0033288	A1	2/2003	Shanahan et al.	707/3
5,787,451	A	7/1998	Mogilevsky	707/533	2006/0047644	A1	3/2006	Bocking et al.	707/4
5,793,972	A	8/1998	Shane	395/200.49	2007/0244907	A1	10/2007	Hedloy	707/10
5,794,228	A	8/1998	French et al.	707/2	2008/0313159	A1	12/2008	Hedloy	707/3
5,794,259	A	8/1998	Kikinis	707/507	2010/0211600	A2*	8/2010	Hedloy	707/770
5,799,302	A	8/1998	Johnson et al.	707/7	2010/0281354	A1	11/2010	Hedloy	715/220
5,805,886	A	9/1998	Skarbo et al.	395/685					
5,805,911	A	9/1998	Miller	395/796					
5,815,830	A	9/1998	Anthony	707/6					
5,826,257	A	10/1998	Snelling, Jr.	707/4					
5,835,089	A	11/1998	Skarbo et al.	345/335					
5,859,636	A*	1/1999	Pandit	715/204					
5,860,073	A	1/1999	Ferrel et al.	715/255					
5,864,848	A	1/1999	Horvitz et al.	707/6					
5,873,107	A	2/1999	Borovoy et al.	707/501					
5,873,108	A	2/1999	Goyal et al.	707/507					
5,884,309	A	3/1999	Vanechanos, Jr.	707/10					
5,893,093	A	4/1999	Wills	707/5					
5,896,321	A	4/1999	Miller et al.	365/189.01					
5,896,533	A	4/1999	Ramos et al.	395/680					
5,907,838	A	5/1999	Miyasaka et al.	707/4					
5,913,214	A	6/1999	Madnick et al.	707/10					
5,923,848	A	7/1999	Goodhand et al.	395/200.49					
5,924,074	A	7/1999	Evans	705/3					
5,924,090	A	7/1999	Krellenstein	707/5					
5,926,808	A	7/1999	Evans et al.	707/3					
5,930,471	A	7/1999	Milewski et al.	395/200.04					
5,930,764	A	7/1999	Melchione et al.	705/10					
5,940,484	A	8/1999	DeFazio et al.	379/142					
5,946,647	A	8/1999	Miller et al.	704/9					
5,946,679	A	8/1999	Ahuja et al.	707/3					
5,999,938	A	12/1999	Bliss et al.	707/102					
6,006,218	A	12/1999	Breese et al.	707/3					
6,006,351	A	12/1999	Peretz et al.	714/751					
6,012,075	A	1/2000	Fein et al.	707/540					
6,018,571	A	1/2000	Langlois et al.	379/207					
6,021,403	A	2/2000	Horvitz et al.	706/45					
6,026,398	A	2/2000	Brown et al.	707/5					
6,028,605	A	2/2000	Conrad et al.	345/354					
6,067,565	A	5/2000	Horvitz	709/218					
6,081,788	A	6/2000	Appleman et al.	705/14					
6,085,201	A*	7/2000	Tso	715/205					

FOREIGN PATENT DOCUMENTS

EP	1 630 642	3/2006
GB	2196454	4/1988
GB	2 318 703	4/1998
WO	WO 97/21183	6/1997
WO	WO 98/16890	4/1998
WO	WO 98/19259	5/1998
WO	WO 98/34391	8/1998

OTHER PUBLICATIONS

- Plotkin, David, *Address for Success*, Bay Area Computer Currents, p. 36-38, Mar. 21, 1995.
- CTAGS(1) Manual Page.
- Wilson, Eve, *Guiding Lawyers: Mapping Law into Hypertext*, Artificial Intelligence Review 6, pp. 161-189, 1992.
- MacWEEK Report, Aug. 8, 1996.
- Langberg, Mike, *Show of Potential Apple Breaks New Ground by Displaying What's on its Drawing Board 'Innovation is at the Heart of What We Do'*, San Jose Mercury News, Aug. 7, 1996, p. 1C.
- Anind K. Dey, et al., *CyberDesk: The Use of Perception in Context-Aware Computing*, PUI Workshop Submission, Proc. Of 1997 Workshop on Perceptual User Interfaces (PUI '97), pp. 26-27, Oct. 1997.
- Dey, Anind K., *Context-Aware Computing: The CyberDesk Project*, Future Computing Environments, AAAI '98 Spring Symposium, Stanford University, pp. 51-55, Mar. 23-25, 1998.
- Anind K. Dey, et al., *CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services*, Knowledge-Based Systems, vol. 11, No. 1, pp. 3-13, Sep. 1998.
- L. Nancy Garrett, et al., *Intermedia: Issues, Strategies, and Tactics in the Design of a Hyermedia Document System*, pp. 163-174, © 1986.
- Langberg, Mike, *'Innovation Is at the Heart of What We Do'*, Apple Breaks New Ground by Displaying What's on it's Drawing Board, Mercury News, pp. 1-2, Aug. 7, 1996.

US 7,917,843 B2

Page 3

- Henry Lieberman, et al., Training Agents to Recognize Text by Example, Proc. of the Third Annual Conference on Autonomous Agents, Seattle, WA, 1999, pp. 116-122.
- Bonnie A. Nardi, et al., Collaborative, Programmable Intelligent Agents, Website, pp. 1-11, Mar. 1998.
- Addressmate Automatic Envelope Addressing Program, User's Manual, 88 pages, © 1991.
- Brown, Peter, Unix Guide, 1 page, 1995.
- N.D. Beitner, et al., Multimedia Support and Authoring in Microcosm: an Extended Model, 12 pages, no date available.
- McMahon, Lee E., SED—A Non-interactive Text Editor, Bell Laboratories, 10 pages, Aug. 15, 1978.
- SED(1), BSD Reference Manual Page, 6 pages, Dec. 30, 1993.
- AddressMate for Windows, Version 2.0, Product Box and License Agreement, 24 pages, no date available.
- "AddressMate Incorporates Correction Capability," The New York Law Publishing Company, 1 page, Jan. 1996.
- Mar. 1996 Reviews, website available at www.techweb.com/winmag/library/1996/0396/03rvh002.htm, 9 pages.
- Magid, Lawrence J., "Addressing the Matter of Labels," Los Angeles Times, 2 pages, Sep. 23, 1996.
- Important Note for New AddressMate Users, 1 page, no date available.
- Important Tips for LabelWriter Owners Using AddressMate, 3 pages, Jan. 1995.
- Addressing the Issues, 4 pages, Jun. 24, 1993.
- "Whye Do it the Olde Way" AddressMate Advertisement, 2 pages, no date available.
- Robinson, Philip, "The Envelope, Please: It's AddressMate," San Jose Mercury News, Sunday, 1 page, Jul. 19, 1992.
- Supkoff, Steve, AddressMate, PCM, 2 pages, Aug. 1992.
- Shannon, L.R., "Addressing Envelopes," The New York Times, 1 page, Tuesday, Jul. 14, 1992.
- Schwabach, Bob, "Addressing for Success," On Computers Column, 1 page, Mar. 16, 1992.
- INFOWORLD, p. 15, Mar. 16, 1992.
- Hart, Ken, Simplify Envelope Printing with AddressMate, Computer Shopper, 1 page, no date available.
- "Colvin's Beta-Testing of AddressMate Sparks Developer's Appreciation," Westview, 1 page, Apr. 6, 1992.
- Press Release: AddressMate Software Automatically Addresses and Bar Codes Envelopes to Save Time and Money, 2 pages, Mar. 9, 1992.
- Long, George M., Letter to Mr. David Block, 1 page, Nov. 11, 1994.
- Damiano, Philip J., Letter to Mr. David Block, 2 pages, Jan. 5, 1994.
- Kahn, Ed, "Envelope Addressing Finally Simplified AddressMate Does Addressing and Much More," Microtimes, 3 pages, Nov. 27, 1995.
- Bernthal, Kristen, "LabelWriter XL Plus," PC Catalog, 1 page, Aug. 18, 1995.
- Address Fixer for Microsoft Word and Office, Product Box, 10 pages, no date available.
- AddressMate Plus, Product Box and License Agreement, 16 pages, no date available.
- Getting Results with Microsoft Office for Windows 95, Version 7.0, 635 pages, © 1995-1997.
- Getting Results with Microsoft Office 97, 701 pages, © 1995-1997.
- Microsoft Word, User's Guide, Version 6.0, 449 pages, © 1993-1994.
- CTAGS(1) Manual Page, 3 pages, Oct. 11, 1993 (Last Change).
- Bernstein, Matt, An Apprentice that Discovers Hypertext Links, pp. 213-223, no date available.
- Contextual Menu Manager/Apple Data Detectors, 5 pages, no date available.
- P.J. Brown et al., A Help System Based on UNIX Help Manual, 7 pages, Feb. 1987.
- Charles H. Franke III, et al., Authoring a Hypertext Unix Help Manual, 8 pages, © 1995.
- Brown, P.J., Guide User Manual, 1985, sixteenth impression, 55 pages, Apr. 1995.
- Wilson, Eve, "Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System," Literary and Linguistic Computing, vol. 5, No. 2, 11 pages, 1990.
- John Robertson, et al., The Hypermedia Authoring Research Toolkit, ECHI '94 proceedings, pp. 177-185, Sep. 1994.
- What is Wiki and Wiki History webpages, available at wiki.org/wiki.cgi?WhatIsWiki and www.c2.org/cgi/wiki?WikiHistory, 8 pages, no date available.
- Wilson, Eve, Integrated Information Retrieval for Law in a Hypertext Environment, Annual ACM Conference on Research and Development in Information Retrieval, INRIA, 23 pages, 1988.
- Multimedia Hyperlinks Automatically Created for Reference Documents, Research Disclosure, 2 pages, Jun. 1993.
- Wiki Wiki Origin, <http://www.c2.com/cgi/wiki?WikiWikiOrigin>, 2 pages, no date available.
- Blak, Joy-Lyn, WikiWikiWeb, Computer World, 4 pages, Jan. 29, 2001.
- Microsoft's Supplemental Responses to Arendi's Interrogatories, in *Arendi USA et al. v. Microsoft Corporation*, Civil Action 02-CV-343 (ECT) from the United States District Court for the District of Rhode Island, 4 pages, Apr. 7, 2003.
- "Apple Introduces Internet Address Detectors," Press Release, 3 pages, 2 pages, Sep. 8, 1997.
- Cunningham, Cara, "Apple Kicks Off Macworld with Talk of Revival," new software demos, InfoWorld Electric, 2 pages, Aug. 7, 1996.
- Staten, James, "Apple Looks to the Future," MacWeek, 2 pages, Aug. 7, 1996.
- Simmons, Mark, "Striking a Key Note," Mac Addict Online, 5 pages, Aug. 8, 1996.
- Miller, Jim, Email regarding Apple Data Detectors, 1 page, Jan. 8, 1997.
- Apple Data Detectors—Now Shipping web page, 5 pages, Jan. 6, 1997.
- The Apple Data Detectors FAQ, 8 pages, Jan. 6, 1997.
- Apple Data Detector Webpages, available on web.archive.org/web/20020601164217/www.apple.com/applescript/data_detectors, 15 pages, © 2002.
- Apple Data Detectors 1.0.2 Read Me, 3 pages, no date available.
- Developer's Guide to Apple Data Detectors, 34 pages, Dec. 1, 1997.
- AppleScript Editors, Utilities & Environments, available at www.applescriptsourcebook.com/links/applescripteditors.html, 4 pages, dated Jan. 11, 2004.
- eMailman Internet Address Detectors, 3 pages, © 1996-2000.
- Tannehill, Steve, News from Jul. 1997, 6 pages, Jul. 31, 1997.
- Control-Click! The Archive, 2 pages, © 1997-2000.
- Contextual Menus: One of the System 8's Greatest Features, in ApplePress, 10 pages, no date available.
- Trygve's CMM Plug-Ins Home, available at Web.archive.org/web/19980130053511/www.bombaydigital.com/cmms, 7 pages, no date available.
- ADD Depot, available from Web.archive.org/web/20000819091818/http://homepage.mac.com/Mathewmiller/add, 4 pages, Jan. 12, 2004.
- Press Release: Apple Introduces Internet Address Detectors, 4 pages, Sep. 8, 1997.
- Apple Introduces Internet Address Detectors, Newsbytes, 11 pages, Sep. 29, 1997.
- Williams, Greg, Strategy Mosaic: Understanding Apple's Dual OS Strategy, 7 pages, no date available.
- "Taking [control] of your Mac with System 8," The MacAuthority, 6 pages, Jan. 1998.
- Apple Data Detectors 1.0.2, TidBITS Updates, 1 page, Mar. 8, 1998.
- Apple Data Detectors 1.0.2, TidBITS #419, 1 page, Mar. 9, 1998.
- Engst, Tanya, More on Context on Contextual Menus, tidbits #399, 2 pages, Sep. 29, 1997.
- Engst, Tanya, Of Mice and Menus, TidBITS #399, 3 pages, Sep. 22, 1997.
- Whaley, Charles, "Will This be Enough to Kick-Start Apple?," Computing Canada, 4 pages, Aug. 4, 1997.
- MacOS8.com—Mac OS 8 InDepth, 3 pages, no date available.
- "A Farewell to the Apple Advanced Technology Group," SIGCHI, vol. 30, No. 2, 3 pages, Apr. 1998.
- James R. Miller, et al., From Documents to Objects, in SIGCHI, vol. 30, No. 2, 11 pages, Apr. 1998.
- Thomas Bonura et al., Drop Zones, in SIGCHI, vol. 30, No. 2, 8 pages, Apr. 1998.

US 7,917,843 B2

Page 4

- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Judge Ernest C. Torres, Brief for the Defendant-Cross Appellant Frank E. Scherkenbach, 59 pages, Jul. 7, 2005.
- United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law that the '853 Patent is Invalid, 16 pages, Oct. 15, 2004.
- United States District Court for the District of Rhode Island, Civ. A. No. 02-CV-343 (ECT), Plaintiff's Reply Memorandum in Support of their Motion for New Trial, Francis A. Connor, 9 pages, Nov. 4, 2004.
- United States Court of Appeals for the Federal Circuit, Arendi Reply Brief, 37 pages, Sep. 2, 2005.
- United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Memorandum in Support of Defendant Microsoft Corporation's Opposition to Arendi's Motion for a New Trial, Patricia A. Sullivan, 15 pages, Oct. 27, 2004.
- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Brief of Plaintiffs-Appellants Arendi U.S.A., Inc. and Arend Holding Limited, Donald R. Dunner, 61 pages, Apr. 25, 2005.
- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Reply Brief for Defendant-Cross Appellant, Frank E. Scherkenbach, 27 pages, Oct. 3, 2005.
- United States District Court for the District of Rhode Island, Order Denying Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law That the '853 Patent is Invalid, C.A. No. 02-343T, Ernest C. Torres, Chief Judge, 6 pages, Nov. 30, 2004.
- United States District Court for the District of Rhode Island, C.A. No. 02-343T, Order Denying Plaintiffs' Motion for New Trial, Ernest C. Torres, Chief Judge, 18 pages, Nov. 30, 2004.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from 2:00 PM, 48 pages, Sep. 13, 2004, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 14, 2004, 165 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 15, 2004, 111 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 16, 2004, 142 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 17, 2004, 66 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 20, 2004, Providence, RI, 225 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 21, 2004, Providence, RI, 170 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 22, 2004, Providence, RI, 156 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 23, 2004, Providence, RI, 140 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 27, 2004, Providence, RI, 17 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 28, 2004, Providence, RI, 106 pages.
- Novell GroupWise User's Guide for Windows 16-BIT, Version 5.2, Novell, Inc., Orem, Utah, MS 125993, 235 pages, 1993.
- Novell GroupWise Webaccess User's Guide, Novell, Inc., Orem, Utah, MS 126785, 37 pages, 1998.
- Novell GroupWise User's Guide for Windows 32-BIT, Novell, Inc., Orem, Utah, MS 126463, 322 pages, 1998.
- Andrew Wood, et al., "CyberDesk: Automated Integration of Desktop and Network Services", GUVU Technical Support, OIT-GVU-97-11, 5 pages, May 1997.
- Gregory D. Abowd, et al., "Applying Dynamic Integration as a Software Infrastructure for Context-Aware Computing," GUVU Technical Report, GIT-GVU-97-18, 21 pages, Sep. 1997.
- Gregory D. Abowd, et al., "Context-awareness in Wearable and Ubiquitous Computing," GUVU Technical Report, GIT-GVU-97-11, 13 pages, Mar. 1997.
- Apple Data Detectors User's Manual, 16 pages, © 1997.
- Apple Internet Address Detectors User's Manual, 16 pages, © 1997.
- Apple Introduces Internet Address Detectors, 4 pages, Sep. 8, 1997.
- Contextual Menu Manager/Apple Data Detectors, 5 pages, date unavailable.
- CoStar User Manual for AddressMate and AddressMate Plus, 219 pages, © 1994-1995.
- Anind K. Dey, et al., "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services," GUVU Technical Report, GIT-GVU-97-10, 18 pages, 1997.
- Wilson, Eve, "Links and Structures in Hypertext Databases for Law," Proceedings of the First European Conference on Hypertext, INRIA, 20 pages, Nov. 1990.
- Milind S. Pandit, et al., "The Selection Recognition Agent: Instance Access to Relevant Information and Operations," Proc. of Intelligent User Interfaces 1997, Orlando, FL, 6 pages, 1997.
- Spell, iSpell, Spellout, 71 pages, © 1994.
- Microsoft Word 97 Help File entitled "Automatically Check Spelling and Grammar as You Type," 1 page, 2003.
- Microsoft Word 97 Help File entitled "Turn Automatic Changes On or Off," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Automatically Correct Text," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Field Codes: Hyperlink Field," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Change the contents of an AutoCorrect Entry," 1 page, no date available.
- Excerpt from Jury Charge Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, 45 pages, Sep. 28, 2004.
- Microsoft Corporation and Dell Inc.'s Answer, Affirmative Defenses and Counterclaims to Complaint, *Arendi Holding Ltd. v. Microsoft Corp. et al.*, C.A. No. 09-119-JJF, from the United States District Court for the District of Delaware, pp. 1-17, Apr. 30, 2009.
- "More about The Database," <http://c2.com/cgi/wiki?MoreAboutTheDatabase>, pp. 1-3, last edited May 7, 2009.
- "Ward Cunningham," <http://c2.com/cgi/wiki?WardCunningham>, pp. 1-15, last edited Apr. 23, 2009.
- "Adding New Pages," <http://c2.com/cgi/wiki?AddingNewPages>, pp. 1-2, last edited Jun. 8, 2009.
- "Wiki Wiki Web," <http://c2.com/cgi/wiki?WikiWikiWeb>, 1 page, last edited Jun. 1, 2009.
- Quinlan, Tom, "Apple System Software Branches Out," 1 page, Mar. 16, 1992.
- "Introducing AddressMate Plus, The Address Book Manager for Microsoft Word," 4 pages, May 1996.
- "Corel Professional Office 7 Quick Results," 534 pages, no date available.
- "Getting Results with Microsoft Office for Windows 95," Microsoft, 636 pages, 1995.
- "AddressMate for Windows, Adds all the missing pieces to your word processor," 6 pages, no date available.
- Opposition against European Patent EP 1 171 836, Letter to the European Patent Office in Preparation for the Oral Proceedings scheduled for May 26, 2009, 31 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 1 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 2 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 3 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 4 Claims, 3 pages, Mar. 26, 2009.

US 7,917,843 B2

Page 5

- Opposition against European Patent EP 1 171 836, Auxillary Request 5 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 6 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 7 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 8 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 9 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 10 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 11 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 12 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 13 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 14 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 15 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 16 Claims, 3 pages, Mar. 26, 2009.
- Expert Report of Professor Benjamin Goldberg, 43 pages, dated Mar. 10, 2008.
- Supplemental Expert Report of Professor Benjamin Goldberg, 15 pages, May 6, 2008.
- Second Supplemental Expert Report of Professor Benjamin Goldberg, 32 pages, Mar. 26, 2009.
- Expert Report of Professor Calvin Gidney III, 3 pages, Mar. 26, 2009.
- European Patent Office, Summons to Attend Oral Proceedings pursuant to Rule 115(1) EPC, 18 pages, Jan. 23, 2009.
- Affidavit of James Miller, 2 pages, Jul. 19, 2006.
- Affidavit of David Block, 23 pages, Jul. 21, 2006.
- Order Re Claim Construction, in *Arendi U.S.A. Inc. et al. v. Microsoft Corp.*, CA No. 02-343-T, from United States District Court for the District of Rhode Island, 4 pages, Sep. 27, 2004.
- Expert Report of W. Bruce Croft, PH.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 66 pages, Mar. 1, 2004.
- Supplemental Expert Report of W. Bruce Croft, PH.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 42 pages, Jun. 9, 2004.
- Affirming Claim Construction Order, in *Arendi USA et al. v. Microsoft Corp.*, No. 05-1170, 05-1171, from Court of Appeals for the Federal Circuit, 1 page, Feb. 8, 2006.
- Miller, Jim, "Portfolio; Innovative Interfaces for Information Management," Website: <http://www.miramontes.com/portfolio/add/addl.html>, 1 page, no date available.
- Church, Kenneth et al., "Commercial Applications of Natural Language Processing," 30 pages, Mar. 13, 2003.
- Yankelovich, Nichole et al., "Intermedia: The concept and the Construction of a Seamless Information Environment," *Electronic Publishing Technologies*, pp. 81-96, Jan. 1988.
- Utting, Kenneth et al., "Context and Orientation in Hypermedia Networks," *AMC Transactions on Information Systems*, vol. 7, No. 1, pp. 58-84, Jan. 1989.
- Kahn, Paul, "Linking Together Books: Experiments in Adapting Published Material into Intermedia Documents," *Hypermedia*, vol. 1 No. 2, pp. 1-37, 1989.
- Combs, James, "Hypertext, Full-Text and Automatic Linking," pp. 83-98, 1990.
- Catlin, Karen Smith et al., "Hypermedia Templates: An Authors Tools," *Hypertext 1991 Proceedings*, pp. 147-160, Dec. 1991.
- Meyrowitz, Norman, "The Missing Link, Why We're All Doing Hypertext Wrong," *The Society of Text, Hypertext, Hypermedia, and the Social Construction of Information*, pp. 107-114, no date available.
- Halasz, Frank, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems," *Hypertext 87 Papers*, pp. 345-365, Nov. 1987.
- Haan, Bernard et al., "IRIS Hypermedia Services," pp. 36-51, Jan. 1992.
- McKnight, Dillon, "Review, The Textbook of the Future," *Hypertext: a Psychological Perspective*, pp. 19-51, 8 pages, Mar. 1993.
- Bouvin, Niels Olof, "Augmenting the Web Through Open Hypermedia," 95 pages, Nov. 2000.
- Halasz, Frank et al., "The Dexter Hypertext Reference Model," pp. 95-133, Dec. 7, 1989.
- Weinreich, Harald et al., "The Look of the Link—Concepts for the User Interface of Extended Hyperlinks," 10 pages, no date available.
- Fountain, Andrew et al., "Microcosm: An Open Model for Hypermedia with Dynamic Linking," *Hypertext: Concept Systems and Applications, Proceedings of the First European Conference on Hypertext*, France, 16 pages, Nov. 1990.
- Justus: Wilson, Eve, "Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System," *Library and Linguistics Computing*, vol. 5, No. 2, pp. 119-128, 1990.
- "Claris Em@iler Getting Started," Claris Corporation, pp. MS112968-MS113028, © 1995-1997.
- "OmniPage Pro for Windows 95," Version 7.0 Caere Corporation, pp. MS110194-MS110250, , © 1996.
- Moi Khor, Siew, Microsoft Office Excel 2003 Preview, 17 pages, Jun. 2003.
- Microsoft Corporation and Dell Inc.'s Invalidity Contentions, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-26, Aug. 14, 2009.
- Exhibit A, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-35, Aug. 14, 2009.
- Exhibit AA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Aug. 14, 2009.
- Exhibit AAA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit B, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit BB, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Aug. 14, 2009.
- Exhibit C, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit CC, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit D, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit DD, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 14, 2009.
- Exhibit E, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit EE, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Aug. 14, 2009.
- Exhibit F, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit FF, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-40, Aug. 14, 2009.
- Exhibit G, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit GG, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.

US 7,917,843 B2

Page 7

- idity, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 31, 2009.
- Microsoft Corp. and Dell, Inc.'s First Amended Answer, Affirmative Defenses and Counterclaims to Complaint, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Oct. 19, 2009.
- Plaintiff's Answer to Defendants' First Amended Counterclaims, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-19, Oct. 30, 2009.
- Microsoft Corp. and Dell, Inc.'s Second Amended Answer, Affirmative Defenses and Counterclaims to Complaint, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Dec. 8, 2009.
- Opposition for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-32, Jul. 26, 2006.
- In Response to Notice of Opposition for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-21, Mar. 7, 2007.
- Decision Revoking European Patent No. EP-B- 1171836, European Patent Office, pp. 1-13, Jul. 9, 2009.
- Notice of Appeal for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-2, Sep. 1, 2009.
- Grounds of Appeal for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-62, Nov. 19, 2009.
- Plaintiff's Answer to Microsoft Corporation's and Dell Inc.'s Second Amended Counterclaims, in *Arendi Holding Ltd. v. Microsoft Corp. and Dell Inc.*, CA No. 09-119-JJF-LPS, from United States District Court for the District of Delaware, pp. 1-20, Dec. 14, 2009.
- Submission of opponent (Microsoft) in opposition against European Patent No. 1 171 836, Appeal T1779/09-3501, in the European Patent Office, 5 pages, Mar. 25, 2010.
- Second Affidavit of David Block, 15 pages, May 15, 2009.
- Second Affidavit of James Miller, 36 pages, May 15, 2009.
- Hall, Wendy, et al., "Rethinking Hypermedia," Kluwer Academic Publishers, 87 pages, 1996.
- Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 196 pages, Aug. 13, 2004.
- Exhibits for the Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 334 pages, Aug. 13, 2004.
- Writ of Summons to Arendi Holding Co., District Court in the Hague and English translation, 65 pages, Oct. 26, 2005.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, 55 pages, date unknown.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 51 pages, date unknown.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, 26 pages, Sep. 24, 2008.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, English Translation, 23 pages, Sep. 24, 2008.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising a Counterclaim, With Exhibit, District Court in the Hague, No. 2006/0782, 58 pages, Jan. 31, 2007.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising a Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 23 pages, Jan. 31, 2007.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the Counterclaim With Exhibits, District Court in the Hague, No. 2006/0782, 126 pages, Mar. 12, 2008.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 33 pages, Mar. 12, 2008.
- Letter from Abraham H. Spierer to David Block, Dec. 19, 1994.
- Claris Em@iler User's Manual, 187 pages, © 1995-1997.
- "Word Lookup Data Detectors," Dictionary.com, available at <http://dictionary.reference.com/mac/add.html>, 3 pages, © 2004.
- Apple.com internet web pages (marked "MS 118139-MS 118143"), 5 pages, © 2001.
- "Overview: What are Apple Data Detectors", Apple.com internet web pages (marked "MA 118078-MS 118092"), 15 pages, © 2002.
- Addressmate Plus User Manual Addendum, 2 pages, Nov. 1, 1995.
- Now Software, Inc., "Now Contact & Up*to*Date Quick Reference Guide", 18 pages, © 1992-95.
- Now Software, Inc., "Now Contact version 3.5 User's Guide", 204 pages, © 1992-95.
- Now Software, Inc., "Now Up*to*Date version 3.5 User's Guide", 210 pages, © 1992-95.
- Now Software, Inc., "Now Utilities Quick Reference Guide", 6 pages, no date available.
- Now Software, Inc., "What's New in Now Up*to*Date & Now Contact", 14 pages, © 1995.
- Reply Brief of Plaintiffs-Appellants Arendi USA, Inc. and Arendi Holding Limited, United States Court of Appeals for the Federal Circuit, 37 pages, Sep. 2, 2005.
- Apple Data Detectors/Internet Address Detectors, ADD-IAD_1.0.2_Info.txt, 1 page, Mar. 5, 1998.
- Cortinas, M., Data Ratchet extracts info, ZD.com, 2 pages, Apr. 3, 1997.
- Glucose Unveils Data Ratchet v.1.1!, Glucose Development Corp., 5 pages, 1997.
- Glucose Releases Data Ratchet Source, Glucose Press Release, 1 page, Dec. 18, 1999.
- Are you a document mechanic?, from Glucose Development Corp.'s webpage (<http://glulabs.com>), 2 pages, 2002.
- Glucose Open Source Tools, from Glucose Development Corp., 2 pages, 1999.
- E-Mail 'Net Differences—Eudora Pro and E-Mail Connection shine in their own ways . . . , Communications Week, 4 pages, Jan. 6, 1997.
- Internet Update Sep. 10, 1997, Newsbytes news Network, 5 pages, Sep. 10, 1997.
- Brouwer, Rene, ADD→Address Book, 1 page, Jun. 8, 1998.
- Byrne, Mike, ADD→New Mail, 1 page, Aug. 23, 1998.
- MacCentral: Apple Macintosh News, 4 pages, Jan. 12-13, 2004.
- Turney, P., Learning to Extract Key Phrases from Text, National Research Council of Canada, 46 pages, Feb. 1999.
- Deposition of Atle Hedloy in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 162-213, Mar. 7, 2010.
- Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-4, 31-52, 56-57, 67, Apr. 9, 2010.
- Exhibit 15, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF from United States District Court for the District of Delaware , pp. 1-6, Apr. 9, 2010.
- Exhibit 16, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.
- Exhibit 17, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 18, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Apr. 9, 2010.
- Exhibit 19, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-13, Apr. 9, 2010.
- Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-4, 72-115, Apr. 9, 2010.
- Exhibit 11, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-7, Apr. 9, 2010.
- Exhibit 12, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.

US 7,917,843 B2

Page 8

- Exhibit 13, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 14, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-18, Apr. 9, 2010.
- Exhibit 15, Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Apr. 9, 2010.
- Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-166, Apr. 30, 2010.
- Exhibit A, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-30, Apr. 30, 2010.
- Exhibit B, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-15, Apr. 30, 2010.
- Exhibit C, Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-90, Apr. 30, 2010.
- Michael Bieber, "On Automating Hypertext Links in Knowledge-Based Decision Support Systems", Technical Report BCCS-90-08, pp. 1-22, Apr. 1990.
- N. Streitz, A. Rizk and J. André, Hypertext: Concepts, Systems and Applications: Proceedings of the First European Conference on Hypertext, INRIA, France, pp. 1-8, Nov. 1990.
- Michael Bieber, "Issues in Modeling A "Dynamic" Hypertext Interface", Technical Report BCCS-91-2, pp. 1-13, Apr. 1991.
- Michael Bieber and Tomás Isakowitz, "Valuation Links: Extending the Computational Power of Hypertext", Technical Report BCCS-91-1, pp. 1-11, Apr. 1991.
- Michael Bieber, Template-Driven Hypertext: A Methodology for Integrating a Hypertext Interface into Information Systems, Technical Report BCCS-91-3, pp. 1-23, Jun. 1991.
- Michael Bieber and Steven O. Kimbrough, "On Generalizing the Concept of Hypertext", Technical report BCCS-91-5, pp. 1-11, Jun. 1991.
- Michael Bieber and Tomás Isakowitz, "Bridge Laws in Hypertext A Logic Modeling Approach", Technical Report BCCS-91-4, pp. 1-30, Jun. 1991.
- Michael Bieber, "On Merging Hypertext Into Dynamic, Non-Hypertext Systems", Technical Report BCCS-91-14, pp. 1-24, Nov. 8, 1991.
- Mark Bernstein, Jay David Bolter, Michael Joyce and Elli Mylonas, "Architectures for Volatile Hypertext", Hypertext '91 Proceedings, pp. 243-280, Dec. 1991.
- Michael Bieber, "Providing Information Systems With Full Hypermedia Functionality", pp. 1-15, Oct. 1992.
- Mail Address Change Notification, IBM Technical Disclosure Bulletin, vol. 36 No. 1, pp. 70-72, Jan. 1993.
- D.D. Cowan, C.J.P. Lucena and R.G. Veitch, Towards CAAI: Computer Assisted Application Integration, Technical Report Series CS-93-17, University of Waterloo, pp. 1-23, Oct. 1993.
- Herman Kaind and Stefan Kramer, Semiautomatic Generation of Dictionary Links in Hypertext, pp. 1-14, Feb. 1, 1995.
- Michael Bieber, "On Integrating Hypermedia into Decision Support and Other Information Systems", Decision Support Systems 14, pp. 251-267, 1995.
- Proceedings, Sixth Message Understanding Conference (MUC-6), pp. 1-64, Nov. 6-8, 1995.
- Sargur N. Srihari, et al., "A System to Read Names and Addresses on Tax Forms", Proceedings of the IEEE, vol. 84, No's, Jul. 1996.
- Ph. Bonnet and S. Bressan, "Extraction and Integration of Data from Semi-structured Documents into Business Applications", Sloan WP#3979, CISL WP# 97-12, pp. 1-9, Sep. 1997.
- Borland, Russel, "Advance Self-Study Step by Step Microsoft Word 97 Advance Topics", Microsoft Press, 331 pages, 1997.
- Microsoft Word Language Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1133 pages, © 1997.
- Microsoft PowerPoint and Microsoft Outlook Language References, Official Technical Reference to Visual Basic for Applications, Microsoft Press, vol. 4, 539 pages, 1997.
- Microsoft Visual Basic for Applications and Shared Libraries References, vol. 5 of 5, Microsoft Press, 1005 pages, part No. 097-001685, © 1997.
- Microsoft Office 97 Resource Kit, Technical Information and Tools for the Support Professional, Microsoft Press, 1162 pages, © 1997.
- Microsoft Office 97 Reviewer's Guide, Professional Edition, 546 pages, Aug. 1996.
- Running Microsoft Word for Windows 95, in Depth Reference and Inside Tips from the Software Experts, Russell Borland, Microsoft Press, 1040 pages, © 1995.
- Running Microsoft Office 97, Standard and Professional, Michael Halvorson and Michael Young, Microsoft Press, 1130 pages, © 1997.
- Microsoft Word 97, self-Study Kit, Covers Microsoft Word 97 for Windows 95 and Windows NT version 4.0, Microsoft Press, 314 pages, © 1997.
- Microsoft Outlook 97, Step by Step, Self-Study Kit, Microsoft Press, 359 pages, © 1997.
- VBA Developer's Handbook, Reusable Code, Visio Professional Demo Copy, and VB5CCE, 1997.
- Microsoft97/Visual Basic Programmer's Guide, Microsoft Press, 548 pages, © 1997.
- Microsoft Access Language Reference and Office Data Access Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1287 pages, © 1997.
- Microsoft Office97 Step by Step, Visual Basic, Developer's Self-Study Guide, Microsoft Press, 359 pages, © 1997.
- Microsoft Office97 Developer's Handbook, Microsoft Press, Christine Solomon, 596 pages, © 1997.
- Microsoft Word Developer's Kit, Microsoft Professional Editions, Third Edition, Manufacturing Copy 1_{st}, Microsoft Press, 1106 pages, © 1995.
- Inside Microsoft Word, vol. 4, No. 1, The Cobb Group, pp. 1-16, Jan. 1997.
- Inside Microsoft Word, vol. 4, No. 2, The Cobb Group, pp. 1-16, Feb. 1997.
- Inside Microsoft Word, vol. 4, No. 3, The Cobb Group, pp. 1-16, Mar. 1997.
- Inside Microsoft Word, vol. 4, No. 4, The Cobb Group, pp. 1-16, Apr. 1997.
- Inside Microsoft Word, vol. 4, No. 7, The Cobb Group, pp. 1-16, Jul. 1997.
- Inside Microsoft Word, vol. 5, No. 9, ZD Journals, pp. 1-16, Sep. 1998.
- Inside Microsoft Word, vol. 5, No. 10, ZD Journals, pp. 1-16, Oct. 1998.
- Inside Microsoft Word, vol. 5, No. 11, ZD Journals, pp. 1-16, Nov. 1998.
- Eudora Mail Pro, Version 3.0 for Windows, User Manual, QUALCOMM Incorporated, 198 pages, Jun. 1997.
- Eudora Mail Pro, Version 3.1 for Macintosh, User Manual, QUALCOMM Incorporated, 199 pages, Jun. 1997.
- Non-Confidential Redacted Jun. 11, 2010 Rebuttal Expert Report Richard Taylor, 311 pages.
- Spell Checking for Microsoft Windows, Palantir Software, 36 pages, © 1986.
- Johannes, S., et al., "Active Messenger: Email Filtering and Mobile Delivery," *Massachusetts Institute of Technology*, pp. 1-122, Sep. 1999.
- Srihari, S, et al., "A System to Read Names and Addresses on Tax Forms," *Proceedings of the IEEE*, vol. 14, No. 7, pp. 1038-1049, Jul. 1996.
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626 As Filed, 58 pages, Nov. 10, 1998.*
- US Patent No. 6,323,853, Preliminary Amendment, U.S. Appl. No. 09/189,626, 20 pages, Apr. 29, 1999.*
- US Patent No. 6,323,853, Office Action, U.S. Appl. No. 09/189,626, 9 pages, Apr. 26, 2000.*
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626, Interview Summary, 2 pages, Jun. 14, 2000.*

US 7,917,843 B2

Page 9

US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Jul. 25, 2000.*

US Patent No. 6,323,853, Final Office Action, U.S. Appl. No. 09/189,626, 10 pages, Sep. 18, 2000.*

US Patent No. 6,323,853, Interview Summary, U.S. Appl. No. 09/189,626, 1 page, Oct. 17, 2000.*

US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Dec. 18, 2000.*

US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 7 pages, Jan. 4, 2001.*

US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 2 pages, Oct. 2, 2001.*

US Patent No. 7,496,854, Application As Filed, 100 pages, Aug. 6, 2001.*

US Patent No. 7,496,854, Preliminary Amendment, U.S. Appl. No. 09/923,134, 7 pages, Dec. 17, 2001.*

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 10 pages, Oct. 15, 2002.

US Patent No. 7,496,854, Response, 4 pages, U.S. Appl. No. 09/923,134, Apr. 15, 2003.

US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 7 pages, Jun. 27, 2003.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 6 pages, Aug. 21, 2003.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 4 pages, Nov. 19, 2003.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 5 pages, Feb. 10, 2004.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 4 pages, May 19, 2004.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 14 pages, Dec. 30, 2004.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 7 pages, Aug. 29, 2005.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 17 pages, Nov. 16, 2005.

US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 10 pages, Feb. 6, 2006.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 22 pages, Aug. 4, 2006.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 8 pages, Oct. 18, 2006.

US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 3 pages, Jan. 23, 2007.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Apr. 18, 2007.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 8 pages, Jul. 13, 2007.

US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 4 pages, Oct. 19, 2007.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 9 pages, Nov. 9, 2007.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Jan. 24, 2008.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 7 pages, Jul. 2, 2008.

US Patent No. 7,496,854, Amendment Under Rule 312; U.S. Appl. No. 09/923,134, 27 pages, Jul. 8, 2008.

US Patent No. 7,496,854, Response to Amendment Under Rule 312, U.S. Appl. No. 09/923,134, 4 pages, Jan. 21, 2009.

U.S. Appl. No. 12/841,302, Application As Filed, 47 pages, Jul. 22, 2010.

U.S. Appl. No. 12/841,302, Accelerated Examination Support Document, 150 pages, Jul. 29, 2010.

U.S. Appl. No. 12/841,302, Office Action, 22 pages, Nov. 24, 2010.

U.S. Appl. No. 12/841,302, Response, 18 pages, Dec. 14, 2010.

US Patent No. 7,272,604, U.S. Appl. No. 09/390,303 As Filed, 55 pages, Sep. 3, 1999.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 9 pages, May 8, 2002.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 7 pages, Aug. 8, 2002.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 11 pages, Sep. 27, 2002.

US Patent No. 7,272,604, Proposed Response, U.S. Appl. No. 09/390,303, 4 pages, Oct. 24, 2002.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 3 pages, Mar. 27, 2003.

US Patent No. 7,272,604, Final Office Action, U.S. Appl. No. 09/390,303, 9 pages, Apr. 21, 2003.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 16 pages, Oct. 21, 2003.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Nov. 17, 2003.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 6 pages, Apr. 6, 2004.

US Patent No. 7,272,604, Supplemental Notice of Allowance, U.S. Appl. No. 09/390,303, 2 pages, Mar. 7, 2006.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 10 pages, Jun. 6, 2006.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 19 pages, Dec. 5, 2006.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Feb. 7, 2007.

U.S. Appl. No. 11/745,186, Application As Filed, 61 pages, May 7, 2007.

U.S. Appl. No. 11/745,186, Office Action, 8 pages, Mar. 6, 2008.

U.S. Appl. No. 11/745,186, Response, 18 pages, May 14, 2008.

U.S. Appl. No. 11/745,186, Office Action, 17 pages, Oct. 14, 2008.

U.S. Appl. No. 11/745,186, Response, 28 pages, Feb. 13, 2009.

U.S. Appl. No. 11/745,186, Final Office Action, 16 pages, Mar. 26, 2009.

U.S. Appl. No. 11/745,186, Interview Summary, 4 pages, May 7, 2009.

U.S. Appl. No. 11/745,186, Response, 22 pages, Jun. 2, 2009.

U.S. Appl. No. 11/745,186, Advisory Action, 3 pages, Jun. 16, 2009.

U.S. Appl. No. 11/745,186, Request for Continued Examination, 3 pages, Jun. 22, 2009.

U.S. Appl. No. 11/745,186, Office Action, 40 pages, Sep. 29, 2009.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Mar. 26, 2010.

U.S. Appl. No. 11/745,186, Response, 29 pages, Mar. 29, 2010.

U.S. Appl. No. 11/745,186, Final Office Action, 10 pages, Jun. 15, 2010.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Aug. 5, 2010.

U.S. Appl. No. 11/745,186, Response, 24 pages, Aug. 10, 2010.

U.S. Appl. No. 11/745,186, Advisory Action, 3 pages, Aug. 16, 2010.

U.S. Appl. No. 11/745,186, Response, 18 pages, Sep. 23, 2010.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Dec. 14, 2010.

U.S. Appl. No. 12/963,744, Application As Filed, 53 pages, Dec. 9, 2010.

U.S. Appl. No. 12/963,744, Accelerated Examination Support Document, 80 pages, Dec. 9, 2010.

U.S. Appl. No. 12/841,302, Interview Summary, 3 pages, Dec. 20, 2010.

* cited by examiner

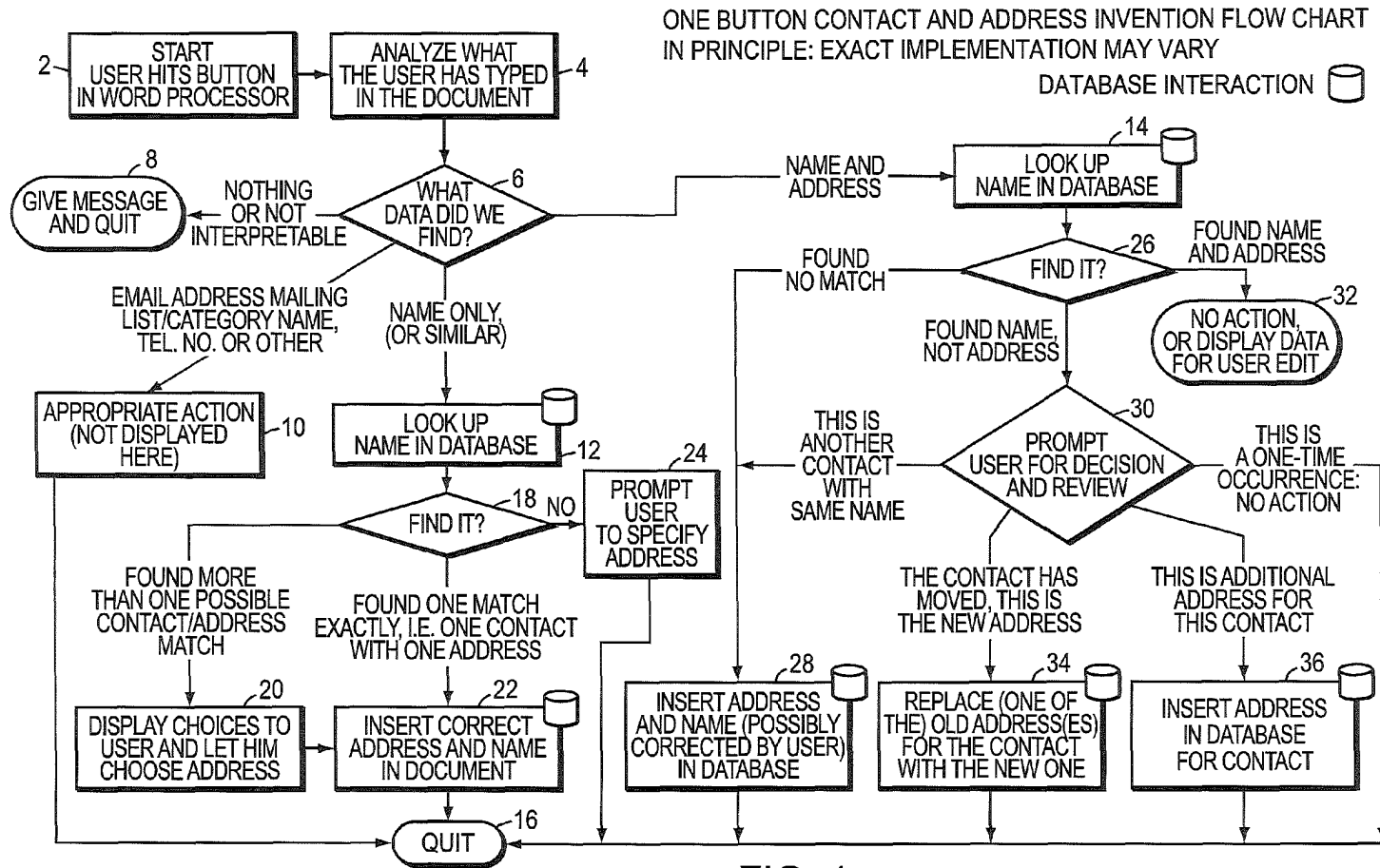


FIG. 1

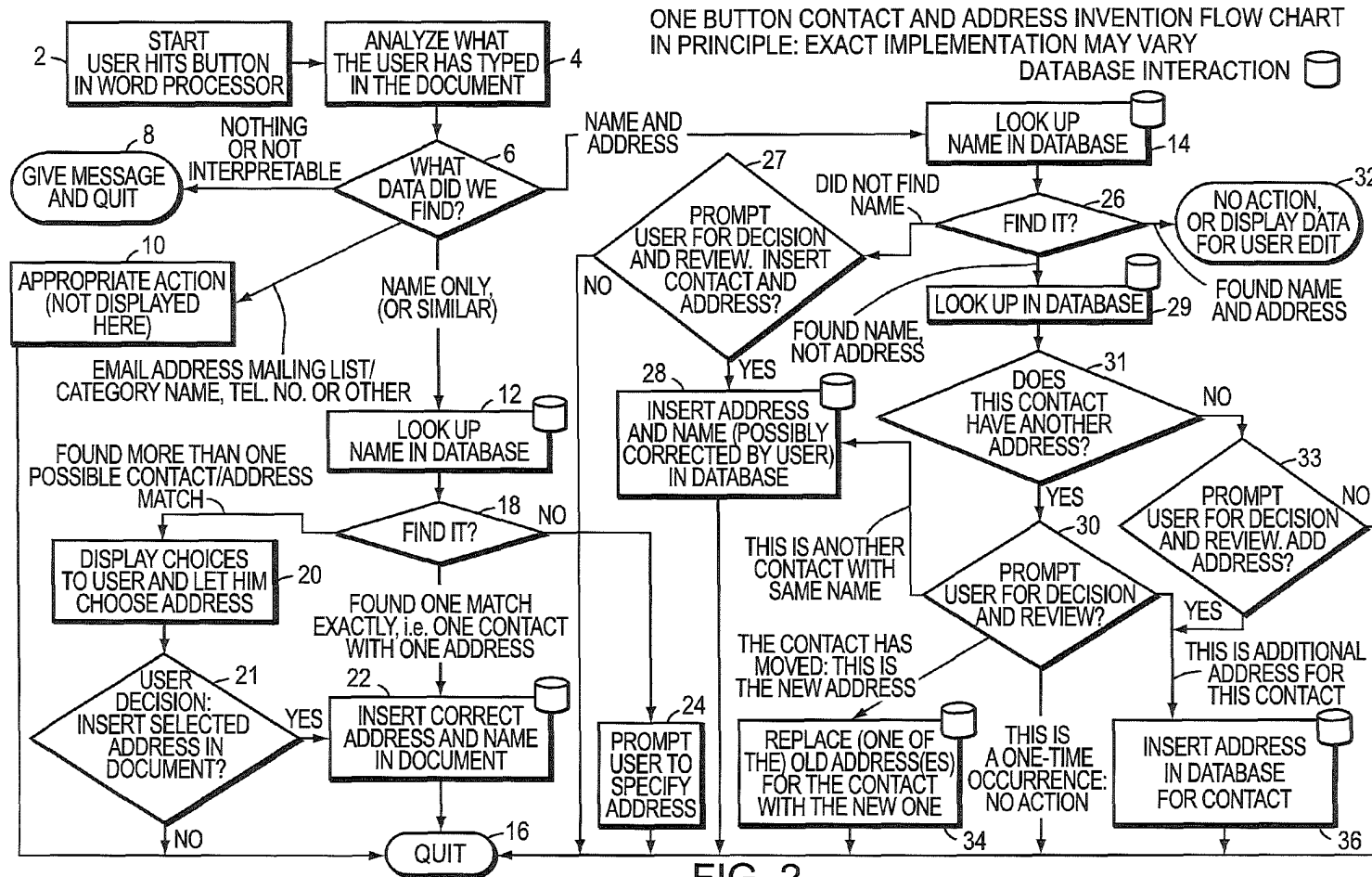


FIG. 2

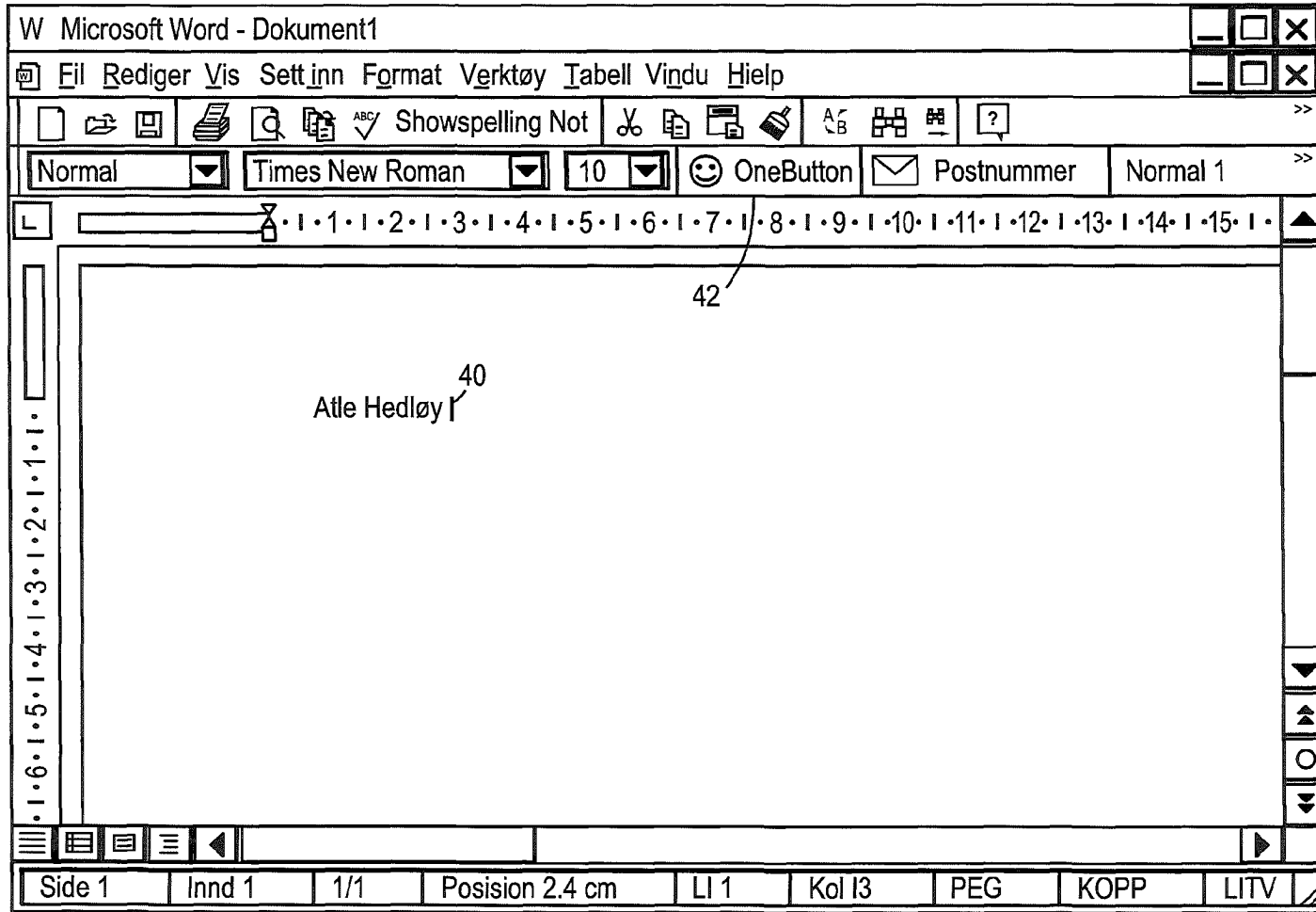


FIG. 3

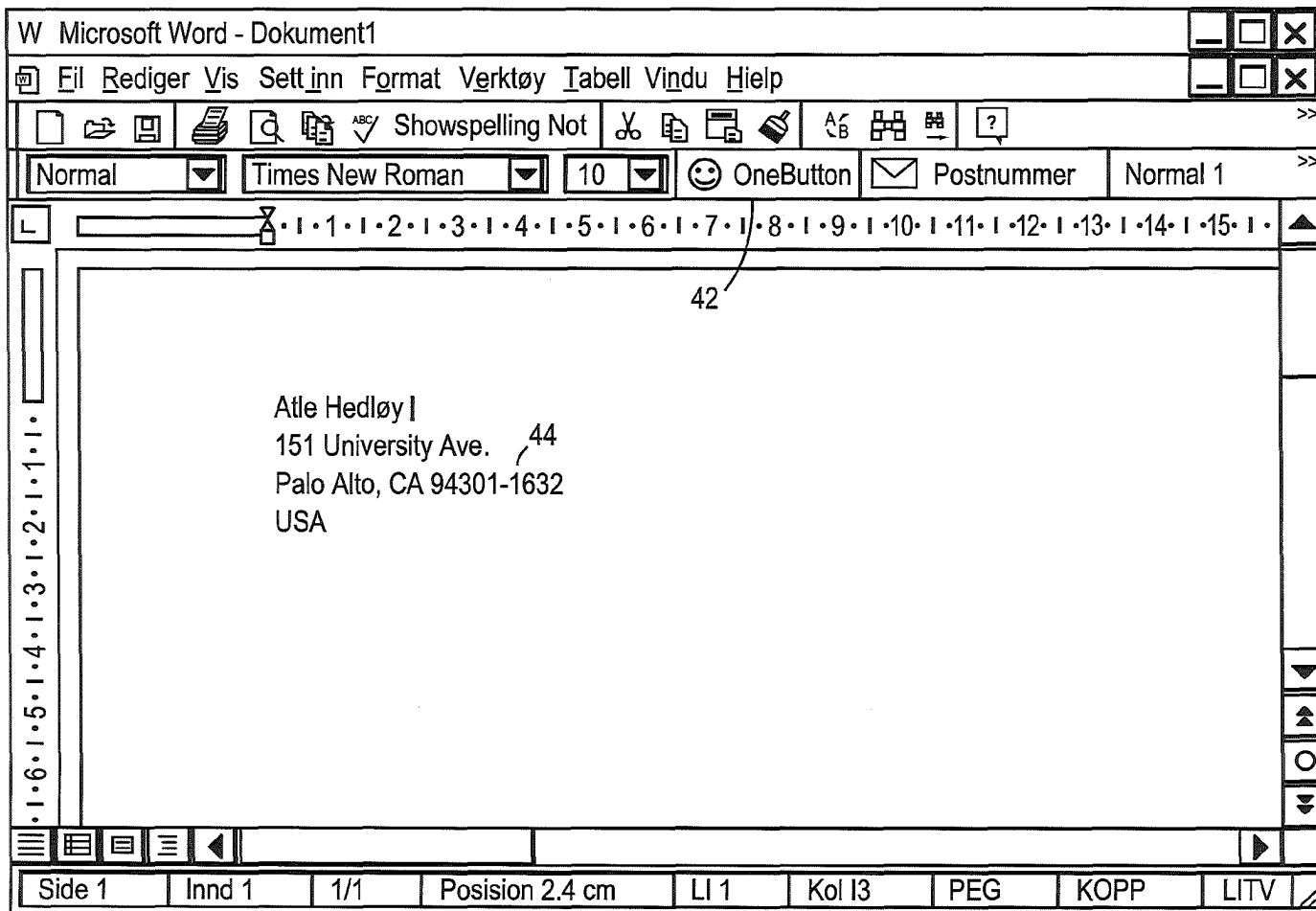


FIG. 4

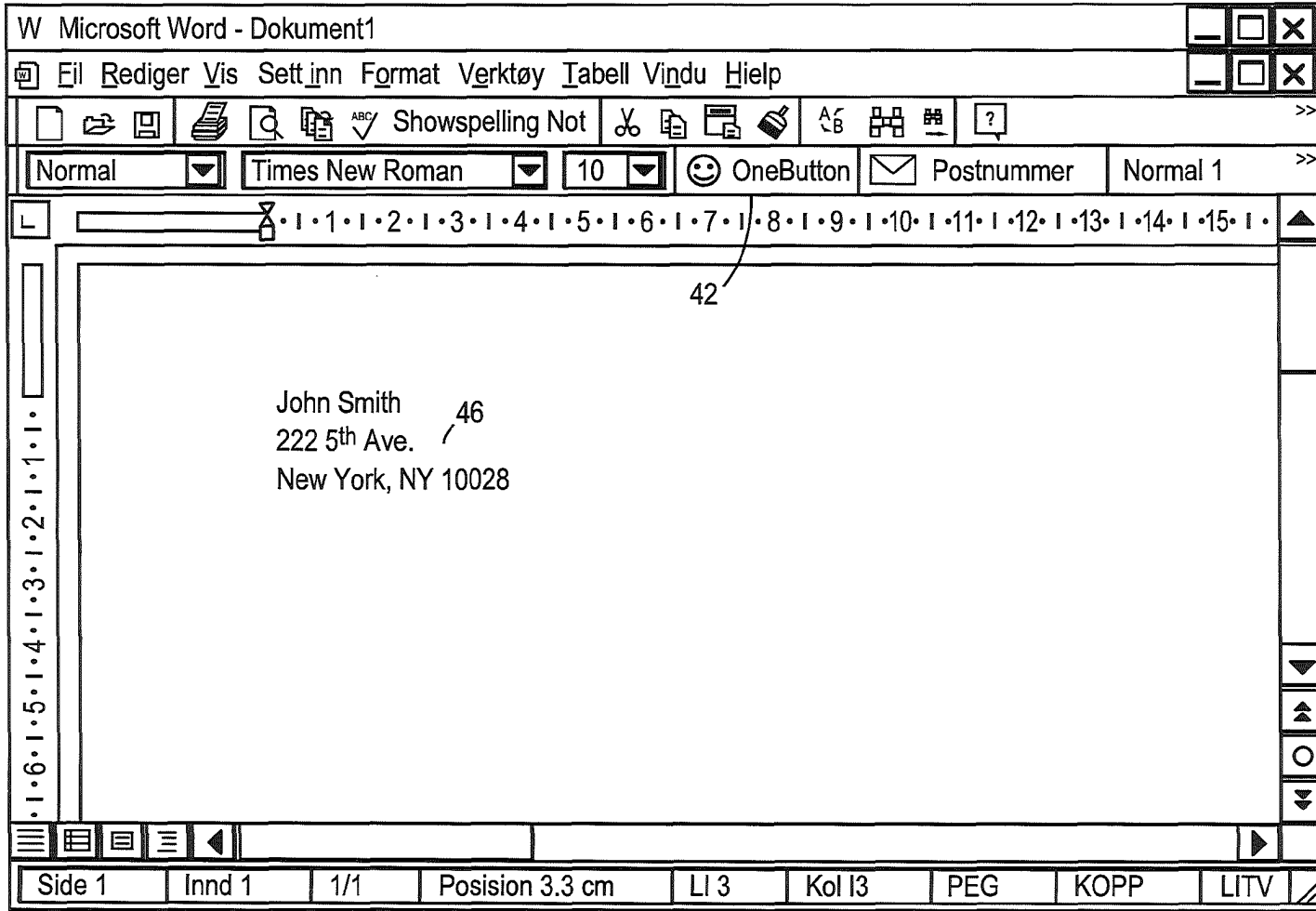


FIG. 5

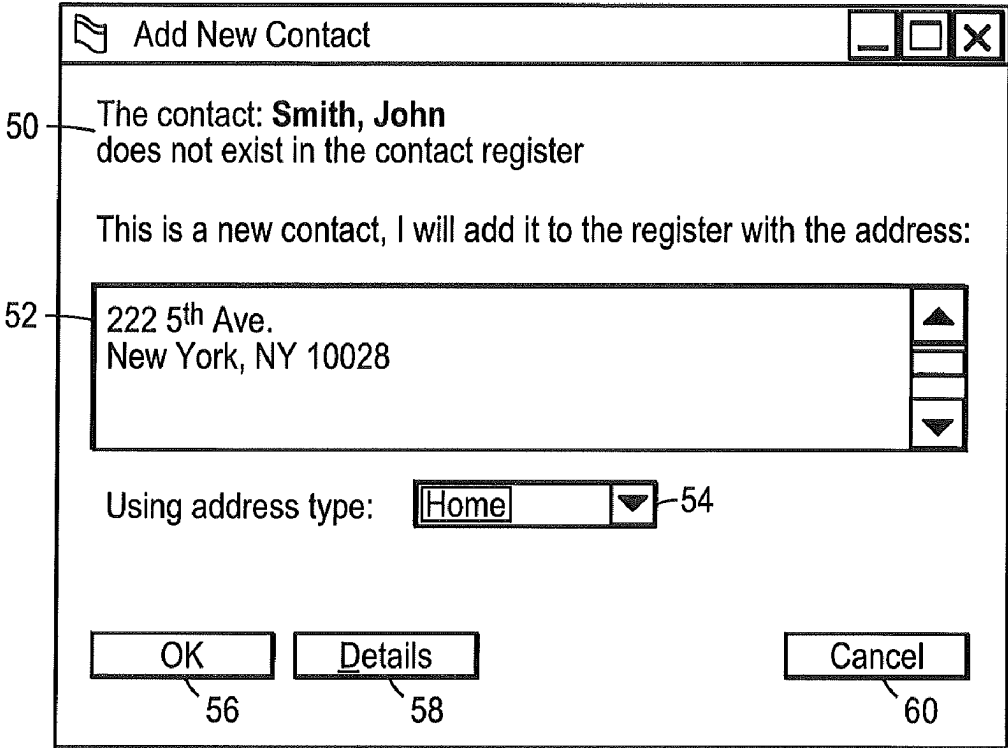


FIG. 6

Arendi OneButton Contact Register

Name

Title:

Address type: Home 54

Street: 222 5th Ave.

City: New York

State/Province: NY

ZIP/Postal: 10028

Country:

First: John

Middle:

Last: Smith

Suffix:

Company:

62

64 Add and Choose

66 Options...

60 Cancel

Dette er en test

FIG. 7

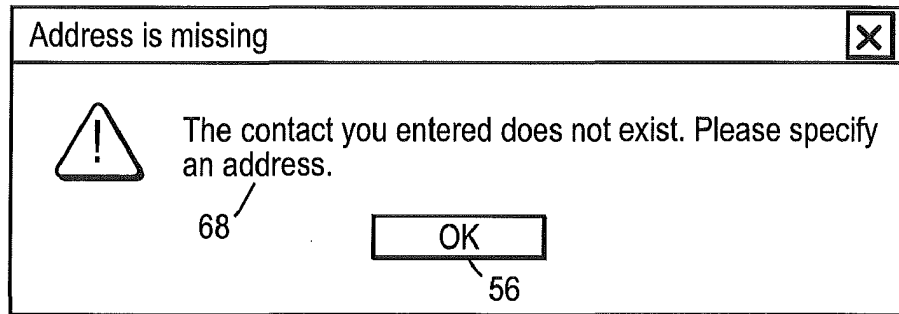


FIG. 8

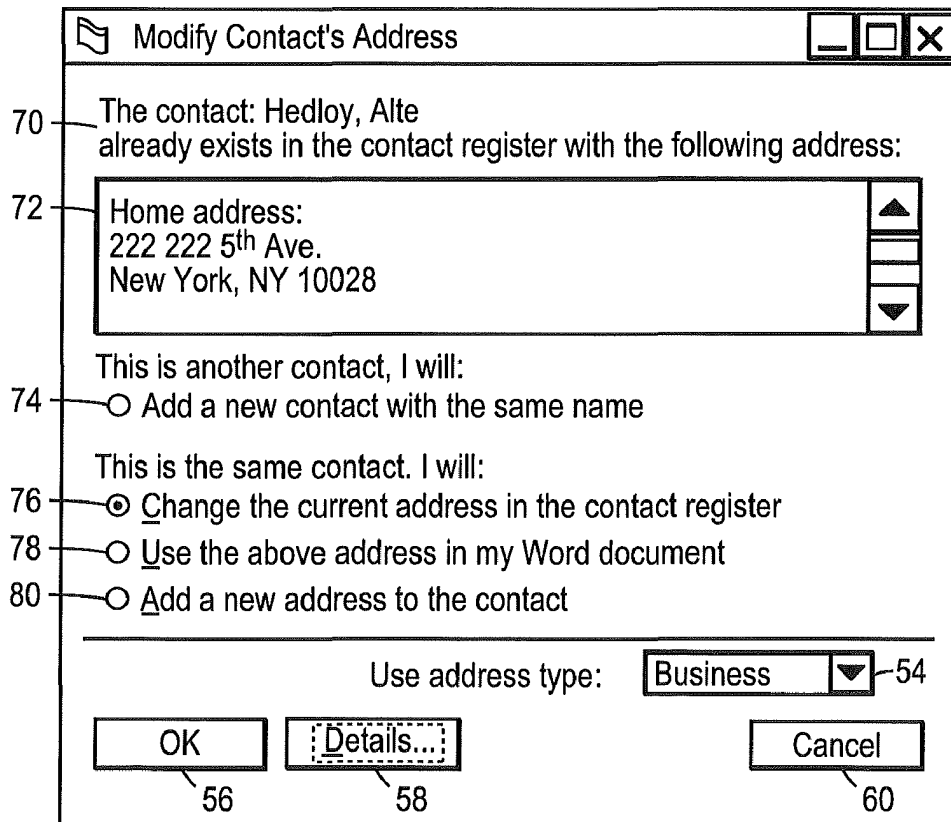


FIG. 9

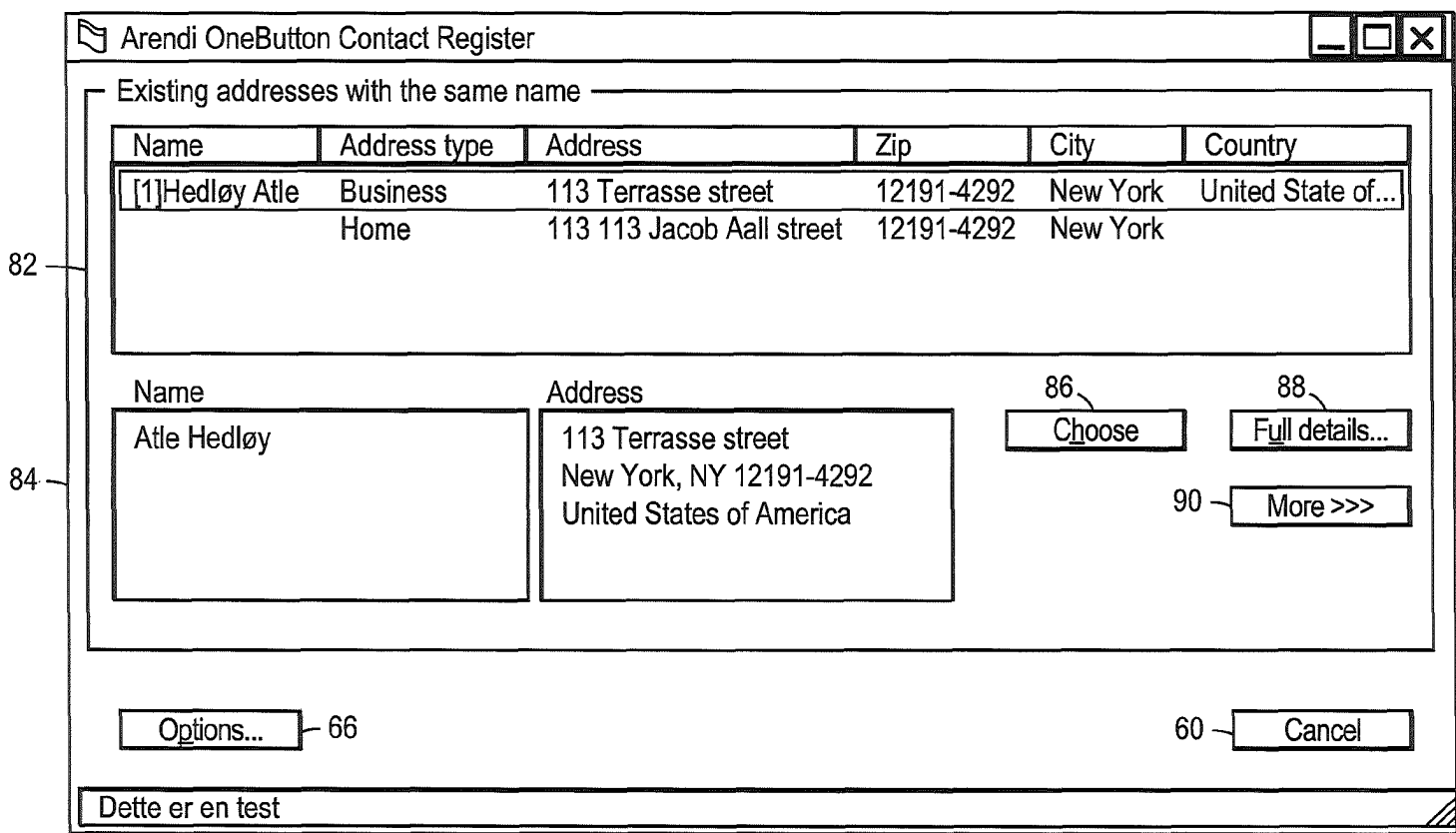


FIG. 10

The screenshot shows a software window titled "Arendi OneButton Contact Register". It features a table of existing addresses and a form for adding a new one.

Name	Address type	Address	Zip	City	Country
[1] Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Below the table, there is a summary view for the selected address:

Name: Atle Hedløy
 Address: 113 Terrasse street, New York, NY 12191-4292, United States of America

Buttons: Choose (86), Full details... (88), <<< Less (90)

The main form for adding a new address includes:

Name: Title, First (Atle), Middle, Last (Hedløy), Suffix, Company

Address type: Home (54)

Street: 151 University Ave.

City: Palo Alto

State/Province: CA

ZIP/Postal: 94301-1632

Country: USA

Buttons: Add and Choose (64), Add this address to the selected options above (92), Options... (66), Cancel (60)

Footer text: Dette er en test

FIG. 11

The image shows a screenshot of a contact management application window titled "Atle Hedløy - Contact". The window has a menu bar with "File", "Edit", "View", "Insert", "Format", "Tools", "Contact", and "Help". Below the menu bar is a toolbar with icons for "Save and Close", "Print", "Copy", "Paste", "Attach", "Print", "Delete", "Refresh", "Up", "Down", and "Help". The main content area has four tabs: "General", "Details", "Journal", and "All Fields". The "General" tab is active and contains the following fields:

- Full Name...**: Atle Hedløy (label 96)
- Job title:** [Empty]
- Company:** [Empty]
- File as:** Hedløy, Atle
- Address...**: 113 Terrasse street, New York, NY 12191-4292, United States of America (label 98)
- Business** (dropdown menu)
- This is the mailing address
- Phone:** Business, Home, Business Fax, Mobile (dropdown menus and input fields)
- E-mail** (dropdown menu and input field with icon) (label 100)
- Web page:** [Empty]
- Categories...** (input field) (label 104)
- Private**

A large empty text area is located below the "E-mail" field (label 102). The window title bar includes standard minimize, maximize, and close buttons.

FIG. 12

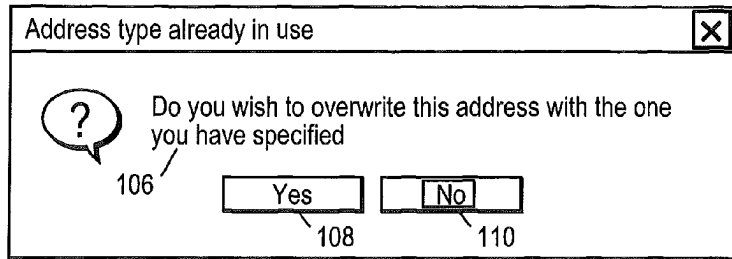


FIG. 13

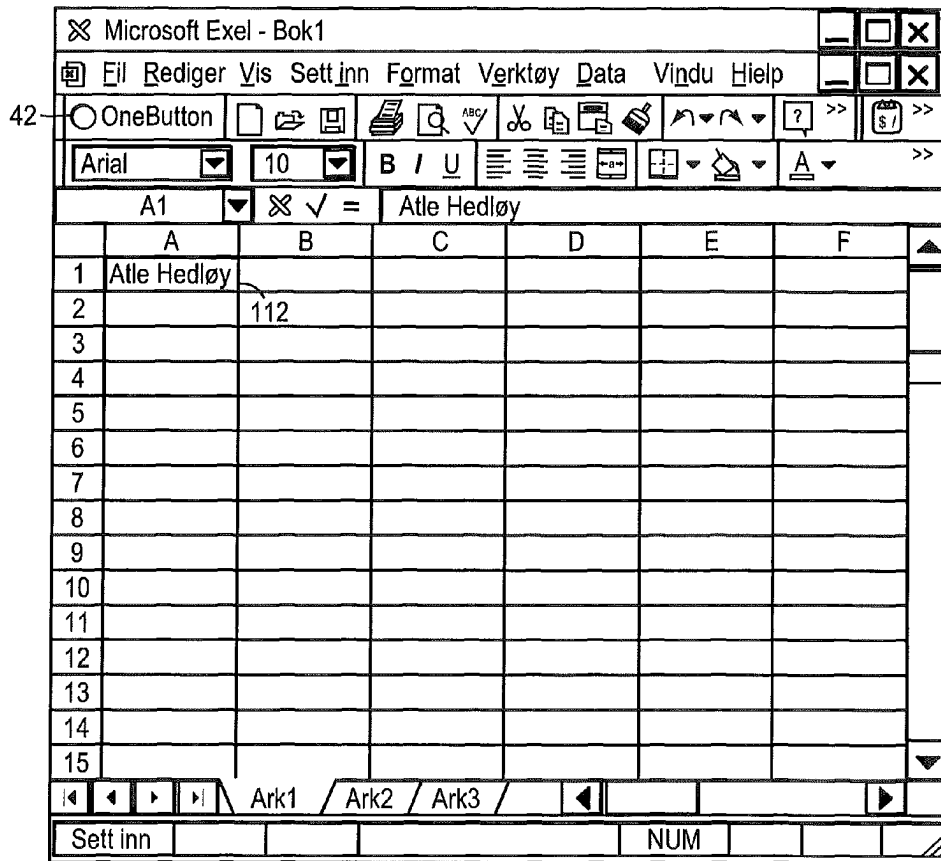


FIG. 14

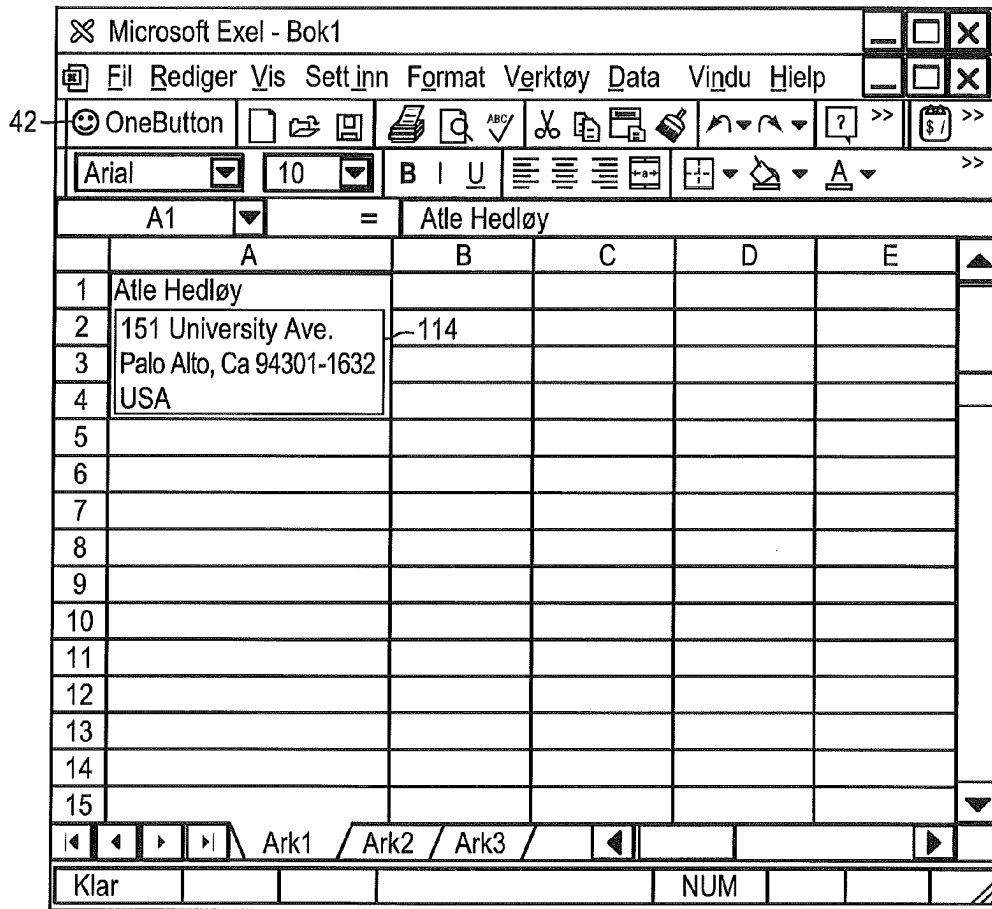
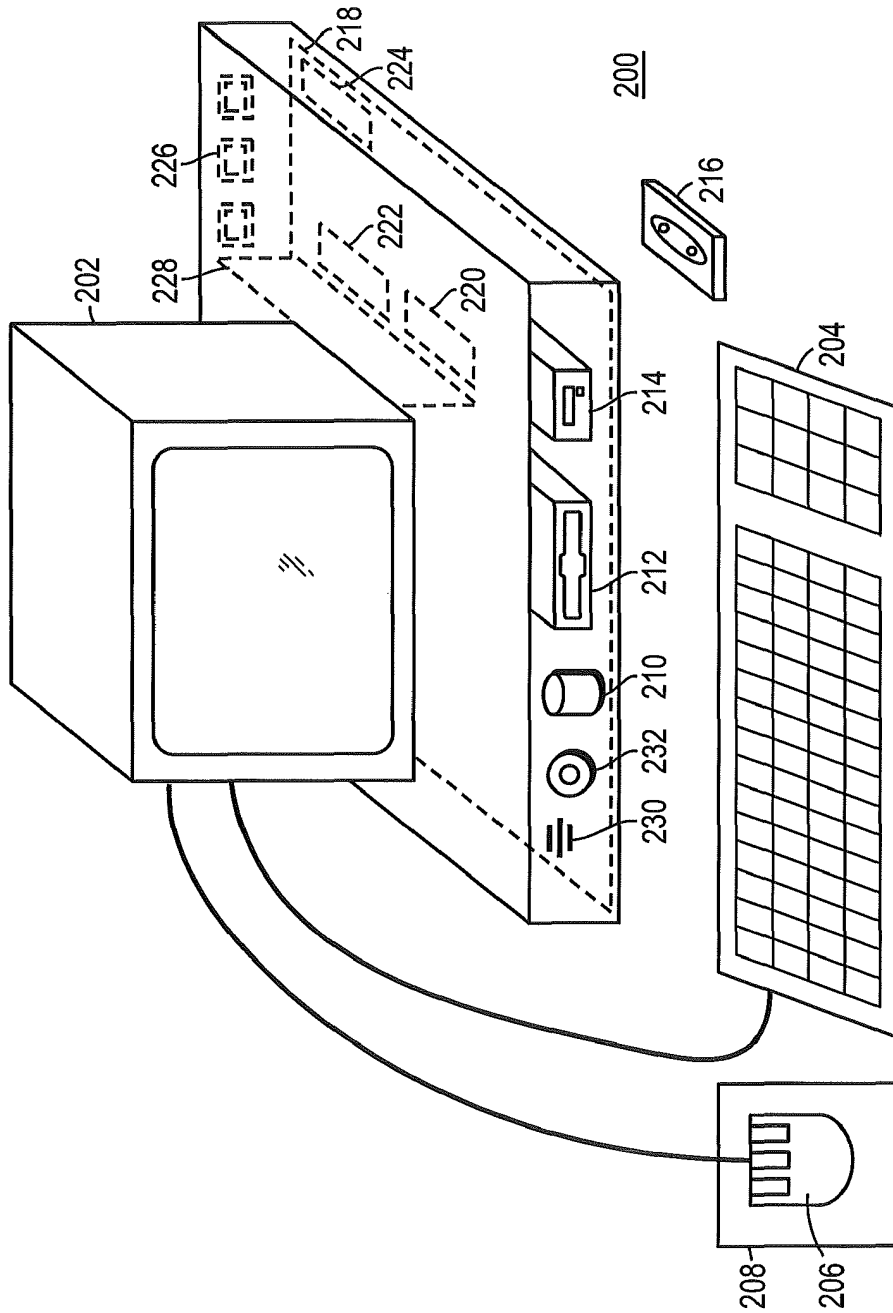


FIG. 15



US 7,917,843 B2

1

**METHOD, SYSTEM AND COMPUTER
READABLE MEDIUM FOR ADDRESSING
HANDLING FROM A COMPUTER PROGRAM**

CROSS REFERENCES TO RELATED
APPLICATIONS

The present application is a continuation of U.S. application Ser. No. 09/923,134 filed on Aug. 6, 2001, which in turn is a continuation of U.S. application Ser. No. 09/189,626 filed on Nov. 10, 1998, and which issued as U.S. Pat. No. 6,323,853. Each of the above-described applications is hereby incorporated by reference in their entireties.

BACKGROUND OF THE INVENTION

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 as 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.

However, the information in the database must constantly be updated by the user. This requires the user to learn how to use and have access to the database. In this case, a change in the information, such as change in address or a name, etc., requires the user of the word processor to implement this change in the database, or alternatively, the change is made to the database centrally by a database administrator.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device provided in the computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word process-

2

ing program, spreadsheet program, etc., using an input device, such as a touch screen, keyboard button, icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device such as a touch screen, keyboard button icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source, such as a database program, contact management program, etc.

The above and other objects are achieved according to the present invention by providing a novel method, system and computer readable medium for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

The present invention also includes a computer readable medium storing program instructions by which the method of the invention can be performed when the stored program instructions are appropriately loaded into a computer, and a system for implementing the method of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

FIG. 1 is a flow chart illustrating a method for address handling within a computer program, according to an exemplary embodiment of the present invention;

FIG. 2 is a flow chart illustrating a method for address handling within a computer program, according to another exemplary embodiment of the present invention;

FIG. 3 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 4 is a screen shot illustrating a retrieved address in a word processor, according to an exemplary embodiment of the present invention;

FIG. 5 is a screen shot illustrating the inputting of a name and address to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 6 is a screen shot illustrating an add new contact message window, according to an exemplary embodiment of the present invention;

FIG. 7 is a screen shot illustrating a contact register message window, according to an exemplary embodiment of the present invention;

US 7,917,843 B2

3

FIG. 8 is a screen shot illustrating an address missing message window, according to an exemplary embodiment of the present invention;

FIG. 9 is a screen shot illustrating a modify contact's address message window, according to an exemplary embodiment of the present invention;

FIG. 10 is a screen shot illustrating a select a contact address register message window, according to an exemplary embodiment of the present invention;

FIG. 11 is a screen shot illustrating a more detailed mode of registering an additional address for the contact register of FIG. 9, according to an exemplary embodiment of the present invention;

FIG. 12 is a screen shot illustrating a contact management program window in a full detailed mode, according to an exemplary embodiment of the present invention;

FIG. 13 is a screen shot illustrating an address already in use message window, according to an exemplary embodiment of the present invention;

FIG. 14 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a spreadsheet, according to an exemplary embodiment of the present invention;

FIG. 15 is a screen shot illustrating a retrieved address in a spreadsheet, according to an exemplary embodiment of the present invention; and

FIG. 16 is a schematic illustration of a general purpose computer for performing the processes of the present invention, according to an exemplary embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In an embodiment of the present invention, single button addressing is achieved by providing an input device, such as a touch screen, keyboard, icon, menu, voice command device, etc. (hereinafter called "button"), in a computer program, such as a word processing program, spreadsheet program, etc. (hereinafter called "word processor"), for executing address handling therein.

Accordingly, 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, such as a touch screen button, keyboard button, icon, menu choice, voice command device, etc. A program then executes and retrieves the typed information from the document, and searches an information management source, such as a database, file, database program, contact management program, etc. (hereinafter called "database") to determine if the information, such as the name or part of the name typed and searched by the program exists in the database. If the program does not find stored information, such as a name, corresponding to the name or part of the name typed, the user is asked by the program whether the information, such as the name that was not found, should be added to the database. In addition, the user may enter any other information besides the name, such as addresses, businesses, telephone numbers, fax numbers, e-mail address, etc., so that this other information can be stored in the database for later use.

If the program finds name(s) and address(es) corresponding to the part of the addressee's name typed, this additional information is automatically entered into the user's word processor, optionally with a confirmation from the user that this is the correct data. If the typed address information does

4

not correspond to data already stored in the database, after clicking on the button, the program, for example, lets the user decide: (1) if this is new data (e.g., a new address) for an existing contact; (2) if the stored data should be changed to what the user just typed; (3) if this is a new contact with the same name as the one already entered into the database; or (4) if the typed address is only to be used once, and therefore not to be stored in the database at all. If, later, for example, a name with several address stored in the database is recalled all addresses for this contact will be displayed, so that the correct address can be selected by the user.

The program may be extended to also store and retrieve other information, such as telephone numbers, fax numbers, e-mail addresses, etc. Once the program recalls the telephone numbers, fax numbers, email addresses, etc., the user can command the program to send e-mails, faxes, etc. Similarly, if the user types in the name of a mailing list, the program create merge letters, group emails, etc.

Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views, and more particularly to FIGS. 1 and 2 thereof, there is illustrated flow charts of single button addressing, according to exemplary embodiments of the present invention.

In FIG. 1, after the user has inserted the address in the word processor, the user commands the button at step 2 and the program analyzes what the user has typed in the document at step 4. At step 6, the program decides what was found in the document and if the program found nothing in the document or what it found was un-interpretable the program goes to step 8 and outputs an appropriate message to the user and then quits at step 16. The program analyzes what the user has typed in the document at step 4, for example, by analyzing (i) paragraph/line separations/formatting, etc.; (ii) street, avenue, drive, lane, boulevard, city, state, zip code, country designators and abbreviations, etc.; (iii) Mr., Mrs., Sir, Madam, Jr., Sr. designators and abbreviations, etc.; (iv) Inc., Ltd., P.C., L.L.C, designators and abbreviations, etc.; and (v) a database of common male/female names, etc.

If the program find an e-mail address mailing list/category name telephone number or other information, at step 10 an appropriate action is performed by the program and then the program execution quits at step 16. If the program only finds a name or initials, or the like, the program looks up the name in the database at step 12 and at step 18 the program determines what was found. If the program finds more than one possible contact/address match, at step 20 the program displays menu choices to the user to let him choose an appropriate answer. Then at step 22 the program inserts a correct address and name in the document and then at step 16 the program quits execution. If the program finds one match exactly, i.e., one contact with one address, the program inserts the correct address and name in the document at step 22 then quits and then quits execution at step 16. If the program does not find a name in the database, at step 24 the program prompts the user to specify an address and then quits execution at step 16. If the program at step 6 finds a name and an address, at step 14 the name is looked up in the database. Then, at step 26, if no match is found, at step 28 the program inserts an address and a name which are possibly corrected by the user into the database and then quits execution at step 16. If at step 26, the name and address is found, at step 32 the program either takes no action or displays the data for the user to edit. If at step 26, the name is found but not the address, the program prompts the user for a decision at step 30. If the user decides that this is another contact with a same name, the program goes to step 28. If the user decides that this is a one

US 7,917,843 B2

5

time occurrence, no action is taken and the program quits at step 16. If the user decides that the contact has, for example, moved and that this is a new address, at step 34 one of the old addresses for the contact is replaced with the new one and the program quits at step 16. If the user decides that this is an additional address for the contact, at step 36 the additional address is inserted into the database for that contact and execution quits at step 16.

The flowchart shown in FIG. 2 is similar to the flowchart in FIG. 1, except for some additional steps which will now be discussed. At step 6, if the program only finds a name or a similar name then the name is looked up in the database at step 12, then at step 18 if the program found more than one possible contact/address match, the program displays choices to the user to let him choose an address at step 20. Then at step 21 the user decides whether to insert the selected address into the document. If the user does not decide to select the address into the document the program quits execution at step 16. If the user decides to insert the selected address into the document the program inserts the address and name into the document at step 22 and then quits at step 16.

If the program finds a name and address in the database at step 6, then at step 14 the program looks up the name in the database and at step 26 the program determines what it has found. If the program does not find the name at step 26, at step 27 the program prompts the user for a decision and review and whether to insert the contact and address. If the user does not decide to insert the contact address, the program quits at step 16. If the user decides to insert the contact address, at step 28 the program inserts the address and name which may be possibly corrected by the user for program in the database then execution quits at step 16.

If at step 26 the program finds a name and not an address, then at step 29 the name is looked up in the database. Then at step 31 the program decides whether this contact has another address. If the contact does not have another address, at step 33 the program prompts the user for a decision and review and whether to add the address. If the user does not want to add the address at step 33, the program quits at step 16. If the user wants to add the address at step 33 because this is an additional address for the contact, at step 36 the address is inserted in the database for the contact and execution quits at step 16.

At step 30, if the user decides that this is another contact with the same name, then the program goes to step 28. If at step 30 the user decides that this is a one time occurrence, then the program quits at step 16. If at step 30, the user decides that the contact has, for example, moved, the program goes to step 34. If at step 30, the user decides that this is an additional address for the contact, at step 36 the program inserts the address in the database for the contact and then quits at step 16.

Various exemplary screen shots which are generated during execution of the program, according to the present invention, will now be described with reference to FIGS. 3-15 and examples 1-7 as follows.

Example 1

Retrieving an Existing Address from the Database

FIG. 3 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name 40. The user hits the button 42, for example, marked "OneButton" and the program according to the present invention retrieves the name 40 from the document, searches a

6

database for the name 40, and inserts the retrieved address 44 associated with the name 40 into the document as shown in, for example, FIG. 4.

The above example corresponds to steps 2, 4, 6, 12, 18, 22 and 16 in the flow charts of FIGS. 1 and 2.

Example 2

Adding a New Contact to the Database

FIG. 5 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name and address of a new contact 46. The user commands the button 42; for example, marked "OneButton," and the program according to the invention retrieves the new contact 46 from the document, searches a database for the name of the new contact 46 and generates a screen as shown in, for example, FIG. 6. This screen includes a message 50 informing the user that the new contact does not exist in the database, a message 52 including the address retrieved from the document, an address type selection 54, such as home, business, etc., and "OK," "Details," and "Cancel" buttons 56, 58, and 60, respectively.

At this point, the user can cancel the operation by commanding the Cancel button 60, ask the program to store data in the database and return the document by commanding the OK button 56, or check details before storing data into the database by commanding the Details button 58. If the user commands the Details button 58, as shown in, for example, FIG. 7, a message screen is provided so that the user can review and edit data 62 and the selection 54, store the data 62 and 54 in the database by commanding a "Add and Choose" button 64, see more options by commanding an "Options" button 66, or cancel the operation by commanding the Cancel button 60.

The above example corresponds to steps 2, 4, 6, 14, 26, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 27, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 26, 27, 28 and 16 in the flow chart of FIG. 2.

Example 3

Try to Retrieve Existing Address, but Contact is not in Database

FIG. 3 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name of a contact 40. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the name 40 from the document, searches a database for the name of the contact 40 and generates a screen as shown in, for example, FIG. 8. This screen includes a message 68 informing the user that the contact does not exist in the database and to specify an address, and "OK" buttons 56. At this point when the user commands the OK button 56, the user returns to the document so that the contact's address can be included as in Example 2 above.

The above example corresponds to steps 2, 4, 6, 12, 18, 24, and 16 in the flow of charts of FIGS. 1 and 2.

Example 4

Adding a New Address for an Existing Contact (Short Version)

FIG. 4 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed

US 7,917,843 B2

7

a name and new address of an existing contact **44**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **44** from the document, searches a database for the same name of the existing contact **44** and generates a screen as shown in, for example, FIG. **9**. This screen includes a message **70** informing the user that the contact already exists in the database with an existing address, a message **72** including the existing address, add new contact with same name selection **74**, change existing address selection **76**, use existing address in document selection **78**, add the new address contact selection **80**, the address type selection **54**, such as home, business, etc., and the "OK," "Details," and "Cancel" buttons **56**, **58** and **60** respectively. At this point, the user may select one of the four options **74-80**, and command the OK button **56** to execute the selected options. The user can also cancel the operation by commanding the Cancel button **60**, or check details before storing data into the database by commanding the Details button **58**.

The above example corresponds to steps **2, 4, 6, 14, 26, 28, 30, 34, 36**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 14, 26, 29, 31, 30, 28, 34, 36**, and **16** in the flow chart of FIG. **2**.

Example 5

Selecting Between Several Possible Matching Addresses

FIG. **3** illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and possibly address of at least one existing contact **40**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **40** from the document, searches a database for the name of the existing contact **40** and generates a screen as shown in, for example, FIG. **10**. This screen includes a message informing the user the name corresponds to several addresses and possible contacts which already exist in the database, with existing contacts and addresses for selection **82**, a message **84** including the full name and address for the contact that the user selects in **82**; the Options button **66**, a "Choose" button **86**, a "Full details" button **88**, a "More >>>" button **90**, and the Cancel button **60**. The above screen indicates to the user that at least one contact with the same name exists, and that there are more than one addresses and/or contacts that match.

At this point the user may command the Choose button **86** to use the selected address and return to the document, or the user may command the More >>> button **90** to view how the program interpreted what the user typed in the word processor, and possibly change this data, wherein the program generates an updated screen as shown in, for example, FIG. **11**. The updated screen includes the data **62** which displays the name typed in the word processor as interpreted by the program, address fields, and the fields for the address type selection **54**, such as home, business, etc., which may be changed by the user before the program stores it in the database, the Add and Choose button **64**, a "<<< Less" button **90** corresponding to the More >>> button **90** for returning to the screen of FIG. **10**, and an "Add this address to the selected contact above" button **92**. The user might then command the Add this address to the selected contact above button **92** and the result in the word processor is illustrated in FIG. **4**. The user can also cancel the operations by commanding the Cancel button **60**, or command the add choose button **64** to add this name and address as a new contact and address, or open

8

the database before storing data into the database by commanding a "Full details" button **88** as will be later described.

The above example corresponds to steps **2, 4, 6, 12, 18, 20, 22**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 12, 18, 20, 21, 22**, and **16** in the flow chart of FIG. **2**.

Example 6

Adding a New Address for an Existing Contact (Long Version)

FIG. **4** illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and new address of an existing contact **44**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **44** from the document, searches a database for the name of the existing contact **44** and generates a screen as shown in, for example, FIG. **9**. As previously described the screen includes a message **70** informing the user that the contact already exists in the database with an existing address, and the user may command the Details button **58** to see the details of the new address for potentially modify the details before they are stored in the database and the program generates a screen as shown in, for example, FIG. **10**. From this screen, the user may choose to use another address than the one he typed, and return to the document, or the user may command the "Full details" button **88** to enter a database program, such as OUTLOOK™, directly as shown in, for example, FIG. **12**. In FIG. **12**, the database program, such as OUTLOOK™, may include portions **94-104** for allowing the user to modify various pieces of data before they are stored in the database.

Alternatively, in the screen shown in FIG. **10**, the user may command the More >>> button **90** at which time the program generates the screen as shown in, for example, FIG. **11** and as previously described. In this screen, the user might then command the Add this address to the selected contact above button **92**. If the address typed is already in use, the program generates a screen including a message **106**, and "Yes" and "No" buttons, **108** and **110**, respectively, as shown in, for example, FIG. **13**. If the user hits the Yes button **108** the program overwrites the contact address with the address specified by the user (e.g., if the contact has moved) and the result in the word processor is shown in, for example FIG. **4**.

The above example corresponds to steps **2, 4, 6, 12, 14, 26, 28, 30, 34, 36**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 12, 14, 26, 29, 31, 30, 28, 34, 36**, and **16** in the flow chart of FIG. **2**.

Example 7

Spreadsheet Application

FIG. **14** illustrates a starting point in word processor document, such as an EXCEL spreadsheet, wherein the user has typed a name **112**. The user hits the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the name **112** from the spreadsheet, searches a database for the name **112**, and inserts the retrieved address **114** into the spreadsheet as shown in, for example, FIG. **15**. Accordingly, the examples 1-6 apply not only to word processor documents, such as WORD™ documents, etc., but to other word processor documents, and spreadsheets, such as EXCEL™ spreadsheets, etc.

The above example corresponds to steps **2, 4, 12, 18, 22**, and **16** in the flow of charts FIGS. **1** and **2**.

US 7,917,843 B2

9

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.

As stated above, the system includes at least one computer readable medium or alternatively, the computer readable medium may be accessed through various paths, such as networks, internet, drives, etc. Examples of computer readable media are compact discs, hard disks, floppy disks, tape, magneto-optical disks, PROMs (EPROM, EEPROM, Flash, EPROM), DRAM, SRAM, SDRAM, etc. Stored on any one or on a combination of computer readable media, the present invention includes software for controlling both the hardware of the computer 200 and for enabling the computer 200 to interact with a human user. Such software may include, but is not limited to, device drivers, operating systems and user applications, such as development tools. Such computer readable media further includes the computer program product of the present invention for performing any of the processes according to the present invention, described above (see, e.g., FIGS. 1-15). The computer code devices of the present invention can be any interpreted or executable code mechanism, including but not limited to scripts, interpreters, dynamic link libraries, Java classes, and complete executable programs, etc.

The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

Address handling, according to this invention; is a significant simplification relative to existing methods, and requires little or no training on the part of a user, as correct addresses are retrieved with a minimal number of user commands, "clicks," keystrokes, etc. In addition, a program according to the present invention, can be programmed and created in most existing programming languages and be connected to most modern word processors. Therefore, according to the present invention, the process of creating and updating records in an address database is significantly simplified, since this may now be performed directly from the word processor.

Although the present invention is defined in terms of word processing documents, such as WORD™ documents and EXCEL™ spreadsheets, the present invention is applicable to all types of word processing documents such as NOTEPAD™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™, etc., as will be readily apparent to those skilled in the art.

10

Although the present invention is defined in terms of information management or database programs, such as OUTLOOK™ etc., the present invention is applicable to all types of information management or database programs such as ACCESS™, ORACLE™, DBASE™, RBASE™, CARD-FILE™, including "flat files," etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of providing an input device, such as a button 42 in a word processor for address handling therein, the present invention may be practiced with all types of input devices, such as touch screen, keyboard button, icon, menu, voice command device, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving information from a document before searching a database, the user may select the information in the document to be searched by the program in the database (e.g., by highlighting, selecting, italicizing, underlining, etc.), as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving a name or portion thereof from a document before searching a database the program may retrieve an address or portion thereof from the document before searching the database and insert, correct, complete, etc., the retrieved address based on the information found in the database corresponding to the retrieved address or portion thereof, as will be readily apparent to those skilled in the art.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of appended claims, the invention may be practiced otherwise than as specifically described herein.

This application claims priority and contains subject matter related to Norwegian patent application No. 984066 filed on Sep. 3, 1998, the entire contents of which are hereby incorporated by reference.

What is claimed is:

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:
 - displaying the document electronically using the first computer program;
 - 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;
 - retrieving the first information;
 - 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;
 - 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
 - if searching finds any second information related to the search term, performing the action using at least part of

US 7,917,843 B2

11

the second information, wherein the action is of a type depending at least in part on the type or types of the first information.

2. A method according to claim 1, wherein the first information comprises at least one of name-, person-, company- and address-related information.

3. A method according to claim 2, wherein performing the action includes performing the action in the first computer program.

4. A method according to claim 1, wherein performing the action includes performing the action in the first computer program.

5. A method according to claim 4, wherein performing the action includes causing addition of at least part of the second information to the first information in the document.

6. A method according to claim 4, wherein performing the action includes causing display of at least part of the second information.

7. A method according to claim 4, wherein performing the action includes causing display of at least part of the second information by the first computer program.

8. A method according to claim 1, further comprising, providing a prompt for updating the information source to include the first information.

9. A method according to claim 1, further comprising, if the search is not successful, providing a prompt for updating the information source to include the first information.

10. A method according to claim 1, wherein receipt by the first computer program of the user command precedes analyzing the document.

11. A method according to claim 1, wherein analyzing the document is completed after the receipt of the user command is completed and before searching is initiated.

12. A method according to claim 1, wherein the input device is a graphical input device.

13. A method according to claim 1, wherein the user command is the only command from a user necessary to initiate performing the operation.

14. A method according to claim 1, wherein the input device is a menu and the entry of the user command includes a user's selection of the menu and click on a menu choice from the menu.

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.

16. A method according to claim 1, wherein the information source is associated with the second computer program and is available on the computer.

17. A method according to claim 1, wherein the information source is associated with the second computer program and is available through the computer.

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.

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.

20. 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: displaying the document electronically using the first computer program; while the document is being displayed, analyzing, in a computer process on the computer, first information from the document to determine if the first information is at least one of a plurality of types of information that

12

can be searched for in order to find second information related to the first information, and wherein the first information comprises at least one of name-, person-, company-, and address-related information;

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 a user editable information source outside 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, wherein the input device includes a menu;

retrieving the first information;

in consequence of receipt by the first computer program of the user command, such user command including a user's selection of the menu and click on a menu choice from the menu, causing a search for the search term in the user editable information source, using a second computer program, in order to find second information related to the search term in the user editable information source; and

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 and performing the action includes at least causing display of at least part of the second information.

21. A method according to claim 20, further comprising, if searching results in a plurality of occurrences of second information, causing display of such instances to enable user selection of one of them for use in performing the action.

22. A method according to claim 20, wherein performing the action includes causing addition of at least part of the second information to the first information in the document.

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: displaying the document electronically using the first computer program; 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;

retrieving the first information;

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;

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,

US 7,917,843 B2

13

using a second computer program, in order to find second information related to the search term; and 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.

24. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the first information comprises at least one of name-, person-, company- and address-related information.

25. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein performing the action includes performing the action in the first computer program.

26. At least one non-transitory computer readable medium according to claim 24, the instructions establishing processes wherein performing the action includes performing the action in the first computer program.

27. At least one non-transitory computer readable medium according to claim 25, the instructions establishing processes wherein performing the action includes causing addition of at least part of the second information to the first information in the document.

28. At least one non-transitory computer readable medium according to claim 25, the instructions establishing processes wherein performing the action includes causing display of at least part of the second information.

29. At least one non-transitory computer readable medium according to claim 25, the instructions establishing processes wherein performing the action includes causing display of at least part of the second information by the first computer program.

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.

31. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes comprising:

if the search is not successful, providing a prompt for updating the information source to include the first information.

32. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein receipt by the first computer program of the user command precedes analyzing the document.

33. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein analyzing the document is completed after the receipt of the user command is completed and before searching is initiated.

34. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the input device is a graphical input device.

35. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the user command is the only command from a user necessary to initiate performing the operation.

36. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the input device is a menu and the entry of the user command includes a user's selection of the menu and click on a menu choice from the menu.

14

37. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes 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.

38. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the information source is associated with the second computer program and is available on the computer.

39. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein the information source is associated with the second computer program and is available through the computer.

40. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein performing the action includes causing insertion of at least part of the second information into the document.

41. At least one non-transitory computer readable medium according to claim 23, the instructions establishing processes wherein performing the action includes causing insertion of at least part of the second information into the document by the first computer program.

42. 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:

displaying the document electronically using the first computer program;

while the document is being displayed, analyzing, in a computer process on the computer, 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, and wherein the first information comprises at least one of name-, person-, company-, and address-related information;

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 a user editable information source outside 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, wherein the input device includes a menu;

retrieving the first information;

in consequence of receipt by the first computer program of the user command, such user command including a user's selection of the menu and click on a menu choice from the menu, causing a search for the search term in the user editable information source, using a second computer program, in order to find second information related to the search term in the user editable information source; and

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 and performing the action includes at least causing display of at least part of the second information.

US 7,917,843 B2

15

43. At least one non-transitory computer readable medium according to claim 42, the instructions establishing processes comprising:

if searching results in a plurality of occurrences of second information, causing display of such instances to enable user selection of one of them for use in performing the action.

16

44. At least one non-transitory computer readable medium according to claim 42, the instructions establishing processes wherein performing the action includes causing addition of at least part of the second information to the first information in the document.

* * * * *

EXHIBIT 2

(12) **United States Patent**
Hedloy

(10) **Patent No.:** **US 8,306,993 B2**
(45) **Date of Patent:** ***Nov. 6, 2012**

(54) **METHOD, SYSTEM AND COMPUTER READABLE MEDIUM FOR ADDRESSING HANDLING FROM AN OPERATING SYSTEM**

4,969,097 A 11/1990 Levin 364/419
4,972,349 A 11/1990 Kleinberger 364/900
5,226,117 A 7/1993 Miklos 395/157
5,267,155 A 11/1993 Buchanan et al. 364/419.14

(75) Inventor: **Atle Hedloy**, Madrid (ES)

(Continued)

(73) Assignee: **Arendi S.A.R.L.**, Luxembourg (LU)

FOREIGN PATENT DOCUMENTS

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

EP 0 093 250 A2 11/1983

(Continued)

This patent is subject to a terminal disclaimer.

OTHER PUBLICATIONS

PD Turney—National Research Council, Institute for Information Learning to Extract Keyphrases from Text. Feb. 1999.*

(21) Appl. No.: **11/745,186**

(Continued)

(22) Filed: **May 7, 2007**

(65) **Prior Publication Data**

Primary Examiner — Khanh Pham

US 2007/0244907 A1 Oct. 18, 2007
US 2010/0211600 A2 Aug. 19, 2010

(74) *Attorney, Agent, or Firm* — Sunstein Kann Murphy & Timbers LLP

Related U.S. Application Data

(63) Continuation of application No. 09/390,303, filed on Sep. 3, 1999, now Pat. No. 7,272,604.

(30) **Foreign Application Priority Data**

Sep. 3, 1998 (NO) 984066

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/769; 707/767**

(58) **Field of Classification Search** **707/767, 707/769**

See application file for complete search history.

(56) **References Cited**

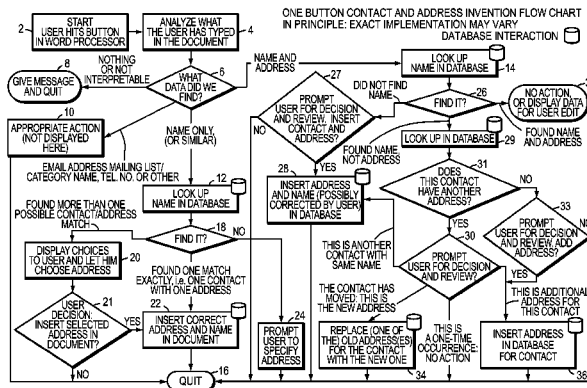
U.S. PATENT DOCUMENTS

3,350,695 A 10/1967 Kaufman et al. 340/172.5
4,270,182 A 5/1981 Asija 364/900
4,453,217 A 6/1984 Boivie 364/300
4,674,065 A 6/1987 Lange et al.

(57) **ABSTRACT**

A method, system and computer readable medium for information handling within an operating system, including providing a record retrieval program; providing an input device within a window or screen of the operating system and configured to enter an execute command which initiates a record retrieval from local and remote information sources using the record retrieval program; using the record retrieval program to enter first information into search fields provided in the record retrieval program; entering the execute command using the input device after the step of entering the first information; searching, using the record retrieval program, the local and remote information sources for second information associated with the first information; and displaying the second information in the record retrieval program, when one of the local and remote information sources includes second information associated with the first information.

30 Claims, 17 Drawing Sheets



US 8,306,993 B2

Page 2

U.S. PATENT DOCUMENTS

5,331,555	A	7/1994	Hashimoto et al.	364/419.07	6,026,398	A	2/2000	Brown et al.	
5,355,472	A	10/1994	Lewis	395/600	6,026,410	A	2/2000	Allen et al.	707/104
5,375,200	A	12/1994	Dugan et al.	395/159	6,028,605	A *	2/2000	Conrad et al.	715/840
5,388,150	A	2/1995	Schneyer et al.	379/67	6,055,531	A *	4/2000	Bennett et al.	707/5
5,392,386	A	2/1995	Chalas		6,065,012	A *	5/2000	Balsara et al.	707/102
5,416,901	A	5/1995	Torres	395/159	6,067,565	A	5/2000	Horvitz	
5,491,783	A	2/1996	Douglas et al.	395/159	6,073,138	A *	6/2000	de l'Etraz et al.	707/104.1
5,491,784	A	2/1996	Douglas et al.	395/159	6,081,788	A	6/2000	Appleman et al.	705/14
5,491,785	A	2/1996	Robson et al.	395/162	6,085,201	A	7/2000	Tso	
5,493,105	A	2/1996	Desai	235/375	6,085,226	A	7/2000	Horvitz	
5,495,565	A	2/1996	Millard et al.	395/146	6,092,074	A	7/2000	Rodkin et al.	707/102
5,500,859	A	3/1996	Sharma et al.	370/81	6,108,686	A *	8/2000	Williams, Jr.	709/202
5,530,853	A	6/1996	Schell et al.	395/600	6,151,584	A	11/2000	Papierniak et al.	705/10
5,546,447	A	8/1996	Skarbo et al.	379/142	6,178,411	B1 *	1/2001	Reiter	705/408
5,566,330	A	10/1996	Sheffield	395/600	6,182,133	B1	1/2001	Horvitz	
5,576,955	A	11/1996	Newbold et al.		6,189,026	B1	2/2001	Birrell et al.	709/206
5,579,467	A	11/1996	Capps	395/768	6,192,380	B1	2/2001	Light et al.	707/505
5,587,911	A	12/1996	Asano et al.	364/444.2	6,208,975	B1	3/2001	Bull et al.	705/14
5,606,712	A	2/1997	Hidaka	395/800	6,209,005	B1	3/2001	Harker et al.	707/501
5,623,652	A *	4/1997	Vora et al.	707/10	6,223,570	B1	5/2001	Horvitz et al.	
5,628,004	A	5/1997	Gormley et al.	395/615	6,247,029	B1	6/2001	Kelley et al.	707/507
5,634,121	A	5/1997	Tracz et al.	395/602	6,253,188	B1	6/2001	Witek et al.	705/14
5,640,565	A	6/1997	Dickinson	395/683	6,260,035	B1	7/2001	Horvitz et al.	
5,666,502	A	9/1997	Capps	345/352	6,262,730	B1	7/2001	Horvitz et al.	
5,696,962	A *	12/1997	Kupiec	707/4	6,285,987	B1	9/2001	Roth et al.	705/27
5,708,804	A	1/1998	Goodwin et al.	395/603	6,304,881	B1 *	10/2001	Halim et al.	707/201
5,724,597	A	3/1998	Cuthbertson et al.		6,317,761	B1	11/2001	Landsman et al.	707/513
5,732,229	A	3/1998	Dickinson		6,323,853	B1	11/2001	Hedloy	
5,732,395	A	3/1998	Silverman	704/260	6,349,299	B1 *	2/2002	Spencer et al.	707/702
5,734,871	A	3/1998	Kleinerman et al.	395/500	6,373,940	B2	4/2002	Shaffer et al.	379/355.02
5,737,726	A	4/1998	Cameron et al.	705/7	6,377,965	B1 *	4/2002	Hachamovitch et al.	715/203
5,740,433	A *	4/1998	Carr et al.	707/202	6,381,593	B1 *	4/2002	Yano et al.	707/3
5,742,905	A	4/1998	Pepe et al.	455/461	6,385,592	B1	5/2002	Angles et al.	705/14
5,754,636	A	5/1998	Bayless et al.	379/142	6,388,957	B2	5/2002	Yankowski	
5,761,656	A	6/1998	Ben-Shachar		6,405,206	B1 *	6/2002	Kayahara	707/102
5,761,663	A	6/1998	Lagarde et al.	707/10	6,421,691	B1	7/2002	Nishiyama et al.	707/507
5,774,887	A	6/1998	Wolff et al.	707/1	6,421,693	B1	7/2002	Nishiyama et al.	707/507
5,778,367	A	7/1998	Wesinger et al.	707/10	6,442,540	B2 *	8/2002	Sako et al.	707/3
5,781,189	A	7/1998	Holleran et al.	345/335	6,499,021	B1	12/2002	Abu-Hakima	706/10
5,787,451	A	7/1998	Mogilevsky	707/533	6,523,022	B1 *	2/2003	Hobbs	707/3
5,793,972	A	8/1998	Shane	395/200.49	6,651,217	B1	11/2003	Kennedy et al.	715/507
5,794,228	A	8/1998	French et al.		6,711,585	B1 *	3/2004	Copperman et al.	707/104.1
5,794,259	A	8/1998	Kikinis	707/507	6,725,227	B1 *	4/2004	Li	707/102
5,799,302	A	8/1998	Johnson et al.		6,738,615	B1	5/2004	Chow et al.	455/415
5,805,810	A	9/1998	Maxwell	395/200.36	6,782,510	B1	8/2004	Gross et al.	715/533
5,805,886	A	9/1998	Skarbo et al.	395/685	6,826,407	B1	11/2004	Helferich	455/466
5,805,911	A	9/1998	Miller	395/796	6,870,828	B1 *	3/2005	Giordano, III	370/352
5,815,830	A	9/1998	Anthony		7,039,599	B2	5/2006	Merriman et al.	705/14
5,819,273	A *	10/1998	Vora et al.	707/10	7,051,019	B1	5/2006	Land et al.	
5,826,257	A	10/1998	Snelling, Jr.	707/4	7,136,853	B1	11/2006	Kohda et al.	707/6
5,835,089	A	11/1998	Skarbo et al.	345/335	7,149,761	B2 *	12/2006	Cooke et al.	707/203
5,859,636	A	1/1999	Pandit		7,155,451	B1	12/2006	Torres	707/104.1
5,860,073	A	1/1999	Ferrel et al.	715/255	7,216,292	B1	5/2007	Snapper et al.	715/507
5,864,848	A	1/1999	Horvitz et al.		7,272,604	B1	9/2007	Hedloy	707/10
5,873,107	A	2/1999	Borovoy et al.		7,353,246	B1 *	4/2008	Rosen et al.	709/202
5,873,108	A	2/1999	Goyal et al.	707/507	7,496,854	B2	2/2009	Hedloy	715/780
5,884,309	A	3/1999	Vanechanos, Jr.	707/10	7,505,974	B2	3/2009	Gropper	707/10
5,893,093	A	4/1999	Wills	707/5	7,509,349	B2	3/2009	Celik	707/200
5,896,321	A *	4/1999	Miller et al.	365/189.15	7,917,843	B2	3/2011	Hedloy	715/230
5,896,533	A	4/1999	Ramos et al.	395/680	7,921,356	B2	4/2011	Hedloy	715/230
5,907,838	A	5/1999	Miyasaka et al.	707/4	2002/0078030	A1 *	6/2002	Iwayama et al.	707/1
5,913,214	A	6/1999	Madnick et al.	707/10	2002/0188683	A1	12/2002	Lytle et al.	709/206
5,923,848	A	7/1999	Goodhand et al.	395/200.49	2003/0033288	A1	2/2003	Shanahan et al.	707/3
5,924,074	A	7/1999	Evans		2003/0084074	A1 *	5/2003	Balogh et al.	707/201
5,924,090	A	7/1999	Krellenstein	707/5	2003/0167279	A1 *	9/2003	Smiga et al.	707/102
5,926,808	A	7/1999	Evans et al.	707/3	2006/0047644	A1	3/2006	Bocking et al.	707/4
5,930,471	A	7/1999	Milewski et al.	395/200.04	2008/0313159	A1	12/2008	Hedloy	707/3
5,930,764	A	7/1999	Melchione et al.	705/10	2010/0281354	A1	11/2010	Hedloy	715/220
5,940,484	A	8/1999	DeFazio et al.	379/142	2011/0072029	A1	3/2011	Hedloy	707/758
5,946,647	A	8/1999	Miller et al.						
5,946,679	A	8/1999	Ahuja et al.	707/3					
5,999,938	A	12/1999	Bliss et al.						
6,006,218	A	12/1999	Breese et al.						
6,006,351	A	12/1999	Peretz et al.	714/751					
6,012,075	A	1/2000	Fein et al.	707/540					
6,018,571	A	1/2000	Langlois et al.	379/207					
6,021,403	A	2/2000	Horvitz et al.						
6,021,412	A *	2/2000	Ho et al.	707/104.1					

FOREIGN PATENT DOCUMENTS

EP	0520926	12/1992
EP	1630642 A1	1/2006
GB	2196454 A	4/1988
GB	2 318 703	4/1998
JP	08116362	5/1996
WO	WO 97/21183	6/1997

US 8,306,993 B2

Page 3

WO	WO 98/16890	4/1998
WO	WO 98/19259	5/1998
WO	WO 98/34391	8/1998

OTHER PUBLICATIONS

Steve Hitchcock et al. Towards universal linking for electronic journals. Pub. 1998.*

Addressing the Issues, Jun. 24, 1993.

Philip Robinson, The Envelope, Please: It's AddressMate, San Jose Mercury News, Sunday, Jul. 19, 1992.

Steve Supkoff, AddressMate, PCM, Aug. 1992.

L.R. Shannon, Addressing Envelopes, The New York Times, Tuesday, Jul. 14, 1992.

Rob Schwabach, Addressing for Success, On Computers column, Mar. 16, 1992.

Infoworld, p. 15, Mar. 16, 1992.

Colvin's beta-testing of Addressmate sparks developer's appreciation, Westview, Apr. 6, 1992.

Press Release: AddressMate Software Automatically Addresses and bar Codes Envelopes to Save Time and money, Mar. 9, 1992.

Gearge M> Long, Letter to Mr. David Block, Nov. 11, 1994.

Philip J. Damiano, Letter to Mr. David Block, Jan. 5, 1994.

Ed Kahn, Envelope Addressing Finally Simplified AddressMate Does Addressing and Much More, Microtimes, Nov. 27, 1995.

Kirsten Bernthal, LabelWriter XL Plus, PC Catalog, Aug. 18, 1995.

David Plotkin, Address for Success, Bay Area Computer Currents, p. 36-38, Mar. 21, 1995.

Getting Results with Microsoft Office for Windows 95, Version 7.0, 1995.

Getting Results with Microsoft Office 97, copyright 1995-1997.

Microsoft Word, User's Guide, Version 6.0, copyright 1993-1994.

Apple Internet Address Detectors User's Manual, copyright 1997.

Eve Wilson, Links and Structures in Hypertext Databases for Law, in Hypertext: Concepts, Systems and Applications, Proceedings of the First European Conference on Hypertext, Nov. 1990.

Eve Wilson, Guiding Lawyers: Mapping Law into Hypertext, Artificial Intelligence Review 6, pp. 161-189, 1992.

Charles H. Franke III et al, Authoring a Hypertext Unix Help Manual, 1995.

P.J. Brown, Guide User Manual, 1985, sixteenth impression, Apr. 1995.

E. Wilson, Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System, Literary and Linguistic Computing, vol. 5, No. 2, 1990.

Mike Langberg, Apple Breaks New Ground by Displaying What on its drawing board, Aug. 7, 1996.

E. Wilson, Integrated Information Retrieval for Law in a Hypertext Environment, Annual ACM Conference on Research and Development in information Retrieval, 1988.

Multimedia Hyperlinks Automatically Created for Reference Documents, Research Disclosure, Jun. 1993.

Joy-Lyn Blak, WikiWikiWeb, Computer World, Jan. 29, 2001.

Apple Introduces Internet Address Detectors, Press Release, Sep. 7, 1997.

Cara Cunningham, Apple Kicks Off Macworld with talk of revival, new software demos, InfoWorld Electric, Aug. 7, 1996.

James Staten, Apple Looks to the Future, MacWeek, Aug. 7, 1996.

Mark Simmons, Striking a Key Note, Mac Addict Online, Aug. 8, 1996.

Jim Miller, email regarding Apple Data Detectors, Jan. 8, 1997.

Apple Data Detectors web page, Jan. 6, 1997.

Apple Data Detectors—Now Shipping web page, Jan. 6, 1997.

The Apple Data Detectors FAQ, Jan. 6, 1997.

Developer's Guide to Apple Data Detectors, Dec. 1, 1997.

AppleScript Editors, Utilities & Environments, available at www.applescriptsourcebook.com/links/applescripteditors.html, dated Jan. 11, 2004.

Steve Tannehill, News from Jul. 1997.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 27, 2004.

Excerpt from Jury Charge Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 28, 2004.

Apple Data Detectors 1.0.2, TidBITS #419, Mar. 9, 1998.

Tonya Engst, More Context on Contextual menus, TidBITS #399, Sep. 29, 1997.

Tonya Engst, Of Mice and Menus, TidBITS #398, Sep. 22, 1997.

Charles Whaley, Will this be enough to kick-start Apple?, Computing Canada, Aug. 4, 1997.

A Farewell to the Apple Advanced Technology Group, SIGCHI, vol. 30, No. 2, Apr. 1998.

James R. Miller and Thomas Bonura, From Documents to Objects, in SIGCHI, vol. 30, No. 2, Apr. 1998.

Thomas Bonura and James R. Miller, Drop Zones, in SIGCHI, vol. 30, No. 2, Apr. 1998.

Bonnie A. Nardi, et al., Collaborative, Programmable Intelligent Agents, Communications of the ACM, vol. 41, No. 3, Mar. 1998.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 13, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 14, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 15, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 16, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 17, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 20, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 21, 2004.

Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation et al.*, Sep. 22, 2004.

Contextual Menu Manager/Apple Data Detectors (MS 049590-MS 049594), posted 2002 printed Jan. 2004.

CoStar User Manual for AddressMate and AddressMate Plus, 1995.

ctags (UNIX Command), 1993.

Microsoft Word 6.0, for Windows and Macintosh, 1993.

N.D. Beitner, et al, Multimedia Support and Authoring in Microcosm: an extended model, 1994.

Spell, iSpell Spellout, 1993.

SED(1), BSD Reference Manual Page, 1993.

AddressMate for Windows, Version 2.0, Product Box and License Agreement, 1992.

Important Note for New AddressMate Users, 1995.

Important Tips for LabelWriter Owners Using AddressMate, 1993.

AddressMate Advertisement "Whye Do it the Olde Way", 1995.

Ken Hart, Simplify Envelope Printing with AddressMate, Computer Shopper, 1995.

AddressMate for Windows, Advertisements, 1995-1996.

AddressMate Plus, Advertisement, 1995-1996.

Address Fixer for Microsoft Word and office, Product Box, 1996.

AddressMate Plus, Product Box and License Agreement, 1995.

CTAGS(1) Manual Page, 1993.

Matt Bernstein, An Apprentice that Discovers Hypertext Links, 1992.

Contextual Menu Manager/Apple Data Detectors, posted 2002 printed Jan. 2004.

P.J. Brown et al, A Help System Based on UNIX Man Pages, 1987.

John Robertson, et al, The Hypermedia Authoring Research Toolkit, ECHI 194 proceedings, pp. 177-185, 1994.

What is Wiki and Wiki History webpages, available at wiki.org/wiki.cgi?WhatIsWiki and www.c2.org/cgi/wiki?WikiHistory, last edited Jun. 2002 printed on Aug. 2003.

Wiki Wiki Origin, last edited Jun. 2003 printed on Aug. 2003.

Microsoft's Supplemental Responses to Arendi's Interrogatories, in *Arendi U.S.A. et al. v. Microsoft Corporation*, Civil Action 02-CV-343 (ECT) from United States District Court for the District of Rhode Islands, 2004.

Apple Data Detector WebPages, available on web.archive.org/web/20020601164217/www.apple.com/applescript/data_detectors, first established in 1997—last edited 2002.

Apple Data Detectors 1.0.2 Read Me, 1998.

eMailman Internet Address Detectors, 1996-2000.

Control-Click! The Archive, 1997-2000.

Contextual Menus: One of System 8's Greatest Features, in ApplePress, posted 2002.

US 8,306,993 B2

Page 4

- Contextual Menu Manager/Apple Data Detectors, available at web.archive.org/web/20020803063750/www.macemil.com/emailer/CEMH/contextual.shtml, posted 2002.
- Trygve's CMM Plug-Ins Homer, available at web.archive.org/web/19980130053511/www.bombaydigital.com/cmms, 1997.
- ADD Depot, available from web.archive.org/web/20000819091818/http://homepage.mac.com/mathemiller/add, posted 2000.
- MacOS8.com—Mac OS 8 Indepth, 1999.
- Microsoft Corporation and Dell Inc.'s Answer, Affirmative Defenses and Counterclaims to Complaint, *Arendi Holding Ltd. v. Microsoft Corp. et al.*, C.A. No. 09-119-JJF, from the United States District Court for the District of Delaware, pp. 1-17, Apr. 30, 2009.
- "More about the Database," <http://c2.com/cgi/wiki?MoreAboutTheDatabase>, pp. 1-3, last edited May 7, 2009.
- "Ward Cunningham," <http://c2.com/cgi/wiki?WardCunningham>, pp. 1-15, last edited Apr. 23, 2009.
- "Adding New Pages," <http://c2.com/cgi/wiki?AddingNewPages>, pp. 1-2, last edited Jun. 8, 2009.
- "Wiki Wiki Web," <http://c2.com/cgi/wiki?WikiWikiWeb>, 1 page, last edited Jun. 1, 2009.
- "Apple Data Detectors User's Manual," 16 pages, last updated Jul. 31, 1997.
- "ISPELL (1)," 15 pages, Aug. 23, 2003.
- "International Ispell," <http://fmg-www.cs.ucla.edu/geoff/ispell.html>, 5 pages, Aug. 23, 2003.
- "ISPELL," http://theochem.ki.ku.dk/on_line_docs/ispell/ispell_1.html, 4 pages, Aug. 23, 2003.
- "Getting Results with Microsoft Office 97" Microsoft, 701 pages, 1995-97.
- Tom Quinlan, "Apple System Software Branches Out," 3 pages, Mar. 16, 1992.
- "Introducing AddressMate Plus, The Address Book Manager for Microsoft Word," 4 pages, May 1996.
- "AddressMate for Windows, New Version 2.0," 9 pages, Oct. 9, 1994. Mar. 1996 Reviews, <http://www.techweb.com/winmag/library/1996/0396/03rvh002.htm> 9 pages, Jan. 26, 2004.
- "ISPELL (1)" 16 pages, Aug. 23, 2003.
- "ISPELL (4)," 11 pages, Aug. 23, 2003.
- "Corel Professional Office 7 Quick Results," 534 pages, no date available.
- "Getting Results with Microsoft Office for Windows 95," Microsoft, 636 pages, 1995.
- User Manual, "For AddressMate and AddressMate Plus," CoStar, 212 pages, 1994-1995.
- "Addressing the Issues," AddressMate Software, 4 pages, Jun. 24, 1993.
- L. R. Shannon, "Peripherals," *The New York Times*, 1 page, Jul. 14, 1992.
- "AddressMate Software Automatically Addresses and Bar Codes Envelopes to Save Time and Money," AddressMate, 3 pages, Mar. 9, 1992.
- "AddressMate for Windows, Adds all the missing pieces to your word processor," 6 pages, no date available.
- "LabelWriter XL Plus," 18 PC Catalog, 1 page, Aug. 18, 1995.
- "AddressMate Plus," CoStar, 16 pages, no date available.
- "AddressMate Incorporates Correction Capability," *Law Technology*, 3 pages, Jan. 1996.
- "User's Guide for Windows 16-Bit, Version 5.2," Novell, Inc., GroupWise, 235 pages, 1993-1997.
- "User's Guide for Windows, 32-Bit" Novell, Inc., GroupWise, 322 pages, 1998.
- "GroupWise Webaccess User's Guide," Novell, Inc., 37 pages, 1998.
- Opposition against European Patent EP 1 171 836, Letter to the European Patent Office in Preparation for the Oral Proceedings scheduled for May 26, 2009, 31 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 1 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 2 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 3 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 4 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 5 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 6 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 7 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 8 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 9 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 10 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 11 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 12 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 13 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 14 Claims, 3 pp., Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 15 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxillary Request 16 Claims, 3 pages, Mar. 26, 2009.
- Expert Report of Professor Benjamin Goldberg, 43 pages, dated Mar. 10, 2008.
- Supplemental Expert Report of Professor Benjamin Goldberg, 15 pages, May 6, 2008.
- Second Supplemental Expert Report of Professor Benjamin Goldberg, 32 pages, Mar. 26, 2009.
- Expert Report of Professor Calvin Gidney III, 3 pages, Mar. 26, 2009.
- European Patent Office, Summons to Attend Oral Proceedings pursuant to Rule 115(1) EPC, 18 pages, Jan. 23, 2009.
- Affidavit of James Miller, 18 pages, Jul. 19, 2006.
- Affidavit of David Block, 23 pages, Jul. 21, 2006.
- Order Re Claim Construction, in *Arendi U.S.A. Inc. et al. v. Microsoft Corp.*, CA No. 02-343-T, from United States District Court for the District of Rhode Island, 4 pages, Sep. 27, 2004.
- Expert Report of W. Bruce Croft, PH.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 66 pages, Mar. 1, 2004.
- Supplemental Expert Report of W. Bruce Croft, PH.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 42 pages, Jun. 9, 2004.
- Microsoft Word 97 Help File entitled "Automatically check spelling and grammar as you type," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Turn automatic changes on or off," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Automatically Correct Text," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Field Codes: Hyperlink Field," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Change the Contents of an AutoCorrect Entry," 1 page, no date available.
- "User Manual For AddressMate and AddressMate Plus," AddressMate Software, 218 pages, 1994-1995.
- Affirming Claim Construction Order, in *Arendi USA et al. v. Microsoft Corp.*, No. 05-1170, 05-1171, from Court of Appeals for the Federal Circuit, 1 page, Feb. 8, 2006.
- Miller, Jim, "Portfolio; Innovative Interfaces for Information Management," Website: <http://www.miramontes.com/portfolio/add/add1.html>, 1 page, no date available.
- Church, Kenneth et al., "Commercial Applications of Natural Language Processing," 30 pages, Mar. 13, 2003.
- Yankelovich, Nichole et al., "Intermedia: The concept and the Construction of a Seamless Information Environment," *Electronic Publishing Technologies*, pp. 81-96, Jan. 1988.
- Utting, Kenneth et al., "Context and Orientation in Hypermedia Networks," *AMC Transactions on Information Systems*, vol. 7, No. 1, pp. 58-84, Jan. 1989.

US 8,306,993 B2

Page 5

- Kahn, Paul, "Linking Together Books: Experiments in Adapting Published Material into Intermedia Documents," *Hypermedia*, vol. 1 No. 2, pp. 1-37, 1989.
- Combs, James, "Hypertext, Full-Text and Automatic Linking," pp. 83-98, 1990.
- Catlin, Karen Smith et al., "Hypermedia Templates: An Authors Tools," *Hypertext 1991 Proceedings*, pp. 147-160, Dec. 1991.
- Meyrowitz, Norman, "The Missing Link, Why We're All Doing Hypertext Wrong," *The Society of Text, Hypertext, Hypermedia, and the Social Construction of Information*, pp. 107-114, no date available.
- Halasz, Frank, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems," *Hypertext 87 Papers*, pp. 345-365, Nov. 1987.
- Haan, Bernard et al., "IRIS Hypermedia Services," pp. 36-51, Jan. 1992.
- McKnight, Dillon, "Review, The Textbook of the Future," *Hypertext: a Psychological Perspective*, pp. 19-51, 8 pages, Mar. 1993.
- Bouvin, Niels Olof, "Augmenting the Web Through Open Hypermedia," 95 pages, Nov. 2000.
- Halasz, Frank et al., "The Dexter Hypertext Reference Model," pp. 95-133, Dec. 7, 1989.
- Weinreich, Harald et al., "The Look of the Link—Concepts for the User Interface of Extended Hyperlinks," 10 pages, no date available.
- Fountain, Andrew et al., "Microcosm: An Open Model for Hypermedia with Dynamic Linking," *Hypertext: Concept Systems and Applications, Proceedings of the First European Conference on Hypertext*, France, 16 pages, Nov. 1990.
- Justus: Wilson, Eve, "Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System," *Library and Linguistics Computing*, vol. 5, No. 2, pp. 119-128, 1990.
- "Claris Em@iler Getting Started," Claris Corporation, 57 pages, 1995.
- "OmniPage Pro for Windows 95," Version 7.0 Caere Corporation, 57 pages, 1996.
- Microsoft Corporation and Dell Inc.'s Invalidity Contentions, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-26, Aug. 14, 2009.
- Exhibit A, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-35, Aug. 14, 2009.
- Exhibit AA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Aug. 14, 2009.
- Exhibit AAA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit B, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit BB, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Aug. 14, 2009.
- Exhibit C, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit CC, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit D, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit DD, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 14, 2009.
- Exhibit E, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit EE, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Aug. 14, 2009.
- Exhibit F, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit FF, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-40, Aug. 14, 2009.
- Exhibit G, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit GG, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit H, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit HH, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit I-1 through I-6, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-1800, Aug. 14, 2009.
- Exhibit II, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit J, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-89, Aug. 14, 2009.
- Exhibit JJ, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-35, Aug. 14, 2009.
- Exhibit K, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-260, Aug. 14, 2009.
- Exhibit KK, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit L, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-190, Aug. 14, 2009.
- Exhibit LL, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit M, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-58, Aug. 14, 2009.
- Exhibit MM, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-29, Aug. 14, 2009.
- Exhibit N, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-47, Aug. 14, 2009.
- Exhibit NN, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-30, Aug. 14, 2009.
- Exhibit O, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-51, Aug. 14, 2009.
- Exhibit OO, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-29, Aug. 14, 2009.
- Exhibit P, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-133, Aug. 14, 2009.
- Exhibit PP, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 14, 2009.
- Exhibit Q *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-40, Aug. 14, 2009.
- Exhibit QQ, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-52, Aug. 14, 2009.

US 8,306,993 B2

Page 6

- Exhibit R, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-44, Aug. 14, 2009.
- Exhibit RR, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit S, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-51, Aug. 14, 2009.
- Exhibit SS, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit T, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit TT, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit U, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit UU, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit V, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit VV, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit W, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-35, Aug. 14, 2009.
- Exhibit WW, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit X, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-7, Aug. 14, 2009.
- Exhibit XX, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-31, Aug. 14, 2009.
- Exhibit Y, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit YY, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-31, Aug. 14, 2009.
- Exhibit Z, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit ZZ, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Defendants Microsoft Corp. And Dell, Inc.'s Brief in Support of their Motion for Summary Judgment of Invalidity for Failure to Comply with 35 U.S.C. § 112, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-23, Aug. 10, 2009.
- Declaration of W. Bruce Croft, Ph.D., *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-15, Aug. 10, 2009.
- Plaintiff's Opening Brief in Support of Its Proposed Claim Constructions, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-36, Aug. 10, 2009.
- Microsoft Corp. and Dell Inc.'s Claim Construction Brief, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-28, Aug. 10, 2009.
- Plaintiff's Counter-Statement of Disputed Material Facts in Opposition to Defendants' Motion for Summary Judgment of Invalidity, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-19, Aug. 24, 2009.
- Plaintiff's Answering Brief in Support of its Proposed Claim Constructions, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-43, Aug. 25, 2009.
- Microsoft Corp. and Dell, Inc.'s Answering Claim Construction Brief, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-25, Aug. 25, 2009.
- Microsoft Corp. and Dell, Inc.'s Response to Plaintiff's Counter-Statement to Defendants' Motion for Summary Judgment of Invalidity, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 31, 2009.
- Microsoft Corp. and Dell, Inc.'s First Amended Answer, Affirmative Defenses and Counterclaims to Complaint, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Oct. 19, 2009.
- Plaintiff's Answer to Defendants' First Amended Counterclaims, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-19, Oct. 30, 2009.
- Microsoft Corp. and Dell, Inc.'s Second Amended Answer, Affirmative Defenses and Counterclaims to Complaint, *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JFF, from United States District Court for the District of Delaware, pp. 1-34, Dec. 8, 2009.
- Decision Revoking European Patent No. EP-B-1171836, European Patent Office, pp. 1-13, Jul. 9, 2009.
- Notice of Appeal for European Patent No. EP-B-1171836, in the European Patent Office, pp. 1-2, Nov. 1, 2009.
- Grounds of Appeal for European Patent No. EP-B-1171836, in the European Patent Office, pp. 1-62, Nov. 19, 2009.
- Submission of opponent (Microsoft) in opposition against European Patent No. 1 171 836, Appeal T1779/09-3501, in the European Patent Office, 32 pages, Mar. 25, 2010.
- Second Affidavit of David Block, 15 pages, May 15, 2009.
- Second Affidavit of James Miller, 36 pages, May 15, 2009.
- Hall, Wendy, et al., "Rethinking Hypermedia," Kluwer Academic Publishers, 87 pages, 1996.
- Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 196 pages, Aug. 13, 2004.
- Exhibits for the Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 334 pages, Aug. 13, 2004.
- Writ of Summons to Arendi Holding Co., District Court in the Hague and English translation, 65 pages, Oct. 26, 2005.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, 55 pages, date unknown.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 51 pages, date unknown.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, 26 pages, Sep. 24, 2008.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, English Translation, 26 pages, Sep. 24, 2008.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising a Counterclaim, With Exhibit, District Court in the Hague, No. 2006/0782, 57 pages, Jan. 31, 2007.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising a Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 23 pages, Jan. 31, 2007.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the Counterclaim With Exhibits, District Court in the Hague, No. 2006/0782, 123 pages, Mar. 12, 2008.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the

US 8,306,993 B2

Page 7

- Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 33 pages, Mar. 12, 2008.
- Letter from Abraham H. Spierer to David Block, Dec. 19, 1994.
- Claris Em@iler User's Manual, 187 pages, 1995-1997.
- "Word Lookup Data Detectors," Dictionary.com, available at <http://dictionary.reference.com/mac/add.html>, 3 pages, Jan. 12, 2004.
- Apple.com internet web pages (marked "MS 118139-MS 118143"), 5 pages, Dec. 7, 2001.
- "Overview: What are Apple Data Detectors", Apple.com internet web pages (marked "MA 118078-MS 118092"), 15 pages, Jun. 1, 2002.
- Addressmate Plus User Manual Addendum, 2 pages, Nov. 1, 1995.
- Now Software, Inc., "Now Contact & Up*to*Date Quick Reference Guide", 18 pages, 1992-95.
- Now Software, Inc., "Now Contact version 3.5 User's Guide", 229 pages, 1992-95.
- Now Software, Inc., "Now Up*to*Date version 3.5 User's Guide", 232 pages, 1992-95.
- Now Software, Inc., "Now Utilities Quick Reference Guide", (6 pages), 1992-1995.
- Now Software, Inc., "What's New in Now Up*to*Date & Now Contact", 14 pages, 1995.
- Microsoft's Supplemental Response to Arendi's Interrogatory No. 5, United States District Court for the District of Rhode Island, 4 pages, Apr. 7, 2003.
- Microsoft's Supplemental Response to Arendi's Second and Thrid Sets of Interrogatories (Nos. 10,12), United States District Court for the District of Rhode Island, 22 pages, Sep. 8, 2003.
- Reply Brief of Plaintiffs-Appellants Arendi Usa, Inc. and Arendi Holding Limited, United States Court of Appeals for the Federal Circuit, 37 pages, Sep. 2, 2005.
- Apple Data Detectors/Internet Address Detectors, ADD-IAD_1.0.2_Info.txt, 1 page, Mar. 5, 1998.
- Cortinas, M., Data Ratchet extracts info, ZD.com, 2 pages, Apr. 3, 1997.
- Glucose Unveils Data Ratchet v.1.1!, Glucose Development Corp., 4 pages, 1997.
- Glucose Releases Data Ratchet Source, Glucose Press Release, 1 page, Dec. 18, 1999.
- Are you a document mechanic?, from Glucose Development Corp.'s webpage (<http://glulabs.com>), 2 pages, 2002.
- Glucose Open Source Tools, from Glucose Development Corp., 2 pages, 1999.
- E-Mail 'Net Differences—Eudora Pro and E-Mail Connection shine in their own ways . . . , Communications Week, 4 pages, Jan. 6, 1997.
- Internet Update Sep. 10, 1997, Newsbytes news Network, 5 pages, Sep. 10, 1997.
- Brouwer, r., ADD → Address Book, 1 page, Jun. 8, 1998.
- Byrne, M., ADD → New Mail, 1 page, Aug. 23, 1998.
- MacCentral: Apple Macintosh News, 4 pages, Jan. 12-13, 2004.
- Plaintiff's Answer to Microsoft Corporation's and Dell Inc.'s Second Amended Counterclaims, in *Arendi Holding Ltd. v. Microsoft Corp. and Dell Inc.*, CA No. 09-119-JJF-LPS, from United States: District Court for the District of Delaware, pp. 1-20, Dec. 14, 2009.
- Johannes, S., et al., "Active Messenger: Email Filtering and Mobile Delivery," *Massachusetts Institute of Technology*, Sep. 1999, pp. 1-122.
- Srihari, S, et al., "A System to Read Names and Addresses on Tax Forms," *Proceedings of the IEEE*, vol. 14, No. 7, Jul. 1996, pp. 1038-1049.
- U.S. Appl. No. 12/841,302, Interview Summary, 3 pages, Dec. 20, 2010.
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626 As Filed, 58 pages, Nov. 10, 1998.
- US Patent No. 6,323,853, Preliminary Amendment, U.S. Appl. No. 09/189,626, 20 pages, Apr. 29, 1999.
- US Patent No. 6,323,853, Office Action, U.S. Appl. No. 09/189,626, 9 pages, Apr. 26, 2000.
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626, Interview Summary, 2 pages, Jun. 14, 2000.
- US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Jul. 25, 2000.
- US Patent No. 6,323,853, Final Office Action, U.S. Appl. No. 09/189,626, 10 pages, Sep. 18, 2000.
- US Patent No. 6,323,853, Interview Summary, U.S. Appl. No. 09/189,626, 1 page, Oct. 17, 2000.
- US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Dec. 18, 2000.
- US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 7 pages, Jan. 4, 2001.
- US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 2 pages, Oct. 2, 2001.
- US Patent No. 7,496,854, Application As Filed, 100 pages, Aug. 6, 2001.
- US Patent No. 7,496,854, Preliminary Amendment, U.S. Appl. No. 09/923,134, 7 pages, Dec. 17, 2001.
- US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 10 pages, Oct. 15, 2002.
- US Patent No. 7,496,854, Response, 4 pages, U.S. Appl. No. 09/923,134, Apr. 15, 2003.
- US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 7 pages, Jun. 27, 2003.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 6 pages, Aug. 21, 2003.
- US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 4 pages, Nov. 19, 2003.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 5 pages, Feb. 10, 2004.
- US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 4 pages, May 19, 2004.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 14 pages, Dec. 30, 2004.
- US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 7 pages, Aug. 29, 2005.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 17 pages, Nov. 16, 2005.
- US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 10 pages, Feb. 6, 2006.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 22 pages, Aug. 4, 2006.
- US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 8 pages, Oct. 18, 2006.
- US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 3 pages, Jan. 23, 2007.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Apr. 18, 2007.
- US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 8 pages, Jul. 13, 2007.
- US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 4 pages, Oct. 19, 2007.
- US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 9 pages, Nov. 9, 2007.
- US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Jan. 24, 2008.
- US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 7 pages, Jul. 2, 2008.
- US Patent No. 7,496,854, Amendment Under Rule 312; U.S. Appl. No. 09/923,134, 27 pages, Jul. 8, 2008.
- US Patent No. 7,496,854, Response to Amendment Under Rule 312, U.S. Appl. No. 09/923,134, 4 pages, Jan. 21, 2009.
- U.S. Appl. No. 12/182,048, Application as Filed, 47 pages, Jul. 29, 2008.
- U.S. Appl. No. 12/182,048, Office Action, 34 pages, Oct. 28, 2010.
- U.S. Appl. No. 12/182,048, Interview Summary, 4 pages, Dec. 1, 2010.
- U.S. Appl. No. 12/182,048, Response, 36 pages, Dec. 8, 2010.
- U.S. Appl. No. 12/841,302, Application As Filed, 47 pages, Jul. 22, 2010.
- U.S. Appl. No. 12/841,302, Accelerated Examination Support Document, 150 pages, Jul. 22, 2010.
- U.S. Appl. No. 12/841,302, Office Action, 22 pages, Nov. 24, 2010.
- U.S. Appl. No. 12/841,302, Response, 18 pages, Dec. 14, 2010.
- US Patent No. 7,272,604, U.S. Appl. No. 09/390,303 As Filed, 55 pages, Sep. 3, 1999.

US 8,306,993 B2

Page 8

- US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 9 pages, May 8, 2002.
- US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 7 pages, Aug. 8, 2002.
- US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 11 pages, Sep. 27, 2002.
- US Patent No. 7,272,604, Proposed Response, U.S. Appl. No. 09/390,303, 4 pages, Oct. 24, 2002.
- US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 3 pages, Mar. 27, 2003.
- US Patent No. 7,272,604, Final Office Action, U.S. Appl. No. 09/390,303, 9 pages, Apr. 21, 2003.
- US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 16 pages, Oct. 21, 2003.
- US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Nov. 17, 2003.
- US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 6 pages, Apr. 6, 2004.
- US Patent No. 7,272,604, Supplemental Notice of Allowance, U.S. Appl. No. 09/390,303, 2 pages, Mar. 7, 2006.
- US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 10 pages, Jun. 6, 2006.
- US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 19 pages, Dec. 5, 2006.
- US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Feb. 7, 2007.
- U.S. Appl. No. 12/963,744, Application As Filed, 53 pages, Dec. 9, 2010.
- U.S. Appl. No. 12/963,744, Accelerated Examination Support Document, 80 pages, Dec. 9, 2010.
- Person et al., "Special Edition Using Microsoft Word 97" Que Publishing, pp. 475-514, Dec. 16, 1996.
- U.S. Appl. No. 12/182,048, Notice of Allowance, 53 pages, Jan. 19, 2011.
- U.S. Appl. No. 12/841,302, Notice of Allowance, 31 pages, Jan. 13, 2011.
- Deposition of Atle Hedloy in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 162-213, Mar. 7, 2010.
- Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-67, Apr. 9, 2010.
- Exhibit 15, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-26, Apr. 9, 2010.
- Exhibit 16, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.
- Exhibit 17, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 18, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Apr. 9, 2010.
- Exhibit 19, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-13, Apr. 9, 2010.
- Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-115, Apr. 9, 2010.
- Exhibit 11, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-7, Apr. 9, 2010.
- Exhibit 12, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.
- Exhibit 13, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 14, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-18, Apr. 9, 2010.
- Exhibit 15, Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Apr. 9, 2010.
- Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-166, Apr. 30, 2010.
- Exhibit a, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-30, Apr. 30, 2010.
- Exhibit B, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-15, Apr. 30, 2010.
- Exhibit C, Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-90, Apr. 30, 2010.
- Michael Bieber "On Automating Hypertext Links in Knowledge-Based Decision Support Systems", Technical Report BCCS-90-08, pp. 1-22, Apr. 1990.
- N. Streitz, A. Rizk and J. André, Hypertext: Concepts, Systems and Applications: Proceedings of the First European Conference on Hypertext, Inria, France, pp. 1-8, Nov. 1990.
- Michael Bieber, "Issues in Modeling A "Dynamic" Hypertext Interface", Technical Report BCCS-91-2, pp. 1-13, Apr. 1991.
- Michael Bieber and Tomás Isakowitz, "Valuation Links: Extending the Computational Power of Hypertext", Technical Report BCCS-91-1, pp. 1-11, Apr. 1991.
- Michael Bieber, Template-Driven Hypertext: A Methodology for Integrating a Hypertext Interface into Information Systems, Technical Report BCCS-91-3, pp. 1-23, Jun. 1991.
- Michael Bieber and Steven O. Kimbrough, "On Generalizing the Concept of Hypertext", Technical report BCCS-91-5, pp. 1-11, Jun. 1991.
- Michael Bieber and Tomás Isakowitz, "Bridge Laws in Hypertext A Logic Modeling Approach", Technical Report BCCS-91-4, pp. 1-30, Jun. 1991.
- Michael Bieber, "On Merging Hypertext Into Dynamic, Non-Hypertext Systems", Technical Report BCCS-91-14, pp. 1-24, Nov. 8, 1991.
- Mark Bernstein, Jay David Bolter, Michael Joyce and Elli Mylonas, "Architectures for Volatile Hypertext", Hypertext '91 Proceedings, pp. 243-280, Dec. 1991.
- Michael Bieber, "Providing Information Systems With Full Hypermedia Functionality", pp. 1-15, Oct. 1992.
- Mail Address Change Notification, IBM Technical Disclosure Bulletin, vol. 36 No. 1, pp. 70-72, Jan. 1993.
- D.D. Cowan, C.J.P. Lucena and R.G. Veitch, Towards CAAI: Computer Assisted Application Integration, Technical Report Series CS-93-17, University of Waterloo, pp. 1-23, Oct. 1993.
- Herman Kaind and Stefan Kramer, Semiautomatic Generation of Dictionary Links in Hypertext, pp. 1-14, Feb. 1, 1995.
- Michael Bieber, "On Integrating Hypermedia into Decision Support and Other Information Systems", Decision Support Systems 14, pp. 251-267, 1995.
- Proceedings, Sixth Message Understanding Conference (MUC-6), pp. 1-64, Nov. 6-8, 1995.
- Sargur N. Srihari, et al, "A System to Read Names and Addresses on Tax Forms", Proceedings of the IEEE, vol. 84, No's, Jul. 1996.
- Ph. Bonnet and S. Bressan, "Extraction and Integration of Data from Semi-structured Documents into Business Applications", Sloan WP#3979, CISL WP# 97-12, pp. 1-9, Sep. 1997.
- Borland, Russell, "Advance Self-Study Step by Step Microsoft Word 97 Advance Topics", Microsoft Press, 331 pages, 1997.
- Microsoft Word Language Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1133 pages, 1997.
- Microsoft PowerPoint and Microsoft Outlook Language References, Official Technical Reference to Visual Basic for Applications, Microsoft Press, vol. 4, 539 pages, 1997.
- Microsoft Visual Basic for Applications and Shared Libraries References, vol. 5 of 5, Microsoft Press, 1005 pages, part No. 097-001685, 1997.
- Microsoft Office 97 Resource Kit, Technical Information and Tools for the Support Professional, Microsoft Press, 1162 pages, 1997.

US 8,306,993 B2

Page 9

- Microsoft Office97 Reviewer's Guide, Professional Edition, 546 pages, Aug. 1996.
- Russell Borland, Running Microsoft Word for Windows 95, In Depth Reference and Inside Tips from the Software Experts, Russell Borland, Microsoft Press, 1040 pages, 1995.
- Michael Halvorson and Michael Young, Running Microsoft Office 97, Standard and Professional, Michael Halvorson and Michael Young, Microsoft Press, 1130 pages, 1997.
- Microsoft Word 97, self-Study Kit, Covers Microsoft Word 97 for Windows 95 and Windows NT version 4.0, Microsoft Press, 314 pages, 1997.
- Microsoft Outlook 97, Step by Step, Self-Study Kit, Microsoft Press, 359 pages, 1997.
- Ken Getz and Mike Gilbert, VBA Developer's Handbook, Reusable Code, Visio Professional Demo Copy, and VB5CCE, 1997.
- Microsoft97/Visual Basic Programmer's Guide, Microsoft Press, 548 pages, 1997.
- Microsoft Access Language Reference and Office Data Access Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1287 pages, 1997.
- Davis Boctor, Microsoft Office97 Step by Step, Visual Basic, Developer's Self-Study Guide, Microsoft Press, 359 pages, 1997.
- Christine Solomon, Microsoft Office97 Developer's Handbook, Microsoft Press, 596 pages, 1997.
- Microsoft Word Developer's Kit, Microsoft Professional Editions, Third Edition, Manufacturing Copy 1st, Microsoft Press, 1106 pages, 1995.
- Inside Microsoft Word, vol. 4, No. 1, The Cobb Group, pp. 1-16, Jan. 1997.
- Inside Microsoft Word, vol. 4, No. 2, The Cobb Group, pp. 1-16, Feb. 1997.
- Inside Microsoft Word, vol. 4, No. 3, The Cobb Group, pp. 1-16, Mar. 1997.
- Inside Microsoft Word, vol. 4, No. 4, The Cobb Group, pp. 1-16, Apr. 1997.
- Inside Microsoft Word, vol. 4, No. 7, The Cobb Group, pp. 1-16, Jul. 1997.
- Inside Microsoft Word, vol. 5, No. 9, ZD Journals, pp. 1-16, Sep. 1998.
- Inside Microsoft Word, vol. 5, No. 10, ZD Journals, pp. 1-16, Oct. 1998.
- Inside Microsoft Word, vol. 5, No. 11, ZD Journals, pp. 1-16, Nov. 1998.
- Eudora Mail Pro, Version 3.0 for Windows, User Manual, QUALCOMM Incorporated, 198 pages, Jun. 1997.
- Eudora Mail Pro, Version 3.1 for Macintosh, User Manual, QUALCOMM Incorporated, 199 pages, Jun. 1997.
- Non-Confidential Redacted Jun. 11, 2010 Rebuttal Expert Report Richard Taylor, 311 pages, Jun. 11, 2010.
- Spell Checking for Microsoft Windows, Palantir Software, 36 pages, © 1986.
- Opposition for European Patent No. EP-B-1171836, in the European Patent Office, pp. 1-2, Jul. 26, 2006.
- In Response to Notice of Opposition for European Patent No. EP-B-1171836, in the European Patent Office, pp. 1-21, Mar. 7, 2007.
- Brangan, James, Department 1824 Job card System: A New Web-based Business Tool, Sandia National Laboratories, 50 pages, Feb. 1998.
- Fox, Heidi, et al., Learning to Extract and Classify Names from Text, IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, 6 pages, Oct. 11-14, 1998.
- Frank, Martin, et al., Adaptive Forms: An Interaction Technique for Entering Structured Data, University of Southern California, Information Sciences Institute, 11 pages, Sep. 30, 1998.
- Novasoft Systems, Infomax Messaging System v.2.0 User's Manual, 46 pages, Jul. 1, 1996.
- U.S. Appl. No. 13/041,210, Application As Filed, 52 pages, Mar. 4, 2011.
- U.S. Appl. No. 12/963,744, Office Action, 14 pages, Mar. 10, 2011.
- U.S. Appl. No. 12/987,840, Application As Filed, 52 pages, Jan. 10, 2011.
- U.S. Appl. No. 12/987,840, Accelerated Examination Support Document, 74 pages, Jan. 10, 2011.
- U.S. Appl. No. 12/987,939, Application As Filed, 52 pages, Jan. 10, 2011.
- U.S. Appl. No. 12/987,939, Accelerated Examination Support Document, 72 pages, Jan. 10, 2011.
- Plaintiff's Supplemental Opening Brief in Support of Its Proposed Claim Constructions, in *Arendi Holding Ltd. v. Microsoft Corp*, CA No. 09-119-LPS, from United States District Court for the District of Delaware, pp. 1-26, Jan. 21, 2011.
- Microsoft Corporation's Supplemental Claim Construction Brief for Additional Terms to Be Construed, in *Arendi Holding Ltd. v. Microsoft Corp*, CA No. 09-119-LPS, from United States District Court for the District of Delaware, pp. 1-25, Jan. 21, 2011.
- Plaintiff's Supplemental Answering Brief in Support of Its Proposed Claim Constructions, in *Arendi Holding Ltd. v. Microsoft Corp*, CA No. 09-119-LPS, from United States District Court for the District of Delaware, pp. 1-26, Feb. 4, 2011.
- Microsoft Corporation's Responsive Supplemental Claim Construction Brief for Additional Terms to Be Construed, in *Arendi Holding Ltd. v. Microsoft Corp*, CA No. 09-119-LPS, from United States District Court for the District of Delaware, pp. 1-18, Feb. 4, 2011.
- U.S. Appl. No. 12/963,744, Response, 20 pages, Apr. 11, 2011.
- U.S. Appl. No. 13/111,639, Application as Filed; 48 pages, May 19, 2011.
- U.S. Appl. No. 13/111,639, Accelerated Examination Support Document, 160 pages, May 19, 2011.
- U. S. Appl. No. 12/963,744, Office Action, 33 pages, issued Oct. 11, 2011.
- U. S. Appl. No. 12/963,744, Response C, 18 pages, filed Nov. 11, 2011.
- U. S. Appl. No. 12/963,744, Supplement to Response C, 12 pages, filed Nov. 30, 2011.
- U. S. Appl. No. 12/963,744, Office Action, 13 pages, issued Dec. 6, 2011.
- U. S. Appl. No. 12/963,744, Request for Continued Examination and Response D, 16 pages, filed Jan. 6, 2012.
- U. S. Appl. No. 12/963,744 Supplemental to Response D, 15 pages, filed Mar. 19, 2012.
- U. S. Appl. No. 12/963,744 Office Action, 10 pages, issued Mar. 20, 2012.
- U. S. Appl. No. 12/963,744 Response E, 11 pages, filed Mar. 26, 2012.
- U. S. Appl. No. 12/987,840, Office Action, 49 pages, issued Oct. 11, 2011.
- U. S. Appl. No. 12/987,840, Appeal Brief, 36 pages, filed Jan. 11, 2012.
- U. S. Appl. No. 12/987,840, Response to Notice of Non-Compliant Appeal Brief, 7 pages, filed Feb. 22, 2012.
- U. S. Appl. No. 12/987,840, Examiner's Answer, 19 pages, issued Mar. 9, 2012.
- U. S. Appl. No. 12/987,939, Office Action, 48 pages, issued Oct. 11, 2011.
- U. S. Appl. No. 12/987,939, Appeal Brief, 33 pages, filed Jan. 11, 2012.
- U. S. Appl. No. 12/987,939, Examiner's Answer, 18 pages, issued Feb. 27, 2012.
- U.S. Appl. No. 13/041,210, Office Action, 64 pages, issued Jan. 9, 2012.
- U.S. Appl. No. 13/111,639, Office Action, 60 pages, issued Sep. 1, 2011.
- U.S. Appl. No. 13/111,639, Response A, 24 pages, filed Sep. 21, 2011.
- U.S. Appl. No. 13/111,639, Office Action, 36 pages, issued Nov. 10, 2011.
- U.S. Appl. No. 13/111,639, Office Action, 3 pages, issued Nov. 28, 2011.
- U.S. Appl. No. 13/111,639, Response B, 19 pages, filed Dec. 9, 2011.
- U.S. Appl. No. 13/111,639, Advisory Action, 3 pages, issued Dec. 22, 2011.
- User Manual for AddressMate and AddressMate Plus 1994-1995 by AddressMate Software.
- Abowd, Gregory D., et al., "Applying Dynamic Integration as a Software infrastructure for Context-Aware Computing," GVU Technical Report, GIT-GVU-97-18 (Sep. 1997) (MS 019683-MS 019692).

US 8,306,993 B2

Page 10

- Abowd, Gregory D., et al, "Context-Awareness in Wearable and Ubiquitous Computing," Gvu Technical Report, GIT-GVU-97-11 (Mar. 1997) (MS 096928-MS 096936).
- Apple Data Detectors User's Manual (Jul. 1, 1997) (MS 019711-MS 019726).
- Apple Internet Address Data Detectors User's Manual (Aug. 28, 1997) (MS 110172-MS 110187).
- Apple Introduces Internet Address Detectors (Sep. 8, 1997) (MS 049582-MS 049589).
- Wood, Andrew, et al., CyberDesk: Automated Integration of Desktop and Network Services, Gvu Technical Report, OIT-GVU-97-11, May 1997.
- Gregory D. Abowd, Anion Dey and Andy M. Wood, Applying Dynamic Integration as a Software Infrastructure for Context-Aware Computing, Gvu Technical Report, GIT-GVU-97-18, Sep. 1997.
- Gregory D. Abowd, Anind Dey, Robert Orr and Jason Brotherson, Context-awareness in wearable and ubiquitous computing, Gvu Technical Report, GIT-GVU-97-11, Mar. 1997.
- Apple Data Detectors User's Manual, Jul. 1, 1997.
- Apple Internet Address Detector User's Manual, Aug. 28, 1997.
- Apple Introduces Internet Address Detectors, Sep. 8, 1997.
- Anind K. Dey and Gregory D. Abowd, CyberDesk: The Use of Perception in Context-Aware Computing, PUI Workshop Submission, Proc. of 1997 Workshop on Perceptual User Interfaces (PUI '97), pp. 26-27, Oct. 1997.
- Anind K. Dey, Context-Aware Computing: The CyberDesk Project, Future Computing Environments, AAAI '98 Spring Symposium, Stanford University, pp. 51-55, Mar. 23-25, 1998.
- Anind K. Dey, Gregory D. Abowd and Andrew Wood, CyberDesk: A framework for providing self-integrating context-aware services, Knowledge-Based Systems, vol. 11, No. 1, pp. 3-13, Sep. 1998.
- Anind K. Dey, Gregory D. Abowd, Mike Pinkerton and Andrew Wood, CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, Gvu Technical Report, GIT-GVU-97-10, May 1997.
- L. Nancy Garrett, Karen E. Smith and Norman Meyrowitz, Intermedia: Issues, Strategies, and Tactics in the Design of a Hypermedia Document System, (c) 1986, pp. 163-174.
- Eve Wilson, Links and Structures in hypertext databases for law, Proceedings of the First European Conference on Hypertext, INRIA, Nov. 1990.
- Mike Langberg, 'Innovation is at the heart of what we do', Apple breaks new ground by displaying what's on its drawing board, Mercury News, Aug. 7, 1996, pp. 1-2.
- Henry Lieberman, Bonnie A. Nardi and David Wright, Training Agents to Recognize Text by Example, Proc. of the Third Annual Conference on Autonomous Agents, Seattle, WA, pp. 116-122, 1999.
- Bonnie A. Nardi, James R. Miller and David J. Wright, Collaborative, Programmable Intelligent Agents, Website, Mar. 1998, pp. 1-11.
- Milind S. Pandit and Sameer Kalbag, The Selection Recognition Agent: Instance Access to Relevant Information and Operations, Proc. of Intelligent User Interfaces 1997, Orlando, FL, 1997.
- Addressmate Automatic Envelope Addressing Program, User's Manual, 1991.
- Peter Brown, Unix Guide, 1995.
- Lee E. McMahon, SED—A Non-interactive Text Editor, Bell Laboratories, Aug. 15, 1978.
- AddressMate Incorporates Correction Capability, The New York Law Publishing Company, Jan. 1996.
- Lawrence J. Magid, Addressing the Matter of Labels, Los Angeles Times, Sep. 23, 1996.
- Mar. 1996 Reviews, website available at www.techweb.com/winmag/library/1996/0396/03rvh002.htm.
- Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. v. Microsoft Corporation et al.*, Sep. 23, 2004.
- Trial Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. v. Microsoft Corporation et al.*, Sep. 27, 2004.
- Excerpt from Jury Charge Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. v. Microsoft Corporation et al.*, Sep. 28, 2004.
- United States Court of Appeals for the Federal Circuit, Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Judge Ernest C. Torres. Brief for Defendant-Cross Appellant Frank E. Scherkenbach Jul. 7, 2005.
- United States District Court for the District of Rhode Island, 02-CV-343 (ECT) Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law That the '853 Patent Is Invalid, Oct. 15, 2004.
- United States District Court District of Rhode Island, CIV. A. No. 02-CV-343 (ECT), Plaintiffs' Reply Memorandum in Support of Their Motion for New Trial. Francis A. Connor. Nov. 4, 2004.
- United States Court of Appeals Federal Circuit, Brief for Plaintiff-Appellant, Frank E. Scherkenbach, Sep. 2, 2005.
- United States District Court for the District of Rhode Island 02-CV-343 (ECT). Memorandum in Support of Defendant Microsoft Corporation's Opposition to Arendi's Motion for a New Trial. Patricia A. Sullivan, Oct. 27, 2004.
- United States Court of Appeals for the Federal Circuit, Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Brief of Plaintiffs-Appellants Arendi U.S.A., Inc. And Arend Holding Limited. Donald R. Dunner, Apr. 25, 2005.
- United States Court of Appeals for the Federal Circuit, Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Reply Brief for Defendant-Cross Appellant. Frank E. Scherkenbach, Oct. 3, 2005.
- United States District Court for the District of Rhode Island, Order Denying Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law that the '853 Patent Is Invalid, C.A. No. 02-343T. Ernest C. Torres, Chief Judge, Nov. 30, 2004.
- United States District Court for the District of Rhode Island. C.A. No. 02-343T. Order Denying Plaintiffs' Motion for New Trial. Ernest C. Torres, Chief Judge, Nov. 30, 2004.
- Press Release: Apple Introduces Internet Address Detectors, Sep. 8, 1997.
- MacWEEK Report, Aug. 8, 1996.
- Mike Langber, Show of Potential Apple Breaks New Ground by Displaying What's on Its Drawing Board 'Innovation is at the heart of what we do', in San Jose Mercury News, Aug. 7, 1996, p. 1C.
- Apple Introduces Internet Address Detectors, Newsbytes, Sep. 29, 1997.
- Greg Williams, Strategy Mosaic: Understanding Apple's Dual OS Strategy.
- Taking [control] of your Mac with System 8, The MacAuthority, Jan. 1998.
- Apple Data Detectors 1.0.2, TidBITS Updates, Mar. 8, 1998.

* cited by examiner

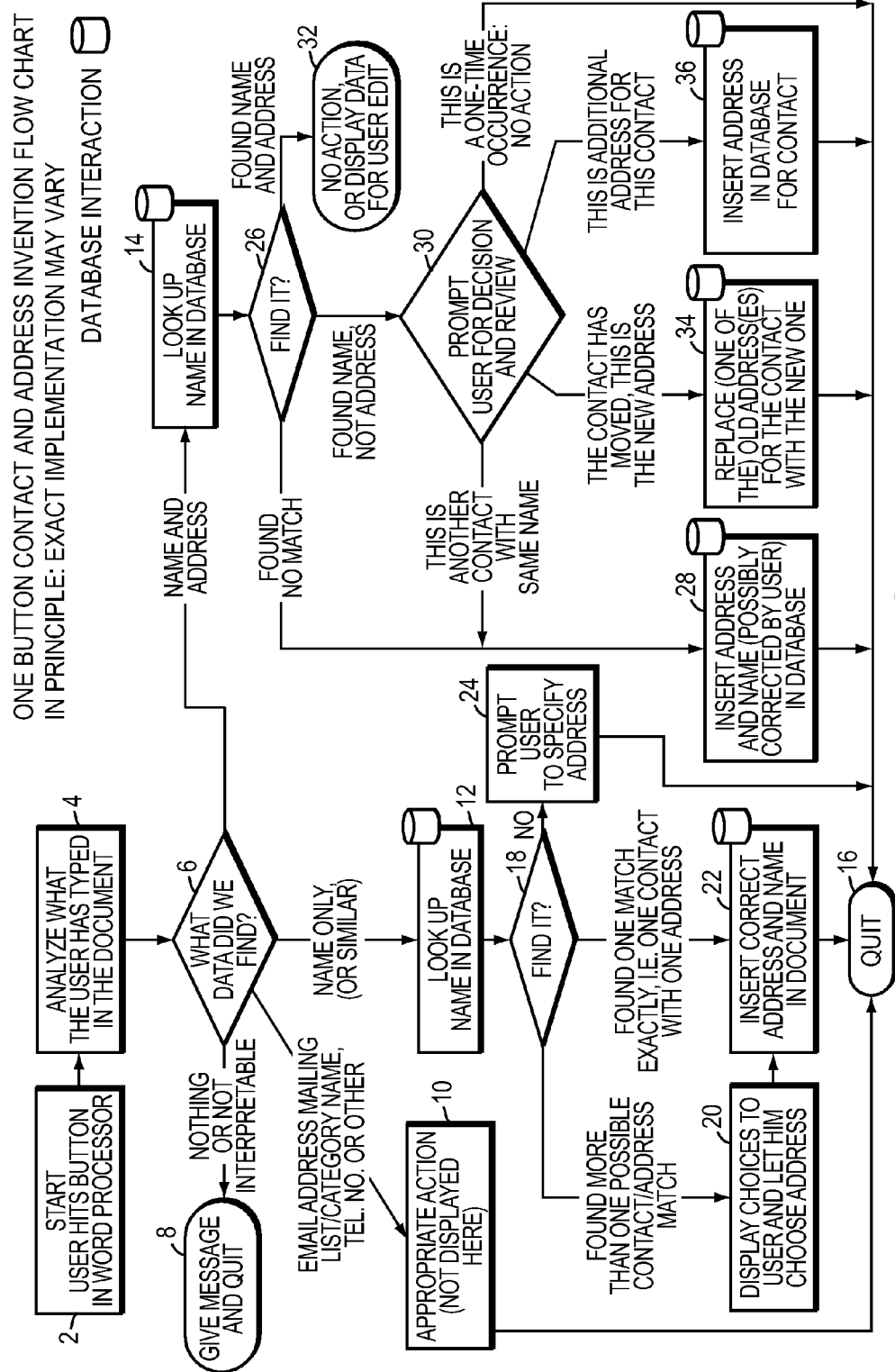


FIG. 1

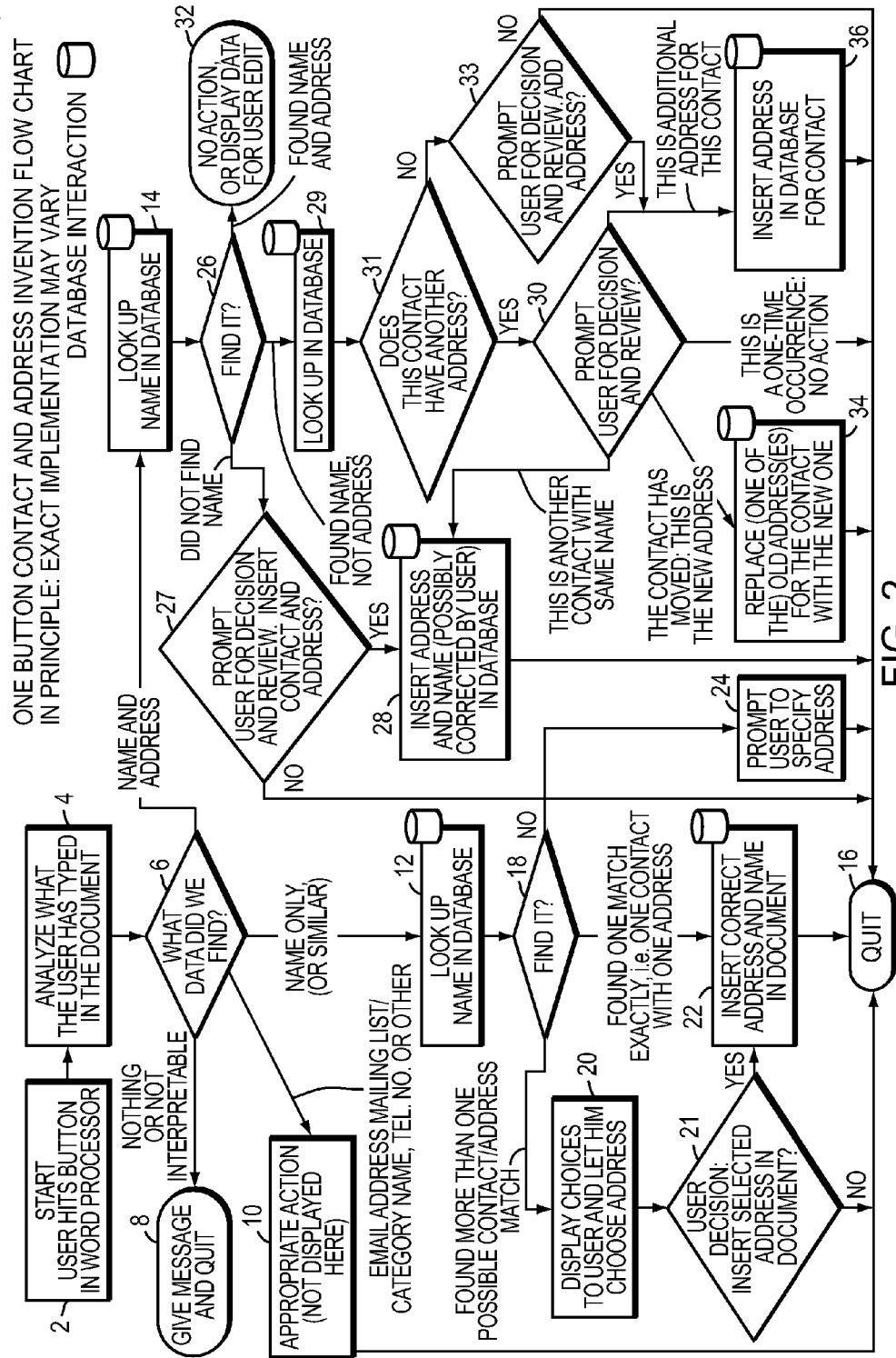


FIG. 2

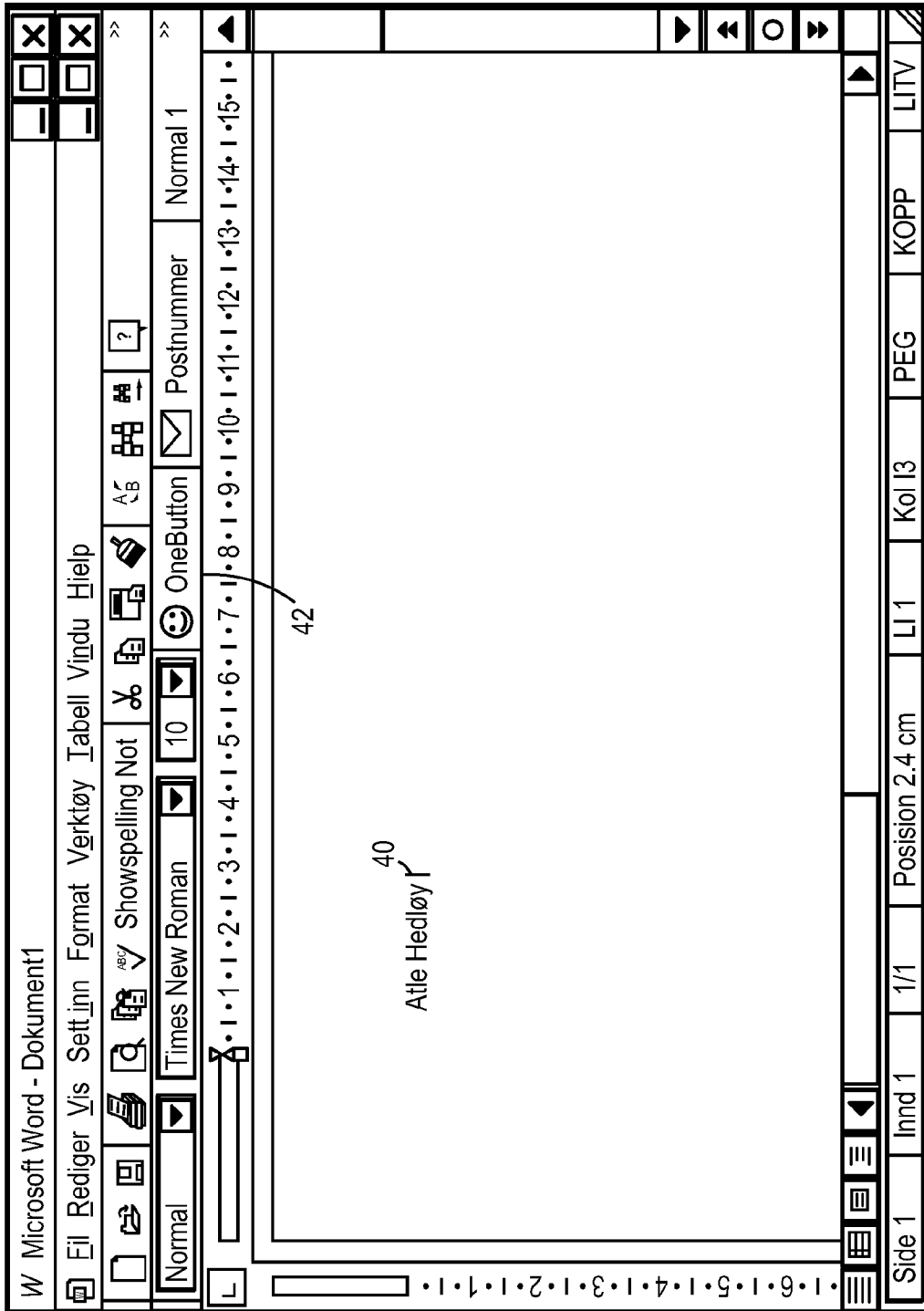


FIG. 3

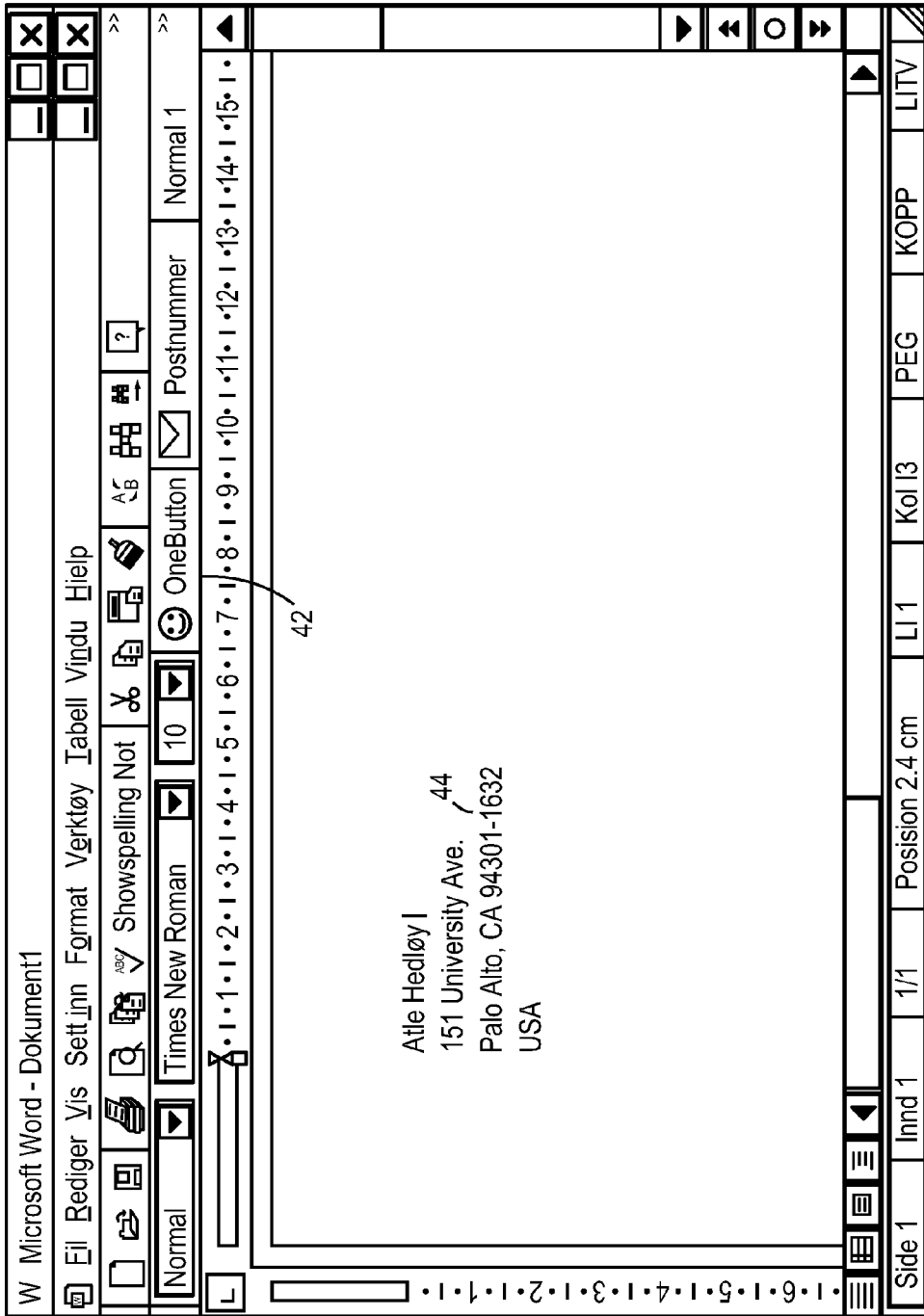


FIG. 4

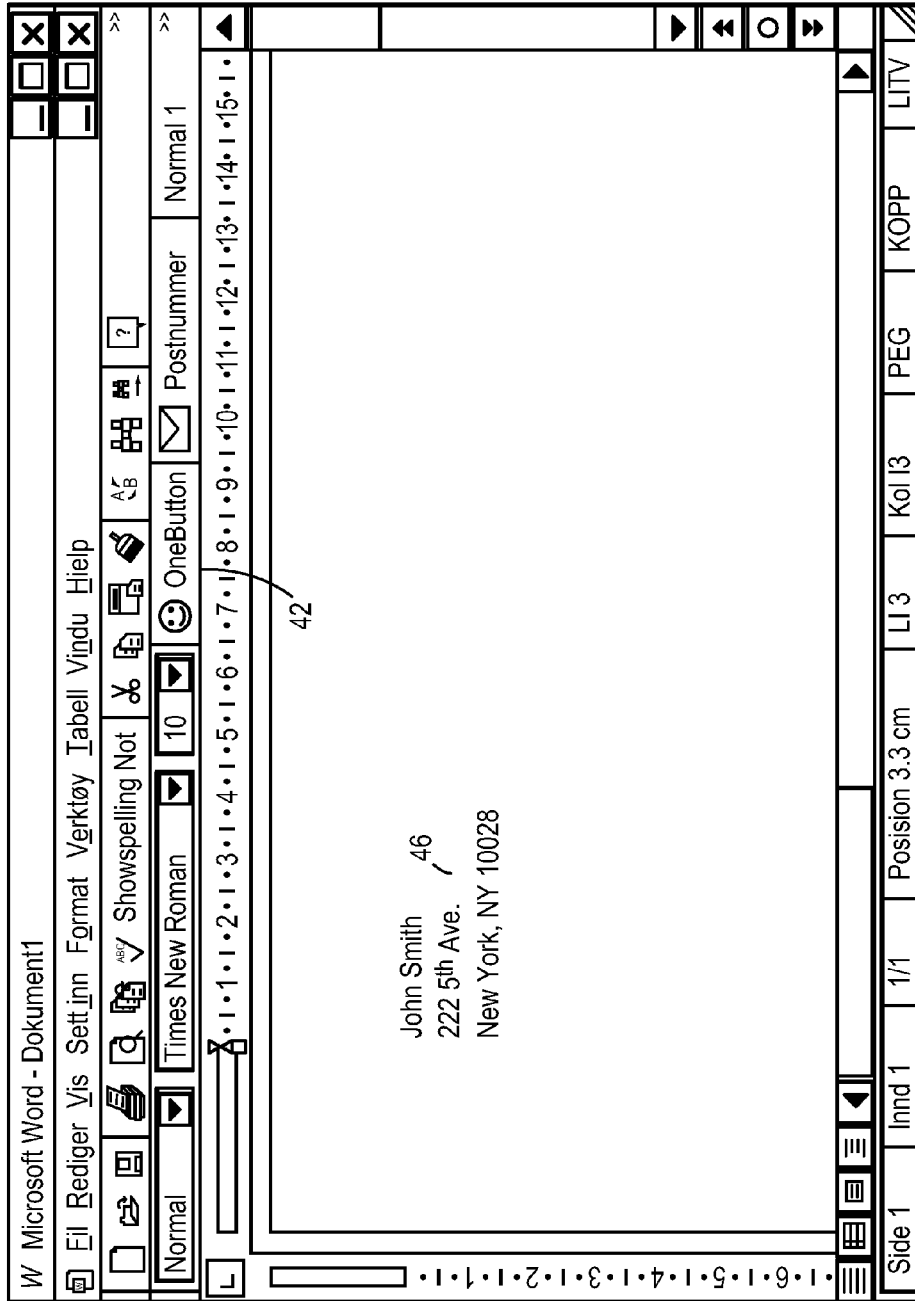


FIG. 5

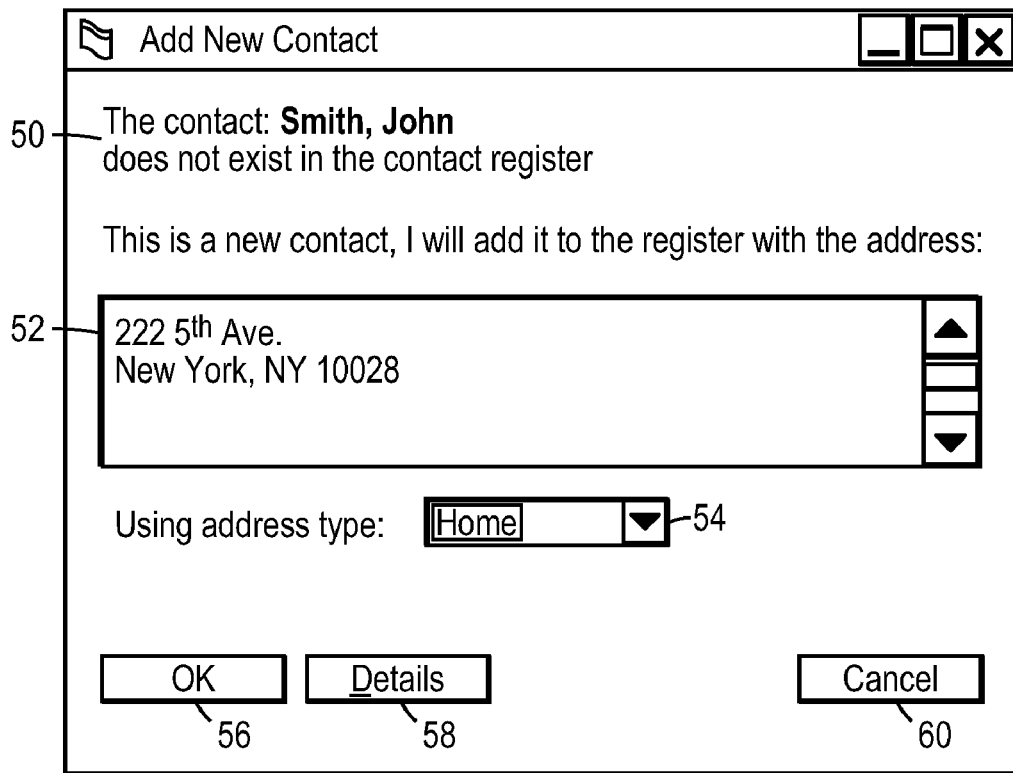


FIG. 6

The screenshot shows a software window titled "Arendi OneButton Contact Register". The window contains a contact form with the following fields and controls:

- Name:** A text field containing "John Smith" with a dropdown arrow on the right.
- Title:** A dropdown menu.
- First:** A text field containing "John".
- Middle:** A text field containing "Smith".
- Last:** A text field containing "Smith".
- Suffix:** A dropdown menu.
- Company:** A text field.
- Address type:** A dropdown menu with "Home" selected.
- Street:** A text field containing "222 5th Ave.".
- City:** A text field containing "New York".
- State/Province:** A text field containing "NY".
- ZIP/Postal:** A text field containing "10028".
- Country:** A dropdown menu.

At the bottom of the form area, there is a button labeled "Add and Choose" (64) and a button labeled "Options..." (66). At the bottom right of the window, there is a "Cancel" button (60) and a status bar containing the text "Delete er en test".

FIG. 7

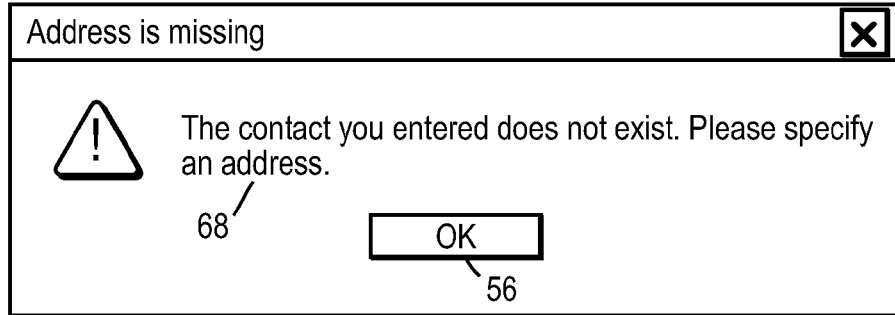


FIG. 8

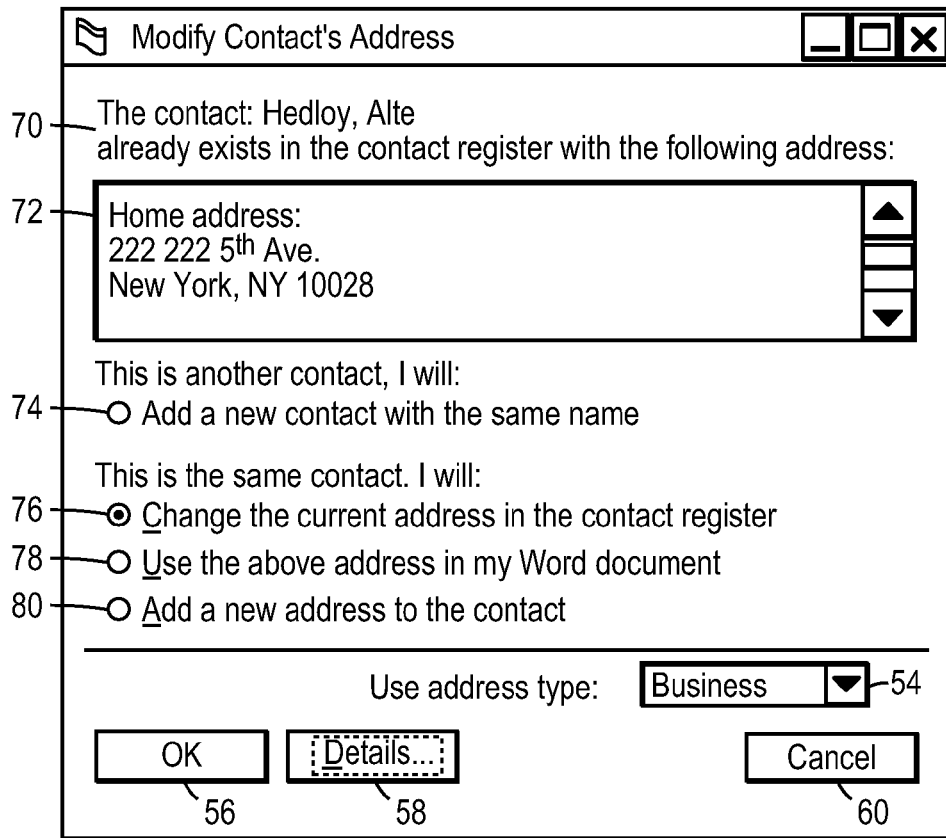


FIG. 9

Arendi OneButton Contact Register

Existing addresses with the same name

Name	Address type	Address	Zip	City	Country
[1]Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Name: Atle Hedløy

Address: 113 Terrasse street
New York, NY 12191-4292
United States of America

Options... 66

Choose 86

Full details... 88

More >>> 90

Cancel 60

82

84

60

66

86

88

90

Dette er en test

FIG. 10

Arendi OneButton Contact Register

Existing addresses with the same name

Name	Address type	Address	Zip	City	Country
[1]Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Name: Atle Hedløy

Address: 113 Terrasse street
New York, NY 12191-4292
United States of America

Buttons: Choose, Full details..., <<< Less

Name: [Dropdown]

Title: [Dropdown]

First: Atle

Middle: [Dropdown]

Last: Hedløy

Suffix: [Dropdown]

Company: [Dropdown]

Address type: Home

Street: 151 University Ave.

City: Palo Alto

State/Province: CA

ZIP/Postal: 94301-1632

Country: USA

Buttons: Add and Choose, Options..., Cancel

Text: Dette er en test

FIG. 11

Atle Hedløy - Contact

File Edit View Insert Format Tools Contact Help

Save and Close

General Details Journal All Fields

Full Name... [Atle Hedløy]

Company: Hedløy, Atle

Address... 113 Terrasse street
New York, NY 12191-4292
United States of America

Business [▼] Home [▼] Business Fax [▼] Mobile [▼]

Job title: Phone: Web page:

File as: [▼]

This is the mailing address

E-mail [▼]

Categories... [▼]

Private

94 96 98 100 102 104

FIG. 12

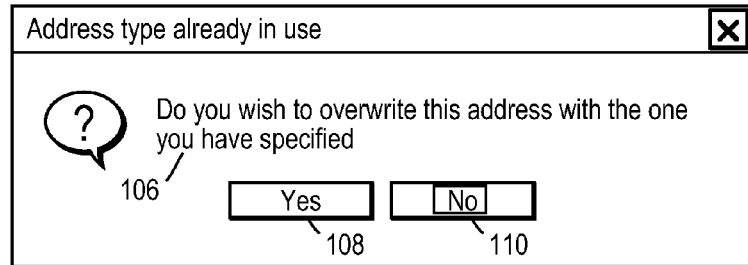


FIG. 13

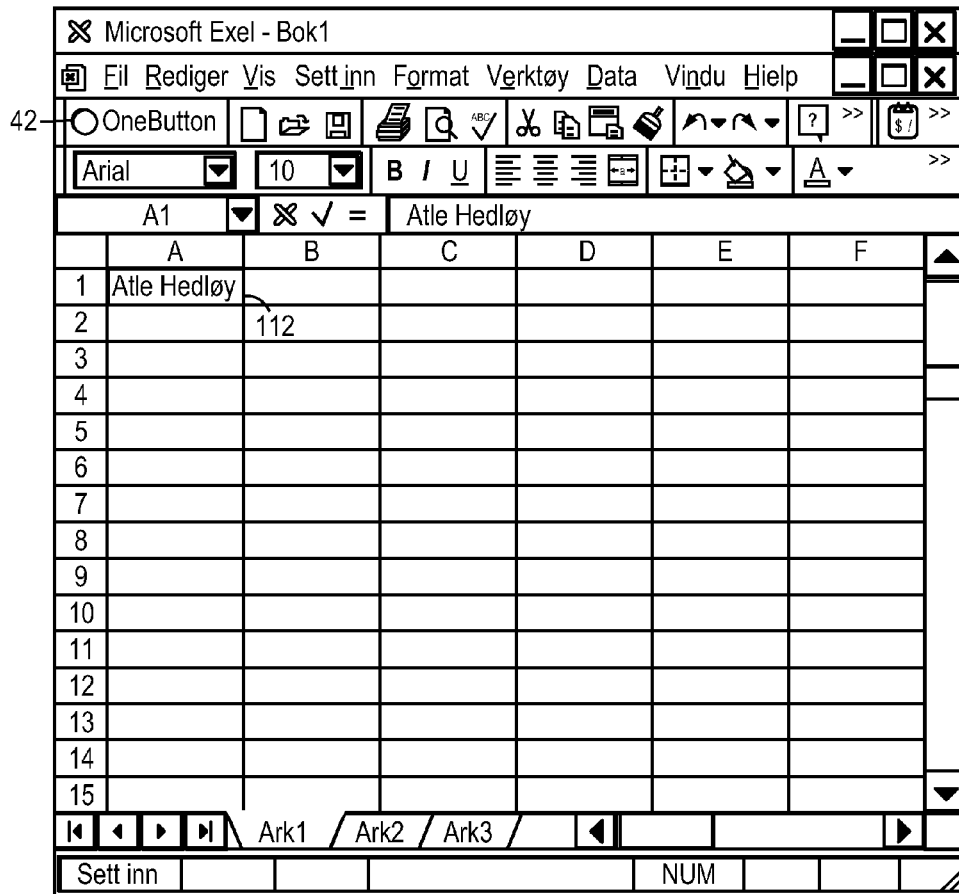


FIG. 14

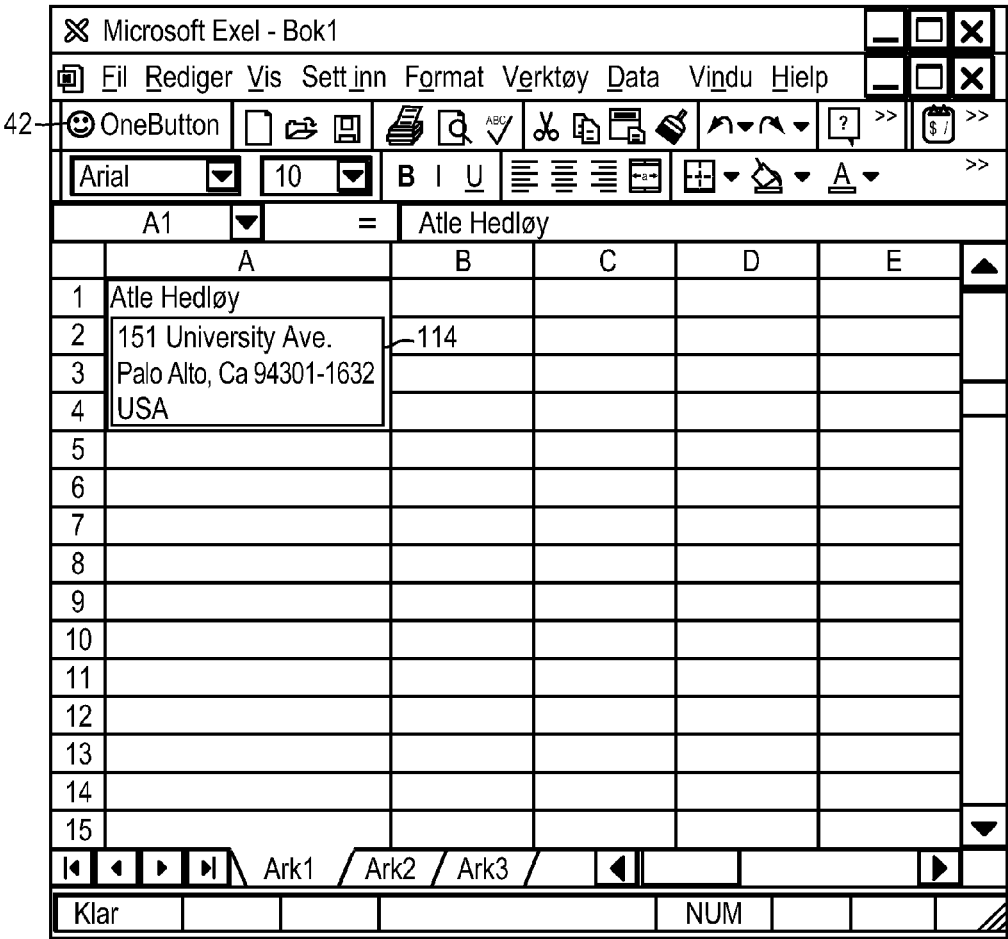


FIG. 15

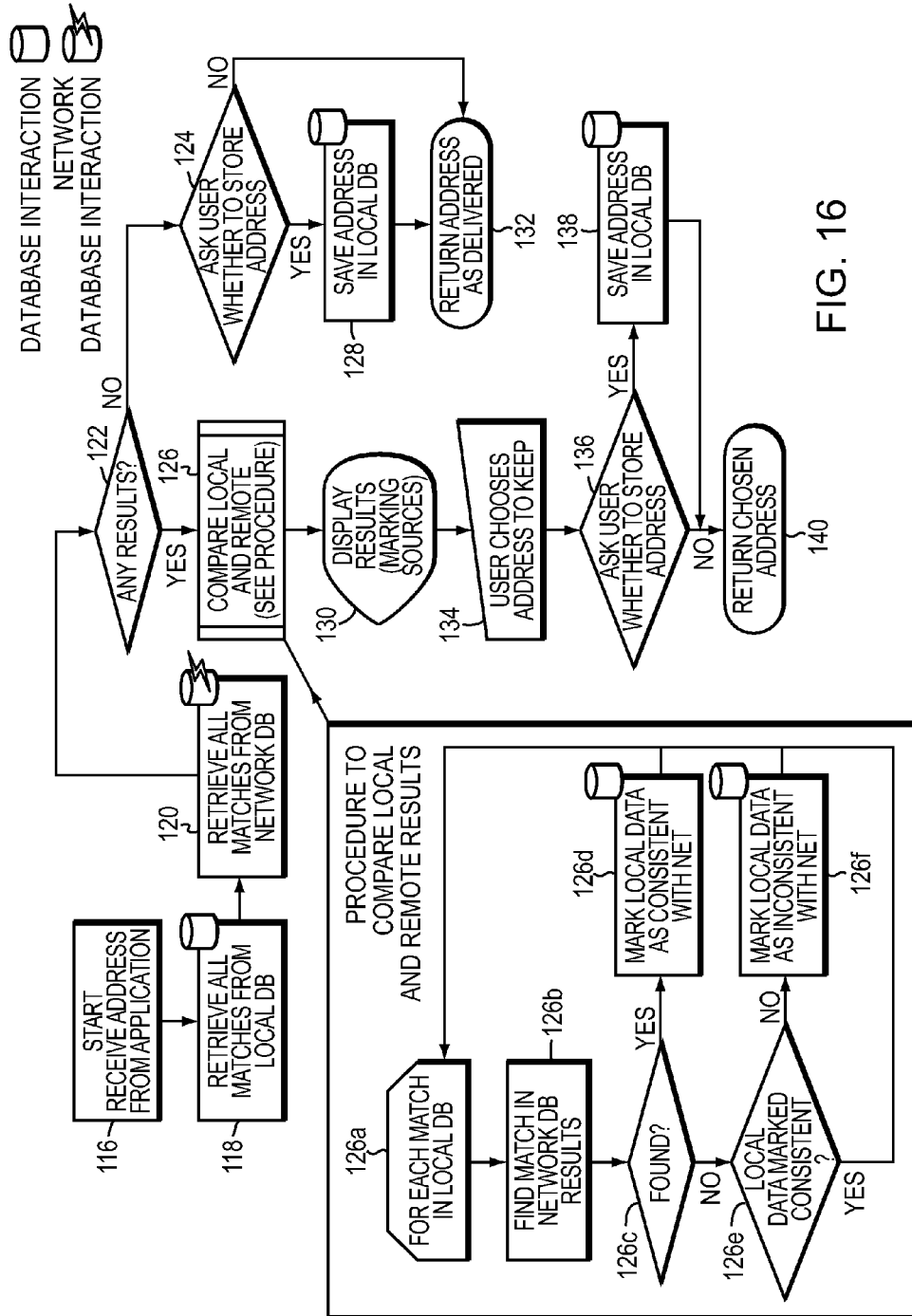


FIG. 16

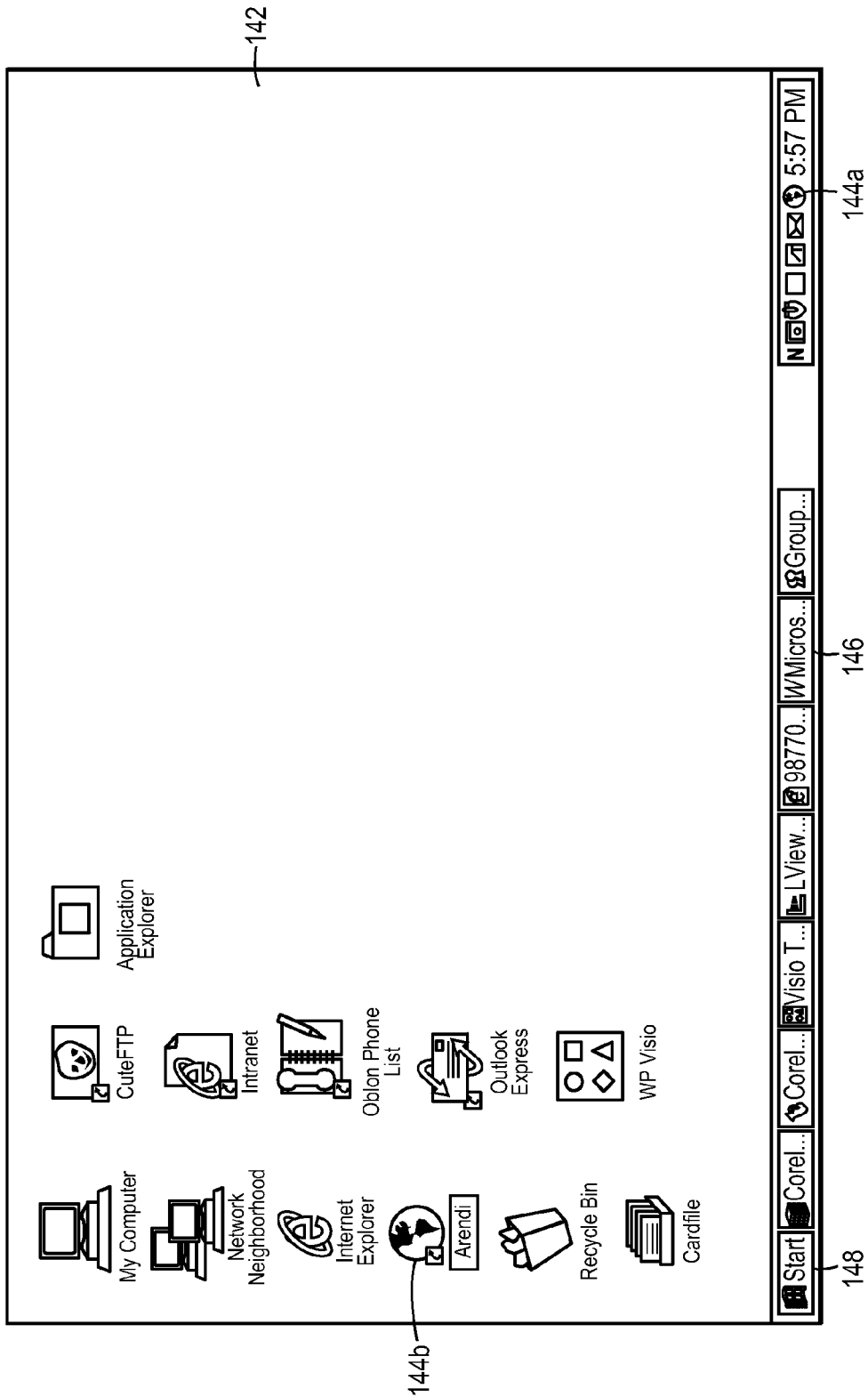


FIG. 17

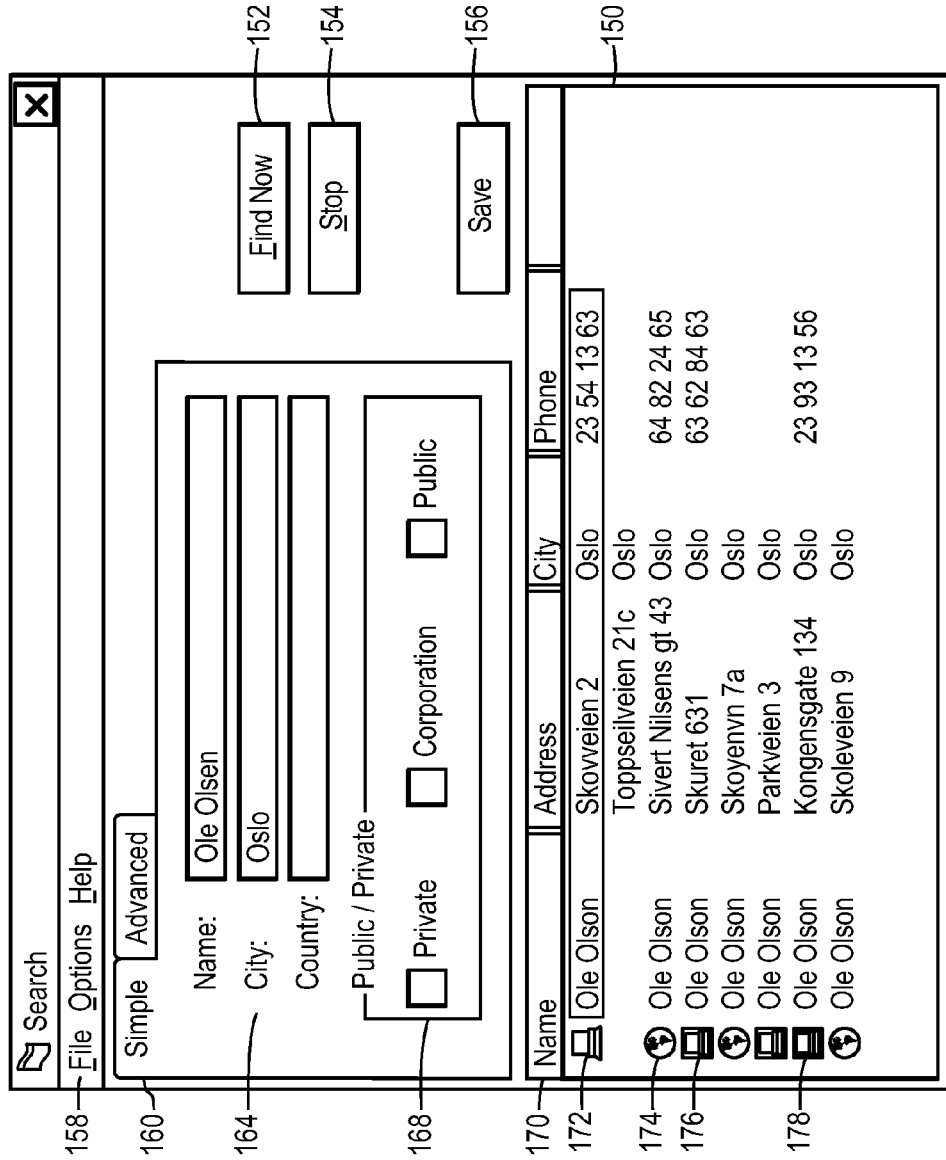


FIG. 18

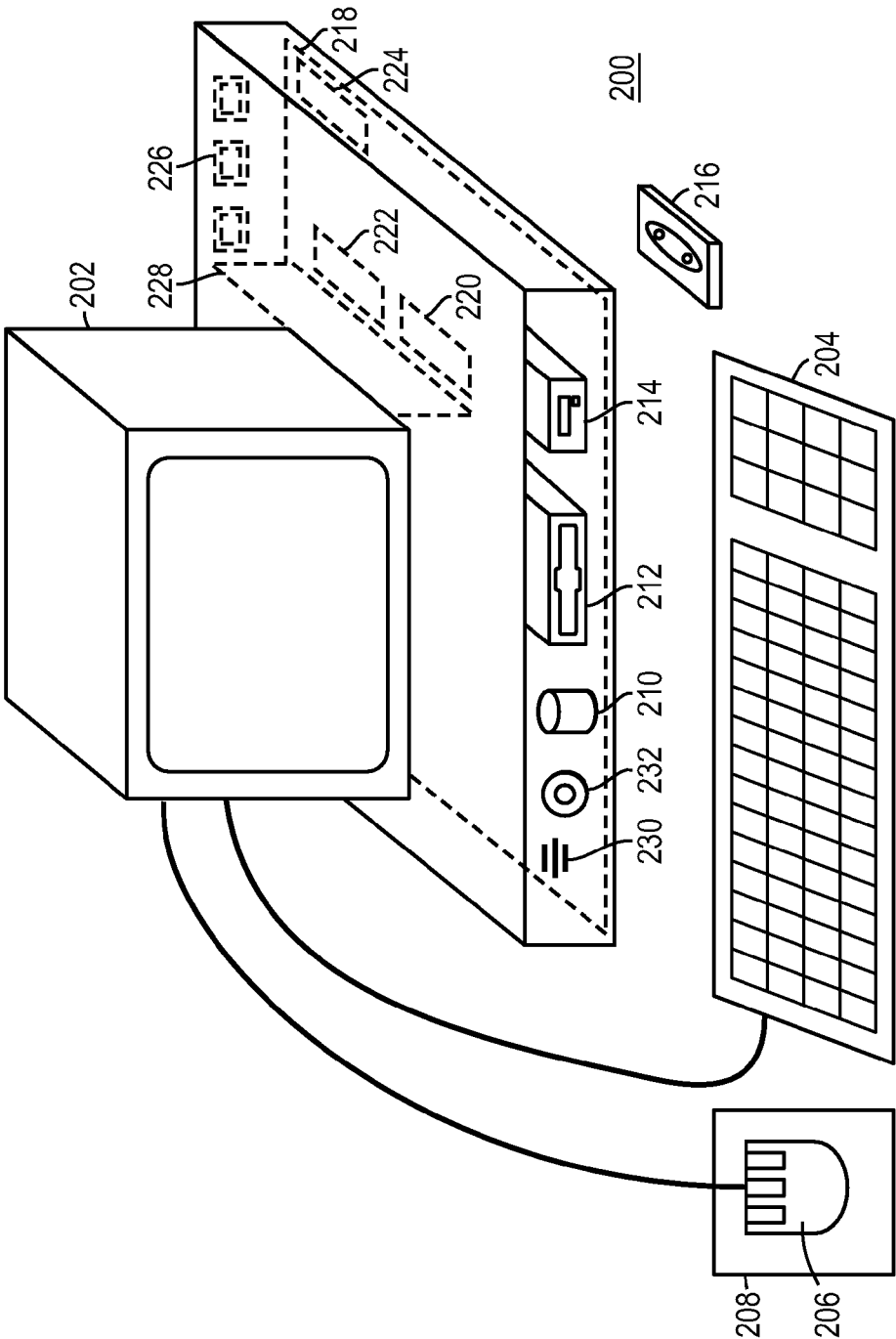


FIG. 19

US 8,306,993 B2

1

**METHOD, SYSTEM AND COMPUTER
READABLE MEDIUM FOR ADDRESSING
HANDLING FROM AN OPERATING SYSTEM**

CROSS REFERENCES TO RELATED
APPLICATIONS

The present application is a continuation of U.S. patent application Ser. No. 09/390,303, which was filed on Sep. 3, 1999, by Atle Hedloy, which is presently incorporated herein by reference.

The present invention is related to commonly owned U.S. patent application Ser. No. 09/189,626, incorporated herein by reference.

BACKGROUND OF THE INVENTION

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 a word processing program, spreadsheet program, etc., or operating system, such as WINDOWS™ operating system, MACINTOSH™ operating system, etc., and coupled to an information management source for providing address handling within a document created by the computer program or within the operating system.

2. Discussion of the Background

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 as 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™, CARD-FILE™, etc.

However, the information in the database must constantly be updated by the user. This requires the user to learn how to use and have access to the database. In this case, a change in the information, such as change in an address or a name, etc., requires the user of the word processor to implement this change in the database, or alternatively, the change is made to the database centrally by a database administrator.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program or operating system.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., or operating system,

2

such as such as WINDOWS™ operating system, MACINTOSH™ operating system, etc.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program or operating system, using an input device provided in the computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program or operating system, using an input device, such as a touch screen, keyboard button, icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program or operating system using an input device provided in the computer program or operating system and coupled to local and/or remote information management source, such as a database program, contact management program, computer network, Internet site, etc.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program or operating system using an input device provided in the computer program or operating system and coupled to local and remote information management source, such as a database program, contact management program, computer network, Internet site, etc., wherein data found in the local database is related to data found in the remote database.

The above and other objects are achieved according to the present invention by providing a novel method, system and computer readable medium for information handling within an operating system, including providing a record retrieval program; providing an input device within a window or screen of the operating system and configured to enter an execute command which initiates a record retrieval from local and remote information sources using the record retrieval program; using the record retrieval program to enter first information into search fields provided in the record retrieval program; entering the execute command using the input device after the step of entering the first information; searching, using the record retrieval program, the local and remote information sources for second information associated with the first information; and displaying the second information in the record retrieval program, when one of the local and remote information sources includes second information associated with the first information.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings wherein:

FIG. 1 is a flow chart illustrating a method for address handling within a computer program, according to an exemplary embodiment of the present invention;

FIG. 2 is a flow chart illustrating a method for address handling within a computer program, according to another exemplary embodiment of the present invention;

FIG. 3 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

US 8,306,993 B2

3

FIG. 4 is a screen shot illustrating a retrieved address in a word processor, according to an exemplary embodiment of the present invention;

FIG. 5 is a screen shot illustrating the inputting of a name and address to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 6 is a screen shot illustrating an add new contact message window, according to an exemplary embodiment of the present invention;

FIG. 7 is a screen shot illustrating a contact register message window, according to an exemplary embodiment of the present invention;

FIG. 8 is a screen shot illustrating an address missing message window, according to an exemplary embodiment of the present invention;

FIG. 9 is a screen shot illustrating a modify contact's address message window, according to an exemplary embodiment of the present invention;

FIG. 10 is a screen shot illustrating a select a contact address register message window, according to an exemplary embodiment of the present invention;

FIG. 11 is a screen shot illustrating a more detailed mode of registering an additional address for the contact register of FIG. 9, according to an exemplary embodiment of the present invention;

FIG. 12 is a screen shot illustrating a contact management program window in a full detailed mode, according to an exemplary embodiment of the present invention;

FIG. 13 is a screen shot illustrating an address already in use message window, according to an exemplary embodiment of the present invention;

FIG. 14 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a spreadsheet, according to an exemplary embodiment of the present invention;

FIG. 15 is a screen shot illustrating a retrieved address in a spreadsheet, according to an exemplary embodiment of the present invention;

FIG. 16 is a flow chart illustrating a method for address handling within an operating system, according to another exemplary embodiment of the present invention;

FIG. 17 is a screen shot illustrating an operating system window including means for address handling therein, according to an exemplary embodiment of the present invention;

FIG. 18 is a screen shot illustrating an address handling program, according to an exemplary embodiment of the present invention; and

FIG. 19 is a schematic illustration of a general purpose computer for performing the processes of the present invention, according to an exemplary embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In an embodiment of the present invention, single button addressing is achieved by providing an input device, such as a touch screen, keyboard, icon, menu, voice command device, etc. (hereinafter called "button"), in a computer program, such as a word processing program, spreadsheet program, etc. (hereinafter called "word processor"), or an operating system, such as WINDOWS™ operating system, MACINTOSH™ operating system, etc., for executing address handling therein.

4

Accordingly, in a word processor or operating system, 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, such as a touch screen button, keyboard button, icon, menu choice, voice command device, etc. A program then executes and retrieves the typed information from the document, and searches a local (i.e., the user's personal computer or an Intranet coupled to the user's personal computer) or remote (i.e., the Internet) information management source, such as a database, file, database program, contact management program, etc. (hereinafter called "database") to determine if the information, such as the name or part of the name typed and searched by the program exists in the database. If the program does not find stored information, such as a name, corresponding to the name or part of the name typed, the user is asked by the program whether the information, such as the name that was not found, should be added to the local database. In addition, the user may enter any other information besides the name, such as addresses, businesses, telephone numbers, fax numbers, e-mail address, etc., so that this other information can be stored in the local database for later use.

If the program finds name(s) and address(es) corresponding to the part of the addressee's name typed, this additional information is automatically entered into the user's word processor, optionally with a confirmation from the user that this is the correct data and stored in the local and/or remote database. If the typed address information does not correspond to data already stored in the local or remote database, after clicking on the button, the program, for example, lets the user decide: (1) if this is new data (e.g., a new address) for an existing contact; (2) if the stored data should be changed to what the user just typed; (3) if this is a new contact with the same name as one already entered into the database; or (4) if the typed address is only to be used once, and therefore not to be stored in the database at all. If, later, for example, a name with several address stored in the local or remote database is recalled, all addresses for this contact will be displayed, so that the correct address can be selected by the user.

The program may be extended to also store and retrieve other information, such as telephone numbers, fax numbers, e-mail addresses, etc. Once the program recalls the telephone numbers, fax numbers, e-mail addresses, etc., the user can command the program to send e-mails, faxes, etc. Similarly, if the user types in the name of a mailing list, the program create merge letters, group e-mails, etc.

Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views, and more particularly to FIGS. 1 and 2 thereof, there is illustrated flow charts of single button addressing, according to exemplary embodiments of the present invention.

In FIG. 1, after the user has inserted the address in the word processor, the user commands the button at step 2 and the program analyzes what the user has typed in the document at step 4. At step 6, the program decides what was found in the document and if the program found nothing in the document or what it found was un-interpretable the program goes to step 8 and outputs an appropriate message to the user and then quits at step 16. The program analyzes what the user has typed in the document at step 4, for example, by analyzing (i) paragraph/line separations/formatting, etc.; (ii) street, avenue, drive, lane, boulevard, city, state, zip code, county designators and abbreviations, etc.; (iii) Mr., Mrs., Sir, Madam, Jr., Sr. designators and abbreviations, etc.; (iv) Inc.,

US 8,306,993 B2

5

Ltd., P.C., L.L.C, designators and abbreviations, etc.; and (v) a database of common male/female names, etc.

If the program finds an e-mail address mailing list/category name telephone number or other information, at step 10 an appropriate action is performed by the program and then the program execution quits at step 16. If the program only finds a name or initials, or the like, the program looks up the name in the database at step 12 and at step 18 the program determines what was found. If the program finds more than one possible contact/address match, at step 20 the program displays menu choices to the user to let him choose an appropriate answer. Then at step 22 the program inserts a correct address and name in the document and then at step 16 the program quits execution. If the program finds one match exactly, i.e., one contact with one address, the program inserts the correct address and name in the document at step 22 and then quits execution at step 16. If the program does not find a name in the database, at step 24 the program prompts the user to specify an address and then quits execution at step 16. If the program at step 6 finds a name and an address, at step 14 the name is looked up in the database. Then, at step 26, if no match is found, at step 28 the program inserts an address and a name which are possibly corrected by the user into the database and then quits execution at step 16. If at step 26, the name and address is found, at step 32 the program either takes no action or displays the data for the user to edit. If at step 26, the name is found but not the address, the program prompts the user for a decision at step 30. If the user decides that this another contact with a same name, the program goes to step 28. If the user decides that this is a one time occurrence, no action is taken and the program quits at step 16. If the user decides that the contact has, for example, moved and that this is a new address, at step 34 one of the old addresses for the contact is replaced with the new one and the program quits at step 16. If the user decides that this is an additional address for the contact, at step 36 the additional address is inserted into the database for that contact and execution quits at step 16.

The flowchart shown in FIG. 2 is similar to the flowchart in FIG. 1, except for some additional steps which will now be discussed. At step 6, if the program only finds a name or a similar name then the name is looked up in the database at step 12, then at step 18 if the program found more than one possible contact/address match, the program displays choices to the user to let him choose an address at step 20. Then at step 21 the user decides whether to insert the selected address into the document. If the user does not decide to select the address into the document the program quits execution at step 16. If the user decides to insert the selected address into the document, the program inserts the address and name into the document at step 22 and then quits at step 16.

If the program finds a name and address in the database at step 6, then at step 14 the program looks up the name in the database and at step 26 the program determines what it has found. If the program does not find the name at step 26, at step 27 the program prompts the user for a decision and review and whether to insert the contact and address. If the user does not decide to insert the contact address, the program quits at step 16. If the user decides to insert the contact address, at step 28 the program inserts the address and name which may be possibly corrected by the user or program in the database and then execution quits at step 16.

If at step 26 the program finds a name and not an address, then at step 29 the name is looked up in the database. Then at step 31 the program decides whether this contact has another address. If the contact does not have another address, at step 33 the program prompts the user for a decision and review and whether to add the address. If the user does not want to add the

6

address at step 33, the program quits at step 16. If the user wants to add the address at step 33 because this is an additional address for the contact, at step 36 the address is inserted in the database for the contact and execution quits at step 16.

At step 30, if the user decides that this is another contact with a same name, then the program goes to step 28. If at step 30 the user decides that this is a one time occurrence, then the program quits at step 16. If at step 30, the user decides that the contact has, for example, moved, the program goes to step 34. If at step 30, the user decides that this is an additional address for the contact, at step 36 the program inserts the address in the database for the contact and then quits at step 16.

Various exemplary screen shots which are generated during execution of the program, according to the present invention, will now be described with reference to FIGS. 3-15 and examples 1-7 as follows.

Example 1

Retrieving an Existing Address from the Database

FIG. 3 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name 40. The user hits the button 42, for example, marked "OneButton" and the program according to the present invention retrieves the name 40 from the document, searches a database for the name 40, and inserts the retrieved address 44 associated with the name 40 into the document as shown in, for example, FIG. 4.

The above example corresponds to steps 2, 4, 6, 12, 18, 22 and 16 in the flow charts of FIGS. 1 and 2.

Example 2

Adding a New Contact to the Database

FIG. 5 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name and address of a new contact 46. The user commands the button 42, for example, marked "OneButton," and the program according to the invention retrieves the new contact 46 from the document, searches a database for the name of the new contact 46 and generates a screen as shown in, for example, FIG. 6. This screen includes a message 50 informing the user that the new contact does not exist in the database, a message 52 including the address retrieved from the document, an address type selection 54, such as home, business, etc., and "OK," "Details," and "Cancel" buttons 58, 58, and 60, respectively.

At this point, the user can cancel the operation by commanding the Cancel button 60, ask the program to store data in the database and return to the document by commanding the OK button 56, or check details before storing data into the database by commanding the Details button 58. If the user commands the Details button 58, as shown in, for example, FIG. 7, a message screen is provided so that the user can review and edit data 62 and the selection 54, store the data 62 and 54 in the database by commanding a "Add and Choose" button 64, see more options by commanding an "Options" button 66, or cancel the operation by commanding the Cancel button 60.

The above example corresponds to steps 2, 4, 6, 14, 26, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 27, 28 and 16 in the flow chart of FIG. 2.

US 8,306,993 B2

7

Example 3

Try to Retrieve Existing Address but Contact is not
in Database

FIG. 3 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name of a contact 40. The user commands the button 42, for example, marked “OneButton,” and the program according to the present invention retrieves the name 40 from the document, searches a database for the name of the contact 40 and generates a screen as shown in, for example, FIG. 8. This screen includes a message 68 informing the user that the contact does not exist in the database and to specify an address, and “OK” buttons 56. At this point when the user commands the OK button 56, the user returns to the document so that the contact’s address can be included as in Example 2 above.

The above example corresponds to steps 2, 4, 6, 12, 18, 24 and 16 in the flow charts of FIGS. 1 and 2.

Example 4

Adding a New Address for an Existing Contact
(Short Version)

FIG. 4 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name and new address of an existing contact 44. The user commands the button 42, for example, marked “OneButton,” and the program according to the present invention retrieves the existing contact 44 from the document, searches a database for the name of the existing contact 44 and generates a screen as shown in, for example, FIG. 9. This screen includes a message 70 informing the user that the contact already exists in the database with an existing address, a message 72 including the existing address, add new contact with same name selection 74, change existing address selection 76, use existing address in document selection 78, add the new address to contact selection 80, the address type selection 54, such as home, business, etc., and the “OK,” “Details,” and “Cancel” buttons 56, 58, and 60 respectively. At this point, the user may select one of the four options 74-80, and command the OK button 56 to execute the selected options. The user can also cancel the operation by commanding the Cancel button 60, or check details before storing data into the database by commanding the Details button 58.

The above example corresponds to steps 2, 4, 6, 14, 26, 28, 30, 34, 36, and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 29, 31, 30, 28, 34, 36, and 16 in the flow chart of FIG. 2.

Example 5

Selecting Between Several Possible Matching
Addresses

FIG. 3 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name and possibly address of at least one existing contact 40. The user commands the button 42, for example, marked “OneButton,” and the program according to the present invention retrieves the existing contact 40 from the document, searches a database for the name of the existing contact 40 and generates a screen as shown in, for example, FIG. 10. This screen includes a message informing the user that the name corresponds to several addresses and possible

8

contacts which already exist in the database, with existing contacts and addresses for selection 82, a message 84 including the full name and address for the contact that the user selects in 82, the Options button 66, a “Choose” button 86, a “Full details” button 88, a “More>>>” button 90, and the Cancel button 60. The above screen indicates to the user that at least one contact with the same name exists, and that there are more than one addresses and/or contacts that match.

At this point, the user may command the Choose button 86 to use the selected address and return to the document, or the user may command the More>>> button 90 to view how the program interpreted what the user typed in the word processor, and possibly change this data, wherein the program generates an updated screen as shown in, for example, FIG. 11. The updated screen includes the data 62 which displays the name typed in the word processor as interpreted by the program, address fields, and the fields for the address type selection 54, such as home, business, etc., which may be changed by the user before the program stores it in the database, the Add and Choose button 64, a “<<<Less” button 90 corresponding to the More>>> button 90 for returning to the screen of FIG. 10, and an “Add this address to the selected contact above” button 92. The user might then command the Add this address to the selected contact above button 92 and the result in the word processor is illustrated in FIG. 4. The user can also cancel the operation by commanding the Cancel button 60, or command the add choose button 64 to add this name and address as a new contact and address, or open the database before storing data into the database by commanding a “Full details” button 88 as will be later described.

The above example corresponds to steps 2, 4, 6, 12, 18, 20, 22, and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 12, 18, 20, 21, 22, and 16 in the flow chart of FIG. 2.

Example 6

Adding a New Address for an Existing Contact
(Long Version)

FIG. 4 illustrates a starting point in word processor document, such as a WORD™ document, wherein the user has typed a name and new address of an existing contact 44. The user commands the button 42, for example, marked “OneButton,” and the program according to the present invention retrieves the existing contact 44 from the document, searches a database for the name of the existing contact 44 and generates a screen as shown in, for example, FIG. 9. As previously described, the screen includes a message 70 informing the user that the contact already exists in the database with an existing address, and the user may command the Details button 58 to see the details of the new address for potentially modify the details before they are stored in the database and the program generates a screen as shown in, for example, FIG. 10. From this screen, the user may choose to use another address than the one he typed, and return to the document, or the user may command the “Full details” button 88 to enter a database program, such as OUTLOOK™, directly as shown in, for example, FIG. 12. In FIG. 12, the database program, such as OUTLOOK™, may include portions 94-104 for allowing the user to modify various pieces of data before they are stored in the database.

Alternatively, in the screen shown in FIG. 10, the user may command the More>>> button 90 at which time the program generates the screen as shown in, for example, FIG. 11 and as previously described. In this screen, the user might then command the Add this address to the selected contact above button 92. If the address typed is already in use, the program

generates a screen including a message **106**, and “Yes” and “No” buttons, **108** and **110**, respectively, as shown in, for example, FIG. **13**. If the user hits the Yes button **108** the program overwrites the contact address with the address specified by the user (e.g., if the contact has moved) and the result in the word processor is shown in, for example, FIG. **4**.

The above example corresponds to steps **2, 4, 6, 12, 14, 26, 28, 30, 34, 36**, and **16** in chart of FIG. 1 and steps **2, 4, 6, 12, 14, 26, 29, 31, 30, 28, 34, 36** and **16** in the flow chart of FIG. 2.

Example 7

Spreadsheet Application

FIG. **14** illustrates a starting point in word processor document, such as an EXCEL™ spreadsheet, wherein the user has typed a name **112**. The user hits the button **42**, for example, marked “OneButton,” and the program according to the present invention retrieves the name **112** from the spreadsheet, searches a database for the name **112**, and inserts the retrieved address **114** into the spreadsheet as shown in, for example, FIG. **15**. Accordingly, the examples 1-6 apply not only to word processor documents, such as WORD™ documents, etc., but to other word processor documents, and spread sheets, such as EXCEL™ spreadsheets, etc.

The above example corresponds to steps **2, 4, 6, 12, 18, 22** and **16** in the flow charts of FIGS. 1 and 2.

Up to this point, the single button addressing program has been described in terms of providing a device for address handling within a computer program, such as a word processor or spread sheet. The following embodiment of the single button addressing program runs on a client (e.g., a computer, cell phone, or palm top device) operating system and integrates local address and phone number data with network data, such as data obtained from an Intranet or the Internet, resolving differences and presenting them in a unified format.

The single button addressing program works within word processing, personal information management, etc., software (e.g., as previously described). The single button addressing program allows the data found on a network Intranet or Internet site to be saved in the local database and checked against network data as it changes, without the network database being aware of the local database. The network can be a public network, such as the Internet, or a private data network, such as an Intranet. The local database can be a database management system, such as Microsoft ACCESS™, Microsoft SQL server, etc., running on the local computer or any accessible server. The local database can also be an application, such as a personal information manager like Microsoft OUTLOOK™ or Symantec Act!™, etc., that maintains a database therein. Similarly, the remote database may be a public or private data service, a Web-based data source, or a CD-ROM of information used in the user’s computer or computer network.

The invention according to the present embodiment performs data integration in the following way: (1) the address handling function is typically implemented as a subprogram within a larger program, such as the single button address program provided in a word processor as previously described or as single button address program provided in an operating system as will be later described; (2) the subprogram is started with a complete or partial name and address; (3) the subprogram queries the local and remote databases and compares the results; (4) the subprogram provides user interface for the user to select the appropriate result wherein the choices are marked based on whether the data is remote or

local. If data was originally remote and the user saved it locally, the user is alerted if the data has changed on the remote database; (5) if the data chosen by the user is different or not present in the local database, the user is given the opportunity to save the data locally; and (6) the chosen address is returned to the calling program, which may, for example, include it in a document.

FIG. **16** illustrates a method for address handling from an operating system, such as WINDOWS™ operating system, MACINTOSH™ operating system, etc., according to the present embodiment of the invention. In FIG. **16**, at step **116**, an address is received from an application or is entered directly into the single button addressing subprogram. At step **118**, the subprogram retrieves all matches from a local database. At step **120**, the subprogram retrieves all matches from a network or remote database.

At step **122**, the subprogram determines whether or not there are any matching address results. If there are no matching results, the user is given the opportunity to store or not to store the address at step **124**. If the user chooses to store the address, at step **128** the address is stored in the local database. If the user chooses not to store the address, at step **132** the address is returned to the calling program as delivered to the user and without being stored in the local database.

If at step **122** the subprogram determines that there are matching results, the local and remote matching database results are compared at step **126** as follows. At step **126a**, for each match in the local database, step **126b** determines whether or not there is a corresponding match in the remote or network database results. At step **126c**, if a match is found between the local and remote databases, the matching result is marked as local data which is consistent with the network data. At step **126c**, if no match is found between the local and remote databases, step **126e** determines whether or no the local data was marked as consistent with the network data. If the local data was not marked as consistent with the network data, at step **126f** the local data is marked as inconsistent with the network data. If the local data was marked as consistent with the network data, control transfers back to step **126a** to process the next match in the local database.

FIG. **17**, illustrates an exemplary operating system screen **142**, such as a WINDOWS 95™ operating system screen, including the single button addressing subprogram implemented as tool bar subprogram **144a** or as a desktop icon subprogram **144b**. The single button addressing subprogram can also be launched from a word processing application **146**, as previously discussed, or via the WINDOWS 95™ Start menu **148**.

FIG. **18** illustrates an exemplary search screen **150** generated by the single button addressing subprogram after it has been launched. In this example, a user ran a search (i.e., from the standalone single button addressing subprogram) against local data stored in, for example, Microsoft OUTLOOK™ and remote data stored in, for example, a remote web service.

In FIG. **18**, the search screen **150** includes, for example, a Find Now button **152** for executing a search, a Stop button for stopping a search in progress and a Save button **156** for saving found data. The search screen **150** includes, for example, File, Option and Help menu selections **158**, search criteria **164**, including, for example, fields for inputting a Name, a City and/or Country, and Public/Private indicators **168**, for indicating Private, Corporation and/or Public database searching options. The search results are displayed in a search window including Name, Address, City and Phone sort buttons **170**, which sort the search results according to the button selected.

The search results are further marked with status indicators **172-178**. Status indicator **172**, for example, includes a com-

US 8,306,993 B2

11

puter icon with no color which indicates that the same data was found both locally and on the remote database (e.g., the Internet). Status indicator **174**, for example, includes a globe icon which indicates that the same data was found on the remote database, but not on the local database. Status indicator **176**, for example, includes a computer icon of a first color (e.g., yellow) which indicates that the same data was found on the local database, but not on the remote database. Status indicator **178**, for example, includes a computer icon of a second color (e.g., red) which indicates that the data was originally added to the local database from the remote database, but now is no longer found on the remote database.

FIG. **19** 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 Enhanced IDE 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 a 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**.

As stated above, the system includes at least one computer readable medium, or alternatively, the computer readable medium may be accessed through various paths, such as networks, internet, drives, etc. Examples of computer readable media are compact discs, hard disks, floppy disks, tape, magneto-optical disks, PROMs (EPROM, EEPROM, Flash EPROM), DRAM, SRAM, SDRAM, etc. Stored on any one or on a combination of computer readable media, the present invention includes software for controlling both the hardware of the computer **200** and for enabling the computer **200** to interact with a human user. Such software may include, but is not limited to, device drivers, operating systems and user applications, such as development tools. Such computer readable media further includes the computer program product of the present invention for performing any of the processes according to the present invention, described above (see, e.g., FIGS. **1-18**). The computer code devices of the present invention can be any interpreted or executable code mechanism, including but not limited to scripts, interpreters, dynamic link libraries, Java classes, and complete executable programs, etc.

The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

Address handling, according to this invention, is a significant simplification relative to existing methods, and requires little or no training on the part of a user, as correct addresses are retrieved with a minimal number of user commands, "clicks", keystrokes, etc. In addition, a program according to

12

the present invention, can be programmed and created in most existing programming languages and be connected to most modern word processors. Therefore, according to the present invention, the process of creating and updating records in an address database is significantly simplified, since this may now be performed directly from the word processor.

Although the present invention is defined in terms of word processing documents, such as WORD™ documents and Excel™ spreadsheets, the present invention is applicable to all types of word processing documents, such as NOTEPAD™, WORDPAD™, WORDPERFECT™, QUATROPRO™, AMIPRO™, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of information management or database programs, such as OUTLOOK™, etc., the present invention is applicable to all types of information management or database programs, such as ACCESS™, ORACLE™, DBASE™, RBASE™, CARD-FILE™, including "flat files," etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of operating systems, such as WINDOWS™, MACINTOSH™, etc., the present invention is applicable to all types of operating systems, such as UNIX™, LINUX™, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of providing an input device, such as a button **42** in a word processor for address handling therein, the present invention may be practiced with all types of input devices, such as a touch screen, keyboard button, icon, menu, voice command device, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving information from a document before searching a database, the user may select the information in the document to be searched by the program in the database (e.g., by highlighting, selecting, italicizing, underlining, etc.), as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving a name or portion thereof from a document before searching a database, the program may retrieve an address or portion thereof from the document before searching the database and insert, correct, complete, etc., the retrieved address based on the information found in the database corresponding to the retrieved address or portion thereof, as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of an embodiment as described with respect to FIGS. **16-18**, all of the relevant features as described with respect to FIGS. **1-15** apply to the embodiment as described with respect to FIGS. **16-18**, as will be readily apparent to those skilled in the art. Similarly, although the present invention is defined in terms of an embodiment as described with respect to FIGS. **1-15**, all of the relevant features as described with respect to FIGS. **16-18** apply to the embodiment as described with respect to FIGS. **1-15**, as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of an address handling program provided in an operating system environment, such as WINDOWS™, MACINTOSH™, etc., of a personal computer, the program may run on an operating system environment, such as WINDOWS CE™, etc., of a client, such as cell phone, palm top device, personal organizer, etc., as will be readily apparent to those skilled in the art.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of the

US 8,306,993 B2

13

appended claims, the invention may be practiced otherwise than as specifically described herein.

This application claims priority and contains subject matter related to Norwegian patent application No. 984066 filed on Sep. 3, 1998, the entire contents of which are hereby incorporated by reference.

What is claimed is:

1. A computer implemented method for information handling, the method comprising:

providing access to a contact database that can also be separately accessed and edited by a user and wherein the contact database includes at least three fields for storing contact information associated with each of one or more contacts, each of the at least three fields within the contact database being specific to a particular type of contact information selected from the group consisting of name, title, address, telephone number, and email address;

analyzing in a computer process textual information in a document configured to be stored for later retrieval to identify a portion of the document as first contact information, without user designation of a specific part of the textual information to be subject to the analyzing, wherein the first contact information is at least one of a name, a title, an address, a telephone number, and an email address;

after identifying the first contact information, performing at least one action from a set of potential actions, using the first contact information previously identified as a result of the analyzing, wherein the set of potential actions includes:

(i) initiating an electronic search in the contact database for the first contact information while it is electronically displayed in order to find whether the first contact information is included in the contact database; and

when a contact in the contact database includes the first contact information, if second contact information in the contact database is associated with that contact, electronically displaying at least a portion of the second contact information, wherein the second contact information is at least one of a name, a title, an address, a telephone number, and an email address;

(ii) initiating electronic communication using the first contact information; and

(iii) allowing the user to make a decision whether to store at least part of the first contact information in the contact database as a new contact or to update an existing contact in the contact database;

wherein the computer implemented method is configured to perform each one of action (i), action (ii), and action (iii) using the first contact information previously identified as a result of the analyzing; and

providing for the user an input device configured so that a single execute command from the input device is sufficient to cause the performing.

2. A method according to claim 1, wherein the computer implemented method is embodied in a client and the client is selected from a group consisting of a computer, a cell phone, a palm top device, and a personal organizer.

3. A method according to claim 2, wherein the first contact information is a name, the second contact information is an address, and the client is a computer.

4. A method according to claim 2, wherein the first contact information is a telephone number.

14

5. A method according to claim 2, wherein the first contact information is a telephone number, the second contact information is a name, and the client is a cell phone.

6. A method according to claim 1, wherein the input device is a menu and the single execute command includes the user's selection of a menu choice from the menu.

7. A method according to claim 1, wherein the input device is a button within a window.

8. A method according to claim 1, wherein when the first contact information is an e-mail address, initiating electronic communication using the first contact information comprises creating an e-mail using the e-mail address.

9. At least one non-transitory computer readable medium encoded with instructions which when loaded on at least one computer, establish processes for information handling, the processes comprising:

providing access to a contact database that can also be separately accessed and edited by a user and wherein the contact database includes at least three fields for storing contact information associated with each of one or more contacts, each of the at least three fields within the contact database being specific to a particular type of contact information selected from the group consisting of name, title, address, telephone number, and email address;

analyzing in a computer process textual information in a document configured to be stored for later retrieval to identify a portion of the document as first contact information, without user designation of a specific part of the textual information to be subject to the analyzing, wherein the first contact information is at least one of a name, a title, an address, a telephone number, and an email address;

after identifying the first contact information, performing at least one action from a set of potential actions, using the first contact information previously identified as a result of the analyzing, wherein the set of potential actions includes:

(i) initiating an electronic search the contact database for the first contact information while it is electronically displayed in order to find whether the first contact information is included in the contact database; and

when a contact in the contact database includes the first contact information, if second contact information in the contact database is associated with that contact, electronically displaying at least a portion of the second contact information, wherein the second contact information is at least one of a name, a title, an address, a telephone number, and an email address;

(ii) initiating electronic communication using the first contact information; and

(iii) allowing the user to make a decision whether to store at least part of the first contact information in the contact database as a new contact or to update an existing contact in the contact database;

wherein the computer implemented method is configured to perform each one of action (i), action (ii), and action (iii) using the first contact information previously identified as a result of the analyzing; and

providing for the user an input device configured so that a single execute command from the input device is sufficient to cause the performing.

10. At least one non-transitory computer readable medium according to claim 9, wherein the at least one non-transitory computer readable medium is embodied in a client and the client selected from a group consisting of a computer, a cell phone, a palm top device, and a personal organizer.

US 8,306,993 B2

15

11. At least one non-transitory computer readable medium according to claim 10, wherein the first contact information is a name, the second contact information is an address, and the client is a computer.

12. At least one non-transitory computer readable medium according to claim 10, wherein the first contact information is a telephone number.

13. At least one non-transitory computer readable medium according to claim 10, wherein the first contact information is a telephone number, the second contact information is a name, and the client is a cell phone.

14. At least one non-transitory computer readable medium according to claim 9, wherein the input device is a menu and the single execute command includes the user's selection of a menu choice from the menu.

15. At least one non-transitory computer readable medium according to claim 9, wherein the input device is a button within a window.

16. At least one non-transitory computer readable medium according to claim 9, wherein when the first contact information is an e-mail address, initiating electronic communication using the first contact information comprises creating an e-mail using the e-mail address.

17. An apparatus for information handling, the apparatus comprising:

a processor; and

a memory storing instructions executable by the processor to perform processes that include:

providing access to a contact database that can also be separately accessed and edited by a user and wherein the contact database includes at least three fields for storing contact information associated with each of one or more contacts, each of the at least three fields within the contact database being specific to a particular type of contact information selected from the group consisting of name, title, address, telephone number, and email address;

analyzing in a computer process textual information in a document configured to be stored for later retrieval to identify a portion of the document as first contact information, without user designation of a specific part of the textual information to be subject to the analyzing, wherein the first contact information is at least one of a name, a title, an address, a telephone number, and an email address;

after identifying the first contact information, performing at least one action from a set of potential actions, using the first contact information previously identified as a result of the analyzing, wherein the set of potential actions includes:

(i) initiating an electronic search in the contact database for the first contact information while it is electronically displayed in order to find whether the first contact information is included in the contact database; and

when a contact in the contact database includes the first contact information, if second contact information in the contact database is associated with that contact, electronically displaying at least a portion of the second contact information, wherein the second contact information is at least one of a name, a title, an address, a telephone number, and an email address;

(ii) initiating electronic communication using the first contact information; and

(iii) allowing the user to make a decision whether to store at least part of the first contact information in the

16

contact database as a new contact or to update an existing contact in the contact database;

wherein the computer implemented method is configured to perform each one of action (i), action (ii), and action (iii) using the first contact information previously identified as a result of the analyzing; and

providing for the user an input device configured so that a single execute command from the input device is sufficient to cause the performing.

18. An apparatus according to claim 17, wherein the apparatus is selected from a group consisting of a computer, a cell phone, a palm top device, and a personal organizer.

19. An apparatus according to claim 18, wherein the first contact information is a name, the second contact information is an address, and the apparatus is a computer.

20. An apparatus according to claim 18, wherein the first contact information is a telephone number.

21. An apparatus according to claim 18, wherein the first contact information is a telephone number, the second contact information is a name, and the apparatus is a cell phone.

22. An apparatus according to claim 17, wherein the input device is a menu and the single execute command includes the user's selection of a menu choice from the menu.

23. An apparatus according to claim 17, wherein the input device is a button within a window.

24. An apparatus according to claim 17, wherein when the first contact information is an e-mail address, initiating electronic communication using the first contact information comprises creating an e-mail using the e-mail address.

25. A computerized method for information handling, the method comprising:

displaying information in a document electronically using a computer program;

electronically analyzing the information to identify a portion of that information as contact information including at least one of a name without an address and a name with an address;

providing an input device configured to allow a user to use the input device to command the program to perform at least one of:

(i) inserting address information from an information source and associated with the name into the document, and

(ii) storing at least part of the contact information in the information source;

wherein the program is configured to perform both actions (i) and action (ii);

during the displaying, receiving an execute command from the input device, wherein accessing and manipulating the input device are the only user actions required to cause initiation and completion of the analyzing;

when the contact information is identified as including a name without an address, electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes the name, if address information in the information source is associated with the name, causing insertion of the address information into the document; and

when the contact information is identified as including a name with an address, (a) electronically prompting the user with an option to save electronically in the information source at least some of the contact information, and (b) electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

US 8,306,993 B2

17

when the information source includes at least one contact with the name, prompting the user to make a decision whether to store the name and address as a new contact or to update one of the at least one contact.

26. At least one non-transitory computer readable medium encoded with instructions which when loaded on at least one computer, establish processes for information handling, comprising:

displaying information in a document electronically using a computer program;

electronically analyzing the information to identify a portion of that information as contact information including at least one of a name without an address and a name with an address;

providing an input device configured to allow a user to use the input device to command the program to perform at least one of:

(i) inserting address information from an information source and associated with the name into the document, and

(ii) storing at least part of the contact information in the information source;

wherein the program is configured to perform both action (i) and action (ii);

during the displaying, receiving an execute command from the input device, wherein accessing and manipulating the input device are the only user actions required to cause initiation and completion of the analyzing;

when the contact information is identified as including a name without an address, electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes the name, if address information in the information source is associated with the name, causing insertion of the address information into the document; and

when the contact information is identified as including a name with an address, (a) electronically prompting the user with an option to save electronically in the information source at least some of the contact information, and (b) electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes at least one contact with the name, prompting the user to make a decision whether to store the name and address as a new contact or to update one of the at least one contact.

27. A computerized method for information handling, the method comprising:

displaying information in a document electronically using a computer program;

electronically analyzing the information to identify a portion of that information as contact information including at least a name;

providing an input device configured to allow a user to use the input device to command the program to perform at least one action selected from the group consisting of:

(i) inserting address information from an information source and associated with the name into the document, and

(ii) storing at least part of the contact information in the information source;

wherein the program is configured to perform both action (i) and action (ii);

18

during the displaying, receiving an execute command from the input device, wherein accessing and manipulating the input device are the only user actions required to cause initiation and completion of the analyzing;

when the program performs action (i), electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes the name, if address information in the information source is associated with the name, causing insertion of the address information into the document; and

when the program performs action (ii), electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes at least one contact with the name, prompting the user to make a decision whether to store the name as a new contact or to update one of the at least one contact.

28. A computerized method for information handling according to claim 27, further comprising:

when the program performs action (i) and the information source includes more than one address associated with the name, prompting the user to choose one of the addresses to use for insertion into the document.

29. At least one non-transitory computer readable medium encoded with instructions which when loaded on at least one computer, establish processes for information handling, comprising:

displaying information in a document electronically using a computer program;

electronically analyzing the information to identify a portion of that information as contact information including at least a name;

providing an input device configured to allow a user to use the input device to command the program to perform at least one action selected from the group consisting of:

(i) inserting address information from an information source and associated with the name into the document, and

(ii) storing at least part of the contact information in the information source;

wherein the program is configured to perform both action (i) and action (ii);

during the displaying, receiving an execute command from the input device, wherein accessing and manipulating the input device are the only user actions required to cause initiation and completion of the analyzing;

when the program performs action (i), electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes the name, if address information in the information source is associated with the name, causing insertion of the address information into the document; and

when the program performs action (ii), electronically searching for the name in the information source, in order to find whether the name is included in the information source; and

when the information source includes at least one contact with the name, prompting the user to make a decision whether to store the name as a new contact or to update one of the at least one contact.

US 8,306,993 B2

19

30. At least one non-transitory computer readable medium according to **29**, wherein the instructions further establish processes wherein:

when the program performs action (i) and the information source includes more than one address associated with

20

the name, prompting the user to choose one of the addresses to use for insertion into the document.

* * * * *

EXHIBIT 3

(12) **United States Patent**
Hedloy

(10) **Patent No.:** **US 7,496,854 B2**
(45) **Date of Patent:** **Feb. 24, 2009**

(54) **METHOD, SYSTEM AND COMPUTER READABLE MEDIUM FOR ADDRESSING HANDLING FROM A COMPUTER PROGRAM**

FOREIGN PATENT DOCUMENTS

EP 0 093 250 A2 9/1983

(75) Inventor: **Atle Hedloy**, Stabekk (NO)

(Continued)

(73) Assignee: **Arendi Holding Limited**, Grand Cayman (KY)

OTHER PUBLICATIONS

User Manual For AddressMate and AddressMate Plus 1994-1995 by AddressMate Software.

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 20 days.

(Continued)

Primary Examiner—Sy D. Luu

(21) Appl. No.: **09/923,134**

(74) *Attorney, Agent, or Firm*—Cesari and McKenna, LLP

(22) Filed: **Aug. 6, 2001**

(57) **ABSTRACT**

(65) **Prior Publication Data**

US 2002/0054092 A1 May 9, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/189,626, filed on Nov. 10, 1998, now Pat. No. 6,323,853.

(51) **Int. Cl.**
G06F 3/00 (2006.01)

(52) **U.S. Cl.** **715/780**; 715/816

(58) **Field of Classification Search** 345/700, 345/705, 710, 744, 764, 804, 805, 808, 809, 345/835, 840, 853, 968; 707/1, 3, 500, 501.1, 707/505, 507, 513, 515, 530; 715/500, 501.1, 715/505, 507, 513, 515, 530, 780, 816
See application file for complete search history.

A method, system and computer readable medium for providing for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

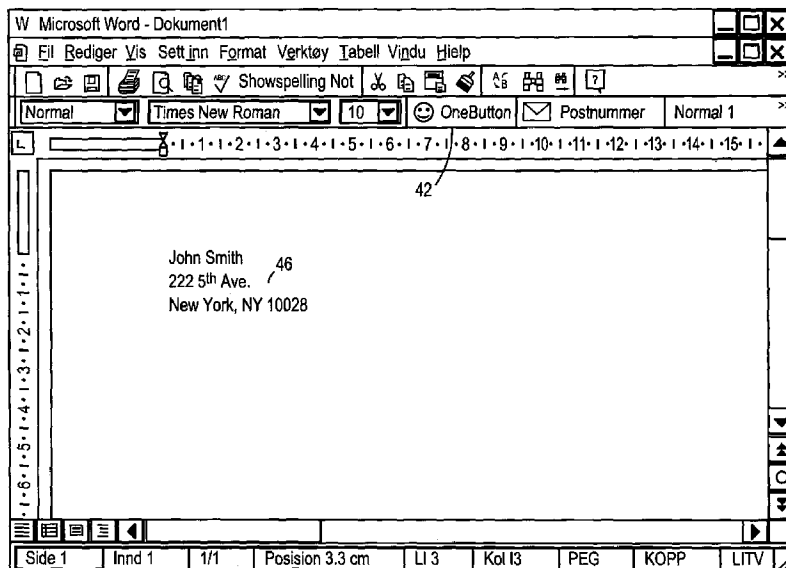
(56) **References Cited**

U.S. PATENT DOCUMENTS

4,674,065 A 6/1987 Lange et al.

(Continued)

101 Claims, 14 Drawing Sheets



US 7,496,854 B2

Page 2

U.S. PATENT DOCUMENTS

5,226,117	A	7/1993	Miklos	345/853
5,267,155	A	11/1993	Buchanan et al.	
5,331,555	A	7/1994	Hashimoto et al.	707/531
5,375,200	A *	12/1994	Dugan et al.	345/804
5,392,386	A	2/1995	Chalas	
5,416,901	A	5/1995	Torres	345/835
5,491,783	A	2/1996	Douglas et al.	345/846
5,491,784	A	2/1996	Douglas et al.	345/810
5,500,859	A	3/1996	Sharma et al.	370/468
5,530,853	A	6/1996	Schell et al.	707/1
5,546,447	A	8/1996	Skarbo et al.	379/142.05
5,576,955	A	11/1996	Newbold et al.	
5,606,712	A	2/1997	Hidaka	712/1
5,640,565	A	6/1997	Dickinson	707/103 R
5,666,502	A	9/1997	Capps	345/811
5,708,804	A	1/1998	Goodwin et al.	707/3
5,724,597	A	3/1998	Cuthbertson et al.	707/531
5,732,229	A	3/1998	Dickinson	345/764
5,761,656	A	6/1998	Ben-Shachar	707/4
5,781,189	A	7/1998	Holleran et al.	345/826
5,793,972	A	8/1998	Shane	709/219
5,794,228	A	8/1998	French et al.	
5,794,259	A	8/1998	Kikinis	707/507
5,799,302	A	8/1998	Johnson et al.	707/7
5,805,886	A	9/1998	Skarbo et al.	709/318
5,815,830	A	9/1998	Anthony	
5,826,257	A	10/1998	Snelling, Jr.	707/4
5,835,089	A	11/1998	Skarbo et al.	345/751
5,859,636	A	1/1999	Pandit	707/501.1
5,864,848	A	1/1999	Horvitz et al.	
5,873,107	A	2/1999	Borovoy et al.	707/501.1
5,884,309	A	3/1999	Vanechanos, Jr.	707/10
5,893,093	A	4/1999	Wills	707/5
5,896,533	A	4/1999	Ramos et al.	709/217
5,907,838	A	5/1999	Miyasaka et al.	707/4
5,913,214	A	6/1999	Madnick et al.	707/10
5,924,090	A	7/1999	Krellenstein	707/5
5,926,808	A	7/1999	Evans et al.	707/3
5,930,471	A	7/1999	Milewski et al.	709/204
5,946,647	A	8/1999	Miller et al.	
5,999,938	A	12/1999	Bliss et al.	
6,006,218	A	12/1999	Breese et al.	
6,021,403	A	2/2000	Horvitz et al.	
6,026,398	A	2/2000	Brown et al.	707/5
6,067,565	A	5/2000	Horvitz	
6,085,201	A	7/2000	Tso	707/505
6,085,226	A	7/2000	Horvitz	
6,182,133	B1	1/2001	Horvitz	
6,223,570	B1	5/2001	Horvitz et al.	
6,260,035	B1	7/2001	Horvitz et al.	
6,262,730	B1	7/2001	Horvitz et al.	
6,323,853	B1 *	11/2001	Hedloy	345/810

FOREIGN PATENT DOCUMENTS

EP 0093250 11/1983

OTHER PUBLICATIONS

Microsoft Word 97 Help File entitled "Automatically check spelling and grammar as you type".

Microsoft Word 97 Help File entitled "Turn automatic changes on or off".

Microsoft Word 97 Help File entitled "Automatically correct text".

Microsoft Word 97 Help File entitled "Field Codes: Hyperlink Field".

Microsoft Word 97 Help File entitled "Change the contents of an AutoCorrect entry".

Wood, Andrew, et al., CyberDesk: Automated Integration of Desktop and Network Services, Gvu Technical Report, OIT-GVU-97-11, May 1997.

Gregory D. Abowd, Anidn Dey and Andy M. Wood, Applying Dynamic Integration as a Software Infrastructure for Context-Aware Computing, Gvu Technical Report, GIT-GVU-97-18, Sep. 1997.

Gregory D. Abowd, Anind Dey, Robert Orr and Jason Brotherson, Context-awareness in wearable and ubiquitous computing, Gvu Technical Report, GIT-GVU-97-11, Mar. 1997.

Apple Data Detectors User's Manual, Jul. 1, 1997.

Apple Internet Address Detector User's Manual, Aug. 28, 1997.

Apple Introduces Internet Address Detectors, Sep. 8, 1997.

Contextual Menu Manager/Apple Data Detectors.

CoStar User Manual for AddressMate and AddressMate Plus.

Ctags (UNIX Command).

Anind K. Dey and Gregory D. Abowd, CyberDesk: The Use of Perception in Context-Aware Computing, PUI Workshop Submission, Proc. of 1997 Workshop on Perceptual User Interfaces (PUI '97), pp. 26-27, Oct. 1997.

Anind K. Dey, Context-Aware Computing: The CyberDesk Project, Future Computing Environments, AAAI '98 Spring Symposium, Stanford University, pp. 51-55, Mar. 23-25, 1998.

Anind K. Dey, Gregory D. Abowd and Andrew Wood, CyberDesk: a framework for providing self-integrating context-aware services, Knowledge-Based Systems, vol. 11, No. 1, pp. 3-13, Sep. 1998.

Anind K. Dey, Gregory D. Abowd, Mike Pinkerton and Andrew Wood, CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services, Gvu Technical Report, GIT-GVU-97-10, May 1997.

L. Nancy Garrett, Karen E. Smith and Norman Meyrowitz, Intermedia: Issues, Strategies, and Tactics in the Design of a Hypermedia Document System, (c) 1986, pp. 163-174.

Eve Wilson, Links and Structures in hypertext databases for law, Proceedings of the First European Conference on Hypertext, INRIA, Nov. 1990.

Mike Langberg, 'Innovation is at the heart of what we do', Apple breaks new ground by displaying what's on its drawing board, Mercury News, Aug. 7, 1996, pp. 1-2.

Henry Lieberman, Bonnie A. Nardi and David Wright, Training Agents to Recognize Text by Example, Proc. of the Third Annual Conference on Autonomous Agents, Seattle, WA, pp. 116-122, 1999.

User's Guide, Microsoft Word 6.0, for Windows and Macintosh.

Bonnie A. Nardi, James R. Miller and David J. Wright, Collaborative, Programmable Intelligent Agents, Website, Mar. 1998, pp. 1-11.

Milind S. Pandit and Sameer Kalbag, The Selection Recognition Agent: Instance Access to Relevant Information and Operations, Proc. of Intelligent User Interfaces 1997, Orlando, FL, 1997.

Spell, iSpell Spellout.

WikiWikiWeb.

United States Court of Appeals for the Federal Circuit, Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Judge Ernest C. Torres. Brief for Defendant-Cross Appellant Frank E. Scherkenbach Jul. 7, 2005.

United States District Court for the District of Rhode Island, 02-CV-343 (ECT) Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law That the '853 Patent is Invalid, Oct. 15, 2004.

United States District Court of Rhode Island, CIV. A. No. 02-CV-343 (ECT), Plaintiffs' Reply Memorandum in Support of Their Motion for New Trial.Francis A. Connor. Nov. 4, 2004.

United States Court of Appeals Federal Circuit, Brief for Plaintiff-Appellant, Frank E. Scherkenbach, Sep. 2, 2005.

United States District Court for the District of Rhode Island 02-CV-343 (ECT). Memorandum in Support of Defendant Microsoft Corporation's Opposition to Arendi's Motion for a New Trial. Patricia A. Sullivan, Oct. 27, 2004.

United States Court of Appeals for the Federal Circuit. Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Brief of Plaintiffs-Appellants Arendi U.S.A., Inc. and Arend Holding Limited. Donald R. Dunner, Apr. 25, 2005.

United States Court of Appeals for the Federal Circuit. Appeal From the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Reply Brief for Defendant-Cross Appellant. Frank E. Scherkenbach, Oct. 3, 2005.

United States District Court for the District of Rhode Island, Order Denying Defendant Microsoft Corporation's Motion for Judgment as

US 7,496,854 B2

Page 3

- a Matter of Law That the '853 Patent is Invalid, C.A. No. 02-343T. Ernest C. Torres, Chief Judge, Nov. 30, 2004.
- United States District Court for the District of Rhode Island. C.A. 02-343T. Order Denying Plaintiffs' Motion for New Trial. Ernest C. Torres, Chief Judge, Nov. 30, 2004.
- Addressmate Automatic Envelope Addressing Program, User's Manual, 1991.
- Peter Brown, Unix Guide, 1995.
- N.D. Beitner, et al, Multimedia Support and Authoring in Microcosm: an extended model.
- Lee E. McMahon, SED—A Non-interactive Text Editor, Bell Laboratories, Aug. 15, 1978.
- SED(1), BSD Reference Manual Page.
- AddressMate for Windows, Version 2.0, Product Box and License Agreement.
- AddressMate Incorporates Correction Capability, The New York Law Publishing Company, Jan. 1996.
- Lawrence J. Magid, Addressing the Matter of Labels, Los Angeles Times, Sep. 23, 1996.
- Mar. 1996 Reviews, website available at www.techweb.com/winmag/library/1996/0396/03rvh002.htm.
- Important Note for New AddressMate Users.
- Important Tips for LabelWriter Owners Using AddressMate.
- Addressing the Issues, Jun. 24, 1993.
- AddressMate Advertisement "Whye Do it the Olde Way".
- Philip Robinson, The Envelope, Please: It's AddressMate, San Jose Mercury News, Sunday, Jul. 19, 1992.
- Steve Supkoff, AddressMate, PCM, Aug. 1992.
- L.R. Shannon, Addressing Envelopes, The New York Times, Tuesday, Jul. 14, 1992.
- Rob Schwabach, Addressing for Success, On Computers column, Mar. 16, 1992.
- INFOWORLD, p. 15, Mar. 16, 1992.
- Ken Hart, Simplify Envelope Printing with AddressMate, Computer Shopper.
- Colvin's beta-testing of Addressmate sparks developer's appreciation, Westview, Apr. 6, 1992.
- Press Release: AddressMate Software Automatically Addresses and bar Codes Envelopes to Save Time and money, Mar. 9, 1992.
- Gearoge M> Long, Letter to Mr. David Block, Nov. 11, 1994.
- Philip J. Damiano, Letter to Mr. David Block, Jan. 5, 1994.
- AddressMate for Windows, Advertisement.
- AddressMate Plus, Advertisement.
- Ed Kahn, Envelope Addressing Finally Simplified AddressMate Does Addressing and Much More, Microtimes, Nov. 27, 1995.
- Kirsten Bernthal, LabelWriter XL Plus, PC Catalog, Aug. 18, 1995.
- David Plotkin, Address for Success, Bay Area Computer Currents, p. 36-38, Mar. 21, 1995.
- Address Fixer for Microsoft Word and office, Product Box.
- AddressMate Plus, Product Box and License Agreement.
- Getting Results with Microsoft Office for Windows 95, Version 7.0, 1995.
- Getting Results with Microsoft Office 97, copyright 1995-1997.
- Microsoft Word, User's Guide, Verison 6.0, copyright 1993-1994.
- Apple Internet Address Detectors User's Manual, copyright 1997.
- CTAGS(1) Manual Page.
- Eve Wilson, Links and Structures in Hypertext Databases for Law, in Hypertext: Concepts, Systems and Applications, Proceedings of the First European Conference on Hypertext, Nov. 1990.
- Matt Bernstein, An Apprentice that Discovers Hypertext Links.
- Contextual Menu Manager/Apple Data Detectors.
- Eve Wilson, Guiding Lawyers: Mapping Law into Hypertext, Artificial Intelligence Review 6, pp. 161-189, 1992.
- P.J. Brown et al, A Help System Based on UNIX Man Pages.
- Charles H. Franke III et al, Authoring a Hypertext Unix Help Manual, 1995.
- P.J. Brown, Guide User Manual, 1985, sixteenth impression, Apr. 1995.
- E. Wilson, Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System, Literary and Linguistic Computing, vol. 5, No. 2, 1990.
- John Robertson, et al, The Hypermedia Authoring Research Toolkit, ECHT 194 proceedings, pp. 177-185.
- Mike Langberg, Apple Breaks New Ground by Displaying What on its drawing board, Aug. 7, 1996.
- What is Wiki and Wiki History webpages, available at wiki.org/wiki.cgi?WhatIsWiki and www.c2.org/cgi/wiki?WikiHistory.
- E. Wilson, Integrated Information Retrieval for Law in a Hypertext Environment, Annual ACM Conference on Research and Development in information Retrieval, 1988.
- Multimedia Hyperlinks Automatically Created for Reference Documents, Research Disclosure, Jun. 1993.
- Wiki Wiki Origin.
- Joy-Lyn Blak, WikiWikiWeb, Computer World, Jan. 29, 2001.
- Microsoft's Supplemental Responses to Arendi's Interrogatories, in *Arendi U.S.A. et al v. Microsoft Corporation*, Civil Action 02-CV-343 (ECT) from United States District Court for the District of Rhode Island.
- Apple Introduces Internet Address Detectors, Press Release, Sep. 7, 1997.
- Cara Cunningham, Apple Kicks Off Macworld with talk of revival, new software demos, InfoWorld Electric, Aug. 7, 1996.
- James Staten, Apple Looks to the Future, MacWeek, Aug. 7, 1996.
- Mark Simmons, Striking a Key Note, Mac Addict Online, Aug. 8, 1996.
- Jim Miller, email regarding Apple Data Detectors, Jan. 8, 1997.
- Apple Data Detectors web page, Jan. 6, 1997.
- Apple Data Detectors—Now Shipping web page, Jan. 6, 1997.
- The Apple Data Detectors FAQ, Jan. 6, 1997.
- Apple Data Detector Webpages, available on web.archive.org/web/20020601164217/www.apple.com/applescript/data_detectors.
- Apple Data Detectors 1.0.2 Read Me.
- Developer's Guide to Apple Data Detectors, Dec. 1, 1997.
- AppleScript Editors, Utilities & Environments, available at www.applescriptsourcebook.com/links/applescripteditors.html, dated Jan. 11, 2004.
- eMailman Internet Address Detectors.
- Steve Tannehill, News from Jul. 1997.
- Contol-Click! The Archive.
- Contextual Menus: One of System 8's Greatest Features, in ApplePress.
- Contextual Menu Manager/Apple Data Detectors, available at web.archive.org/web/20020803063750/www.macemail.com/emailer/CEMH/contextual.shtml.
- Trygve's CMM Plug-Ins Homer, available at web.archive.org/web/19980130053511/www.bombaydigital.com/cmms.
- ADD Depot, available from web.archive.org/web/20000819091818/http://homepage.mac.com/mathewmiller/add.
- Press Release: Apple Introduces Internet Address Detectors, Sep. 8, 1997.
- MacWEEK Report, Aug. 8, 1996.
- Mike Langber, Show of Potential Apple Breaks New Ground By Displaying What's on Its Drawing Board 'Innovation is at the heart of what we do', in San Jose Mercury News, Aug. 7, 1996, p. 1C.
- Apple Introduces Internet Address Detectors, Newsbytes, Sep. 29, 1997.
- Greg Williams, Strategy Mosaic: Understanding Apple's Dual OS Strategy.
- Taking [control] of your Mac with System 8, The MacAuthority, Jan. 1998.
- Apple Data Detectors 1.0.2, TidBITS Updates, Mar. 8, 1998.
- Apple Data Detectors 1.0.2, TidBITS #419, Mar. 9, 1998.
- Tonya Engst, More Context on Contextual menus, TidBITS #399, Sep. 29, 1997.
- Tonya Engst, Of Mice and Menus, TidBITS #398, Sep. 22, 1997.
- Charles Whaley, Will this be enough to kick-start Apple?, Computing Canada, Aug. 4, 1997.
- MacOS8.com—Mac OS 8 Indepth.
- A Farewell to the Apple Advanced Technology Group, SIGCHI, vol. 30, No. 2, Apr. 1998.
- James R. Miller and Thomas Bonura, From Documents to Objects, in SIGCHI, vol. 30, No. 2, Apr. 1998.
- Thomas Bonura and James R. Miller, Drop Zones, in SIGCHI, vol. 30, No. 2, Apr. 1998.
- Bonnie A. Nardi, et al., Collaborative, Programmable Intelligent Agents, Communicaitons of the ACM, vol. 41, No. 3, Mar. 1998.

US 7,496,854 B2

Page 4

Novell GroupWise User's Guide for Windows 16-BIT, Version 5.2, 1993, MS 125993, Novell, Inc., Orem, Utah.

Novell GroupWise Webaccess User's Guide, 1998, MS 126785, Novell, Inc., Orem, Utah.

Novell GroupWise User's Guide For Windows 32-BIT, 1998, MS 126463, Novell, Inc., Orem, Utah.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from 2:00pm Sep. 13, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 14, 2004, Providence RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 15, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 16, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 17, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 20, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 21, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 22, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 23, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 27, 2004, Providence, RI.

Arendi USA, Inc., et al. vs. Microsoft Corporation, et al. C.A. No. 02-343T Court Transcript from Sep. 28, 2004, Providence, RI.

* cited by examiner

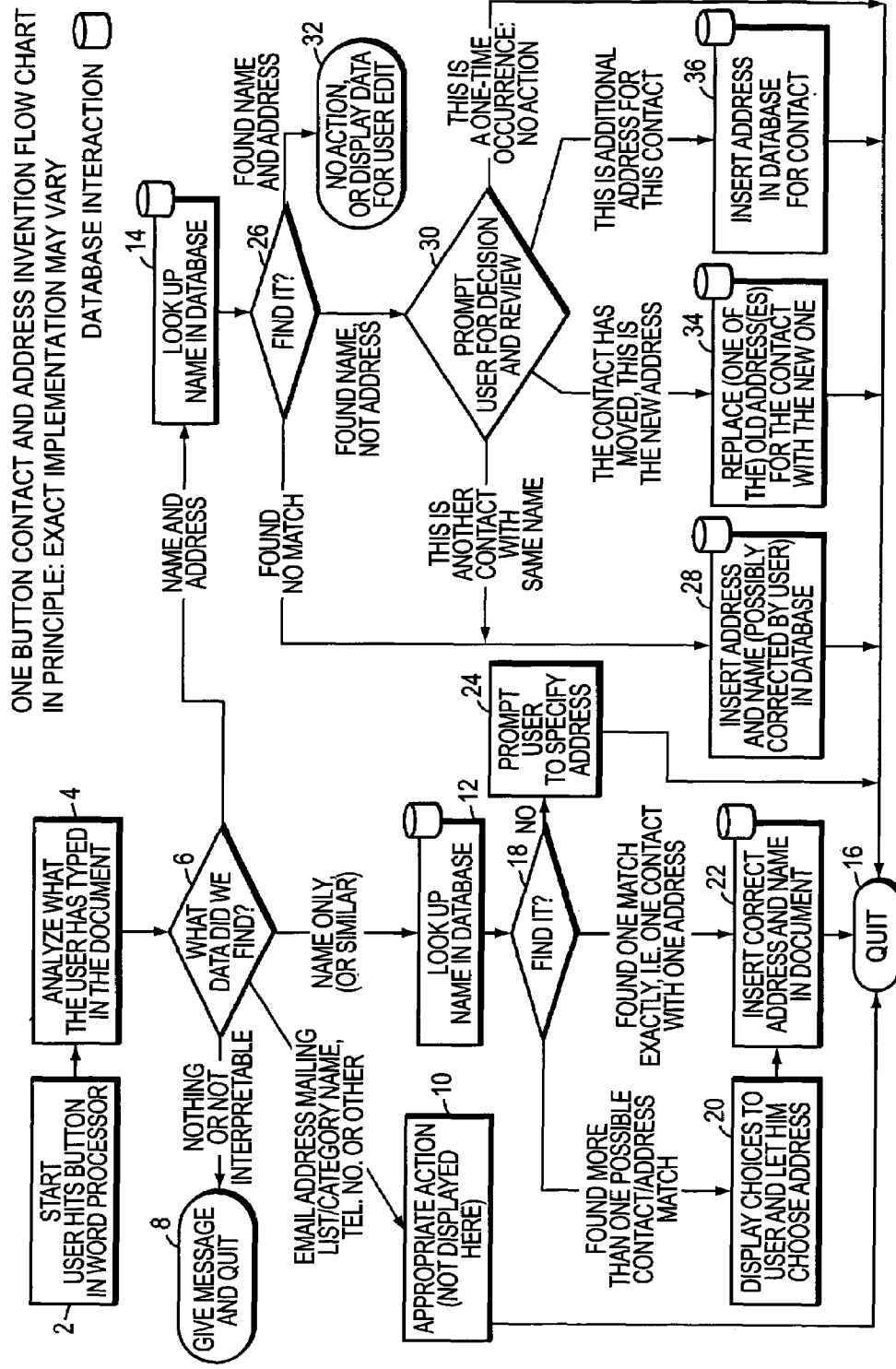


FIG. 1

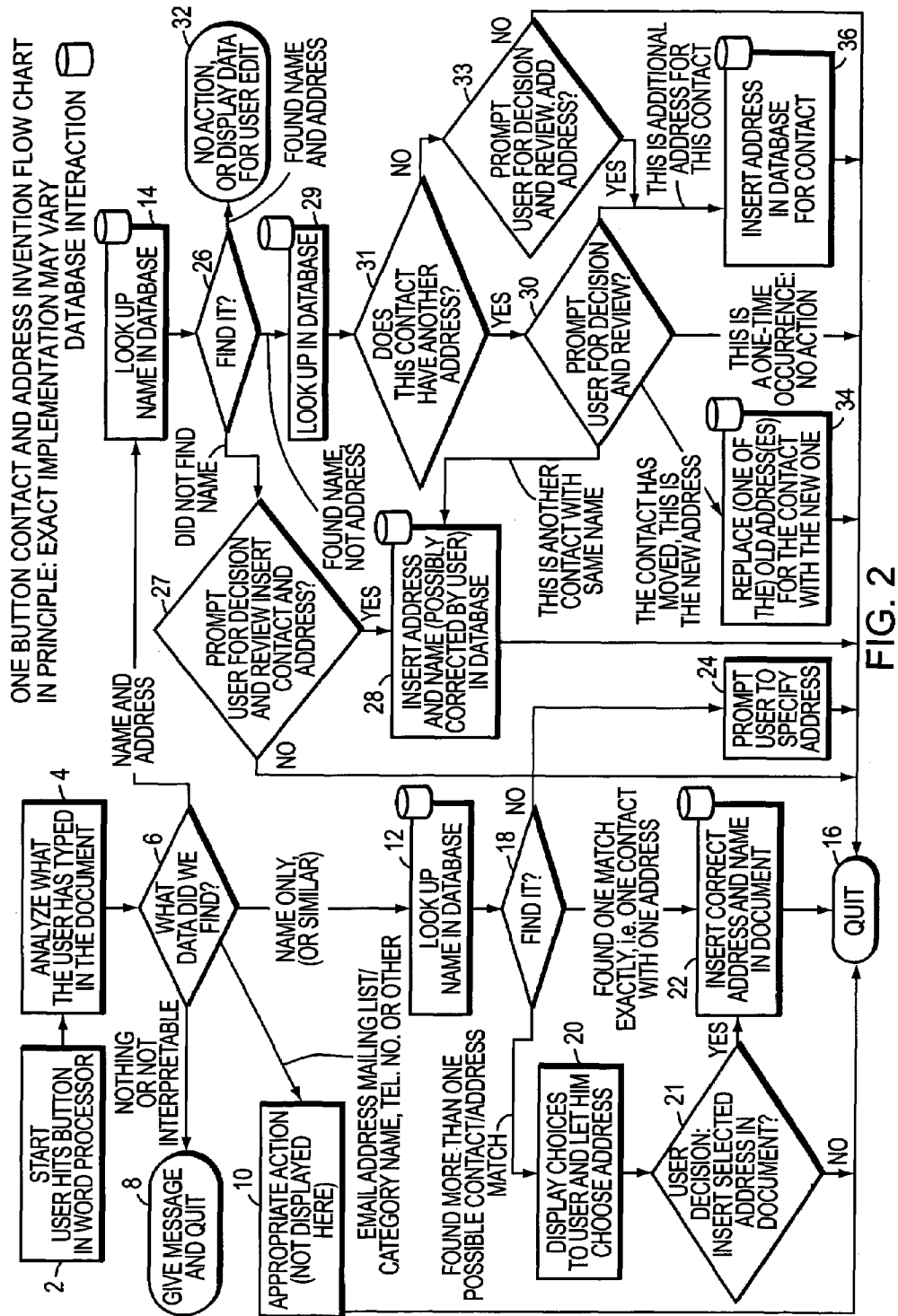


FIG. 2

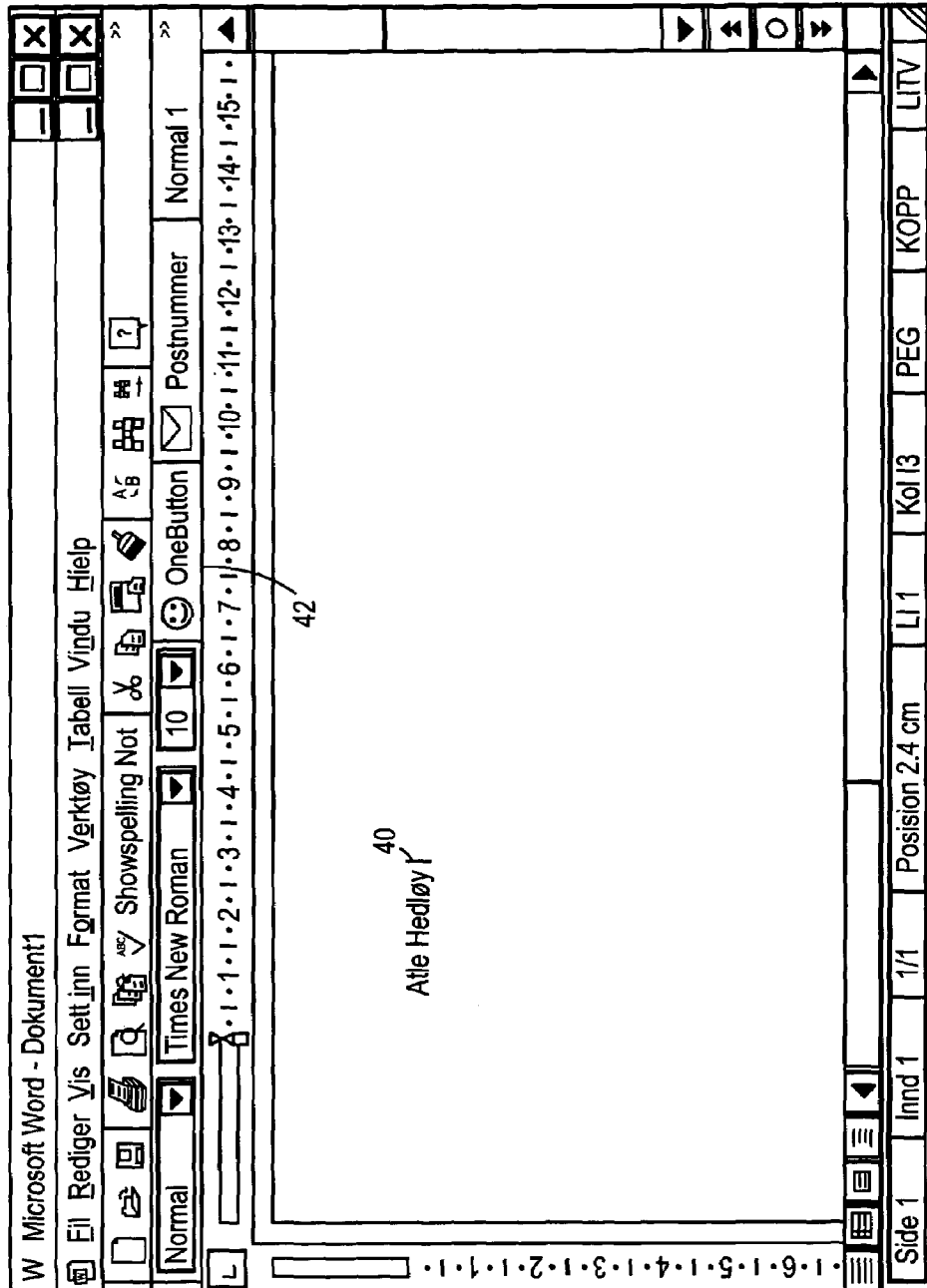


FIG. 3

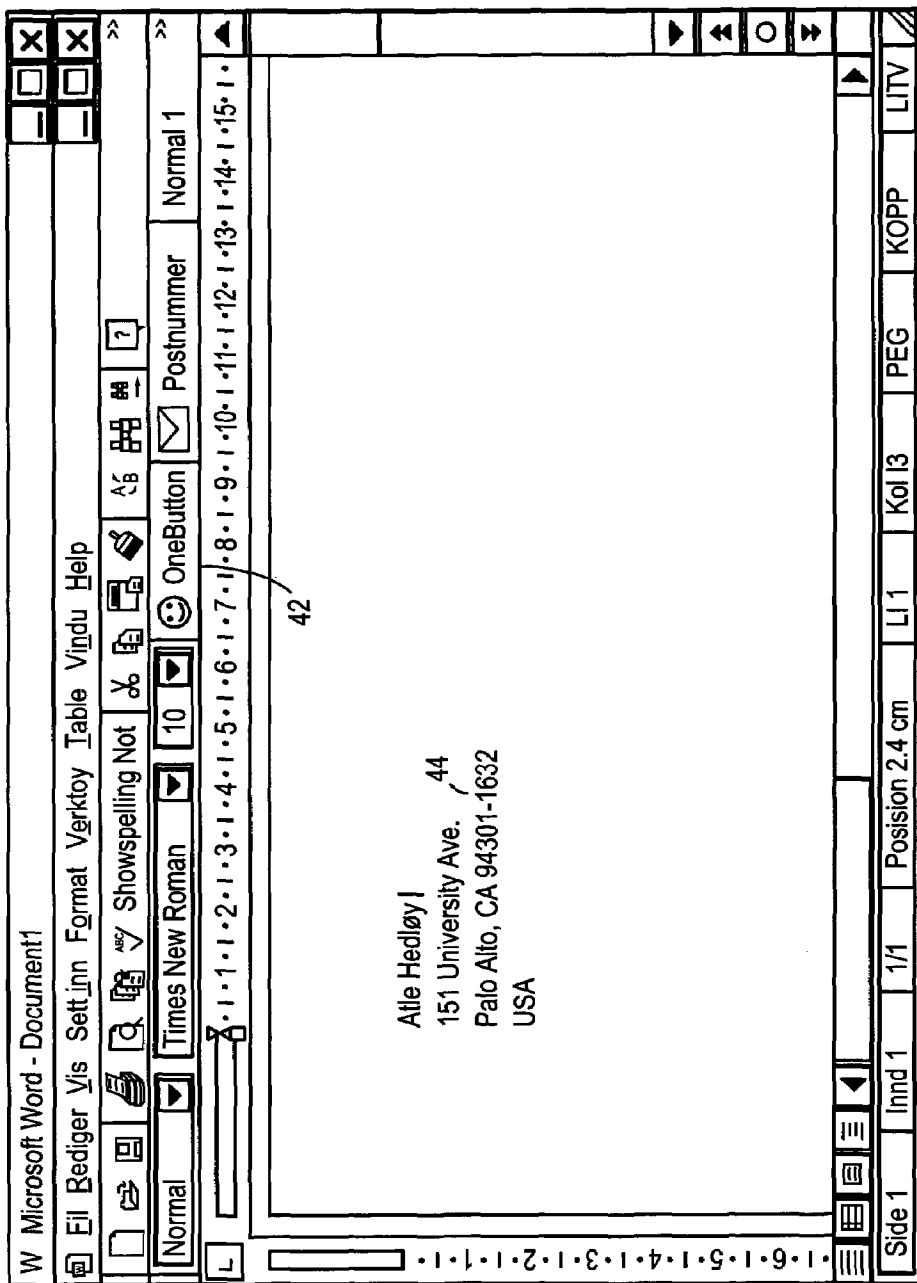


FIG. 4

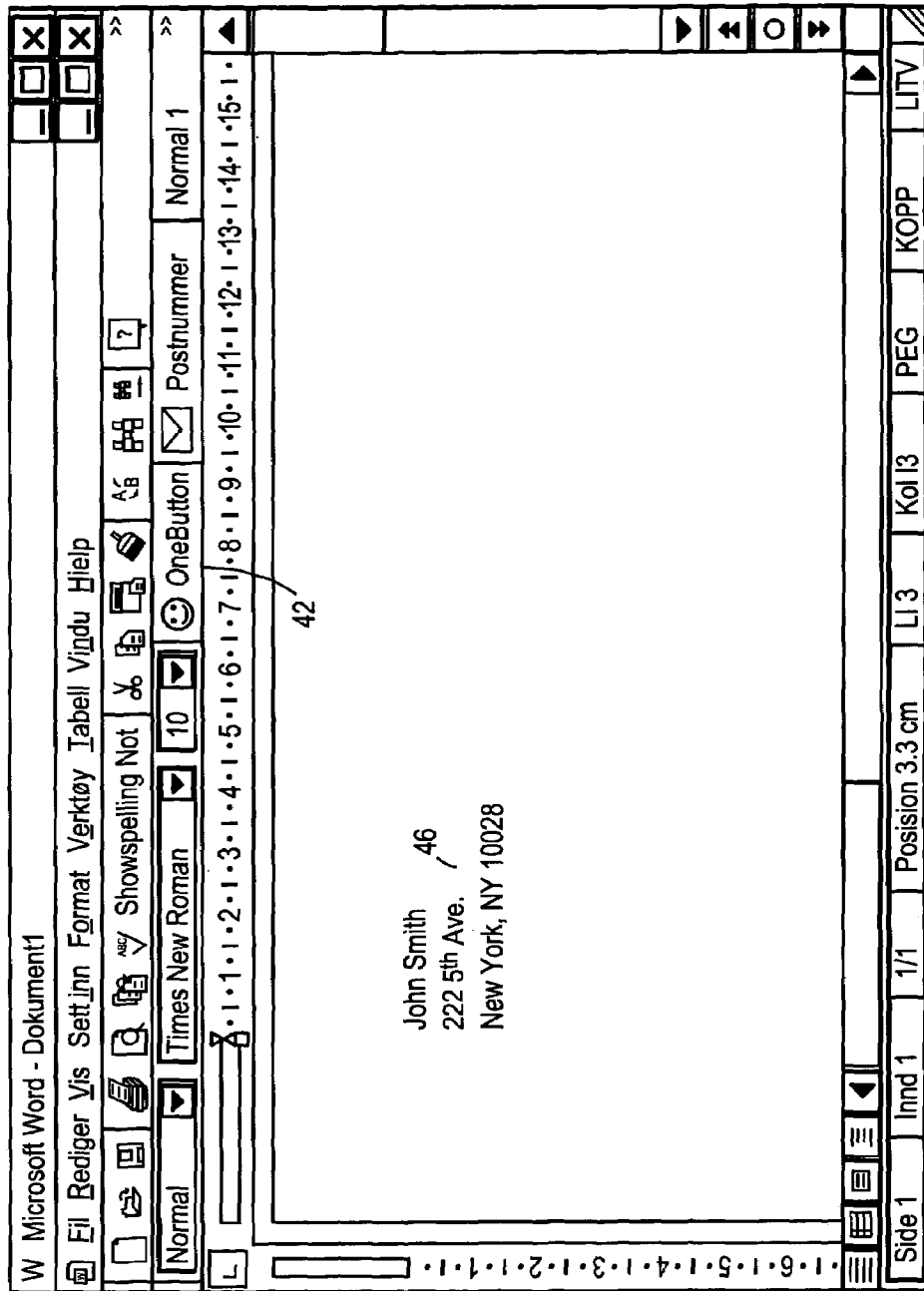


FIG. 5

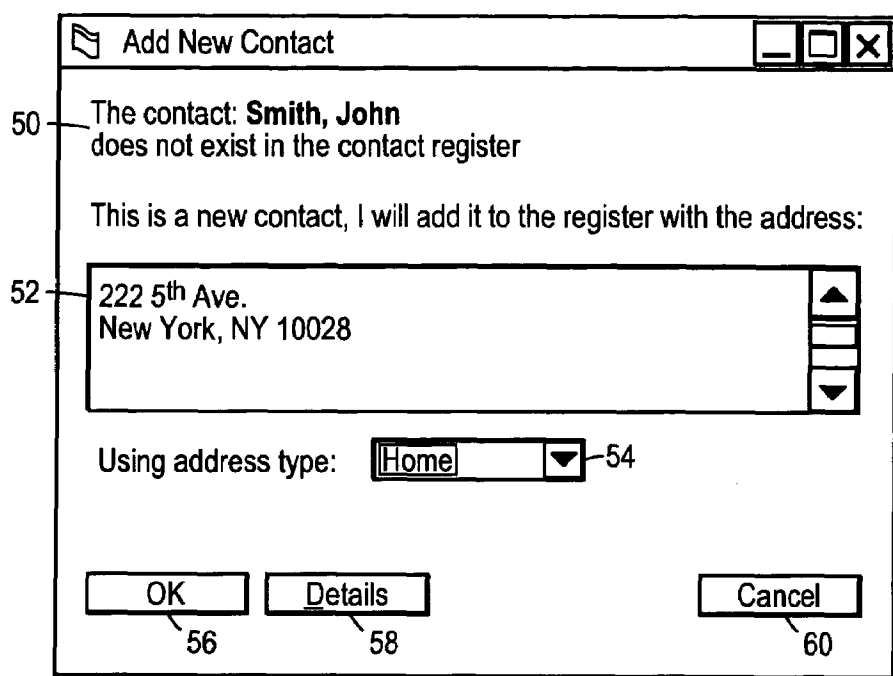


FIG. 6

The screenshot shows a software window titled "Arendi OneButton Contact Register". The window contains a form with the following fields and controls:

- Name:** A dropdown menu (54) with "Home" selected. Below it are input fields for Title, First (John), Middle, Last (Smith), Suffix, and Company.
- Address type:** A dropdown menu (54) with "Home" selected.
- Street:** Input field containing "222 5th Ave."
- City:** Input field containing "New York"
- State/Province:** Input field containing "NY"
- ZIP/Postal:** Input field containing "10028"
- Country:** A dropdown menu.
- Buttons:** "Add and Choose" (64), "Options..." (66), and "Cancel" (60).
- Footer:** "Dette er en test"

FIG. 7

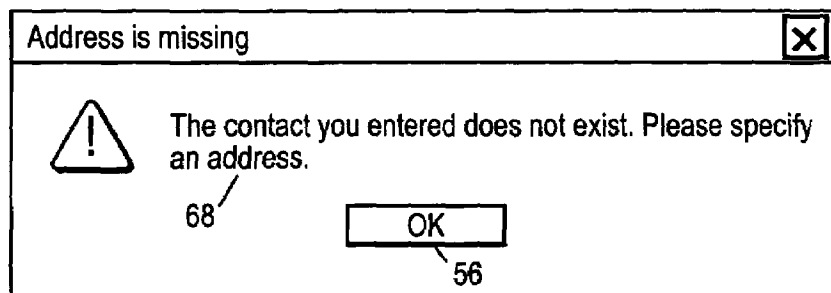


FIG. 8

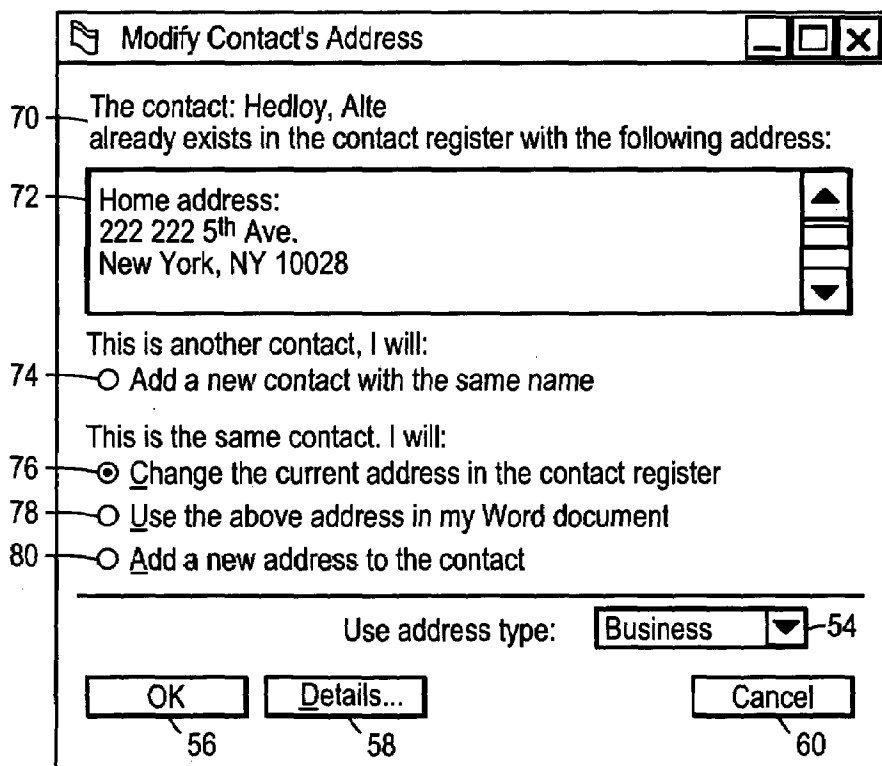


FIG. 9

The screenshot shows a window titled "Arendi OneButton Contact Register" with a close button (X) in the top right corner. Below the title bar, there is a text label "Existing addresses with the same name" followed by a list of two entries:

Name	Address type	Address	Zip	City	Country
[1]Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Below the list, there are two large text boxes. The left box is labeled "Name" and contains "Atle Hedløy". The right box is labeled "Address" and contains "113 Terrasse street", "New York, NY 12191-4292", and "United States of America".

Below the "Name" box is a button labeled "Options..." with reference numeral 66. Below the "Address" box are two buttons: "Choose" (86) and "Full details..." (88). Below the "Options..." button is a button labeled "Dette er en test".

At the bottom right of the window is a "Cancel" button (60). At the bottom center, there is a button labeled "More >>>" (90).

FIG. 10

Arendi OneButton Contact Register

Existing addresses with the same name

Name	Address type	Address	Zip	City	Country
[1]Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Name: Atle Hedløy
Address: 113 Terrasse street, New York, NY 12191-4292, United States of America

Buttons: Choose, Full details..., More >>>

Name: [Dropdown] Title: [Dropdown]
 First: Atle Middle: [Dropdown]
 Last: Hedløy Suffix: [Dropdown] Company: [Dropdown]

Address type: Home
 Street: 151 University Ave.
 City: Palo Alto
 State/Province: CA
 ZIP/Postal: 94301-1632
 Country: USA

Buttons: Add and Choose, Options..., Cancel

Footer: Dette er en test

FIG. 11

Atle Hedløy - Contact

File Edit View Insert Format Tools Contact Help

Save and Close Print Copy Paste Find Home Back Forward Help

General Details Journal All Fields

Full Name... [Atle Hedløy]

Company:

Address... 113 Terrasse street
New York, NY 12191-4292
United States of America

Business

Job title:

File as: Hedløy, Atle

Phone: Business Home Business Fax Mobile

This is the mailing address

E-mail

Web page:

Categories...

Private

94 96 98 100 102 104

FIG. 12

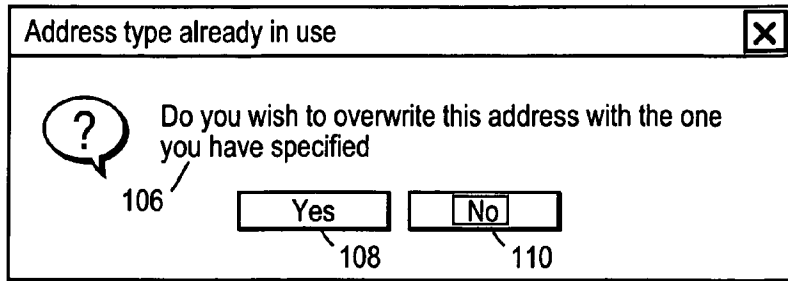


FIG. 13

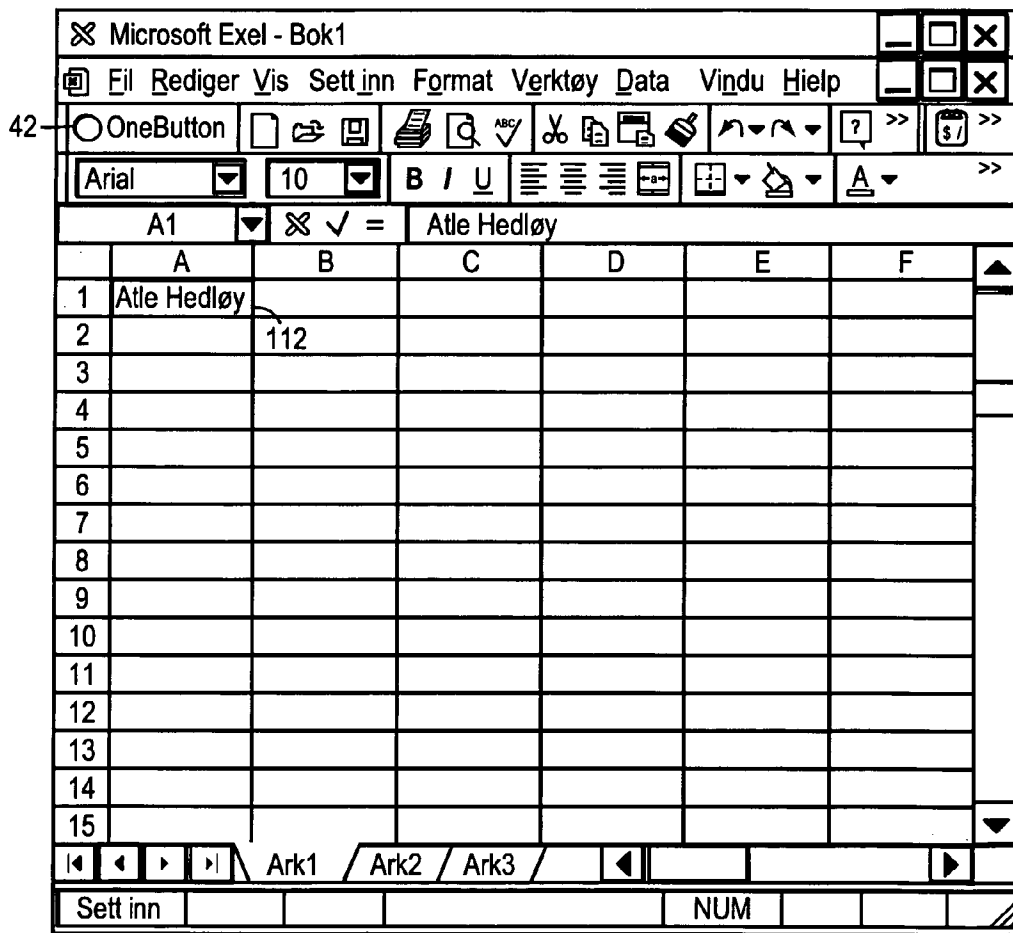


FIG. 14

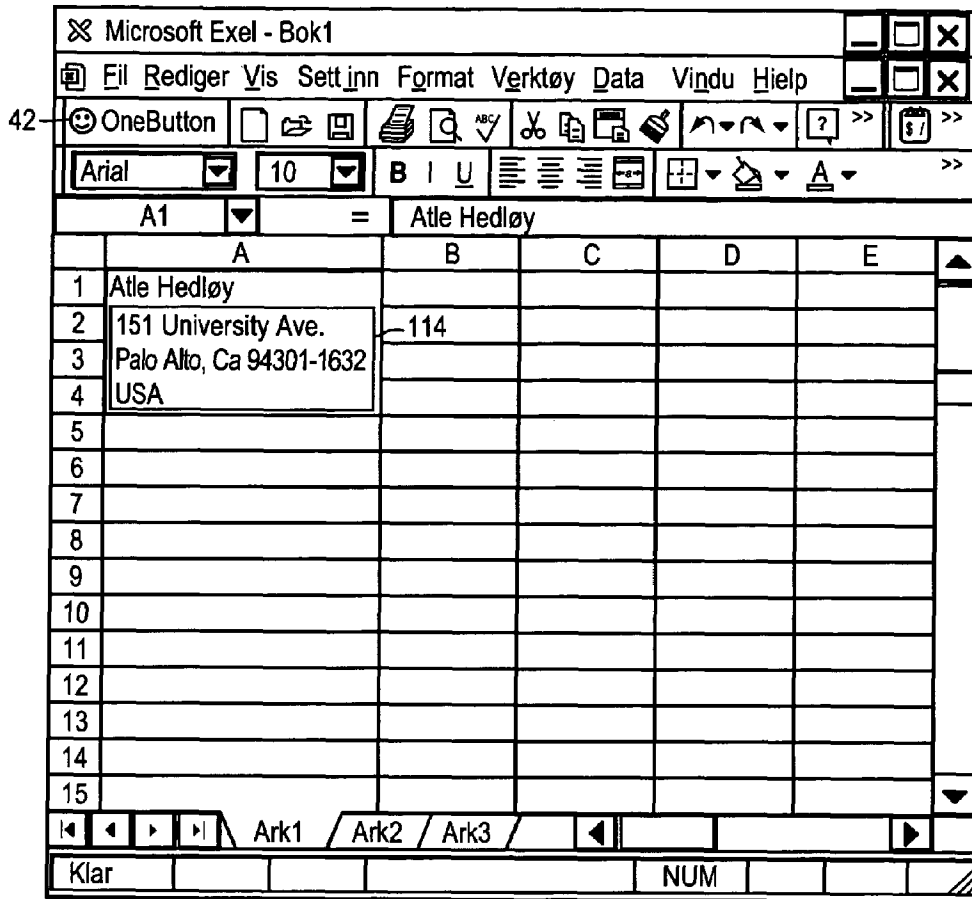
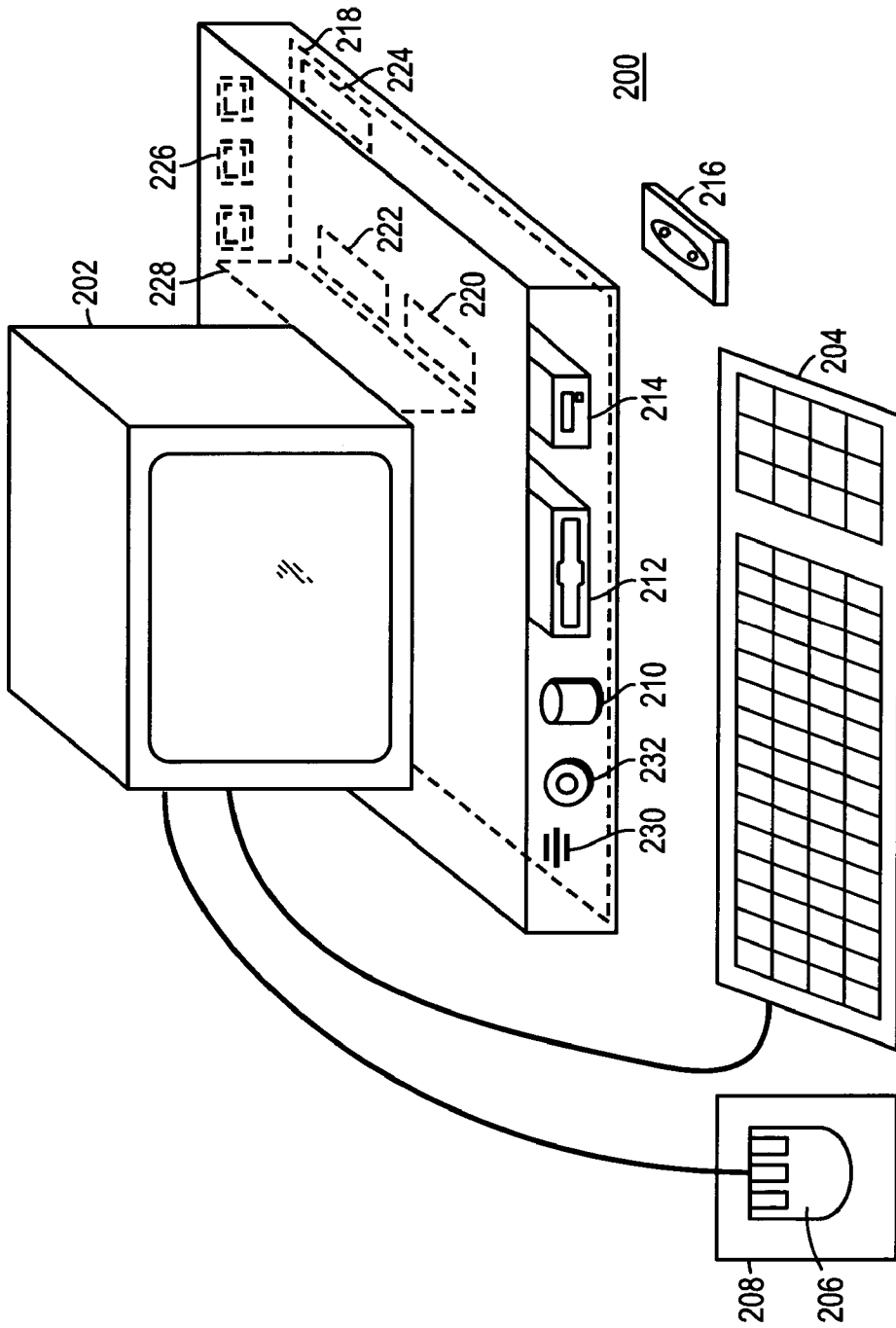


FIG. 15



US 7,496,854 B2

1

**METHOD, SYSTEM AND COMPUTER
READABLE MEDIUM FOR ADDRESSING
HANDLING FROM A COMPUTER PROGRAM**

CROSS-REFERENCE TO RELATED
APPLICATIONS

The present application is a continuation of commonly assigned copending U.S. patent application Ser. No. 09/189,626, which was filed on Nov. 10, 1998 now U.S. Pat. No. 6,323,853, by Hedloy for a METHOD, SYSTEM and COMPUTER READABLE MEDIUM FOR ADDRESSING HANDLING FROM A COMPUTER PROGRAM and is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

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. Background Information

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 ACCESST™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™, etc.

However, the information in the database must constantly be updated by the user. This requires the user to learn how to use and have access to the database. In this case, a change in the information, such as change in address or a name, etc., requires the user of the word processor to implement this change in the database, or alternatively, the change is made to the database centrally by a database administrator.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device provided in the computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address

2

handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device, such as a touch screen, keyboard button, icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device such as a touch screen, keyboard button icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source, such as a database program, contact management program, etc.

The above and other objects are achieved according to the present invention by providing a novel method, system and computer readable medium for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

The present invention also includes a computer readable medium storing program instructions by which the method of the invention can be performed when the stored program instructions are appropriately loaded into a computer, and a system for implementing the method of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

FIG. 1 is a flow chart illustrating a method for address handling within a computer program, according to an exemplary embodiment of the present invention;

FIG. 2 is a flow chart illustrating a method for address handling within a computer program, according to another exemplary embodiment of the present invention;

FIG. 3 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 4 is a screen shot illustrating a retrieved address in a word processor, according to an exemplary embodiment of the present invention;

FIG. 5 is a screen shot illustrating the inputting of a name and address to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 6 is a screen shot illustrating an add new contact message window, according to an exemplary embodiment of the present invention;

FIG. 7 is a screen shot illustrating a contact register message window, according to an exemplary embodiment of the present invention;

US 7,496,854 B2

3

FIG. 8 is a screen shot illustrating an address missing message window, according to an exemplary embodiment of the present invention;

FIG. 9 is a screen shot illustrating a modify contact's address message window, according to an exemplary embodiment of the present invention;

FIG. 10 is a screen shot illustrating a select a contact address register message window, according to an exemplary embodiment of the present invention;

FIG. 11 is a screen shot illustrating a more detailed mode of registering an additional address for the contact register of FIG. 9, according to an exemplary embodiment of the present invention;

FIG. 12 is a screen shot illustrating a contact management program window in a full detailed mode, according to an exemplary embodiment of the present invention;

FIG. 13 is a screen shot illustrating an address already in use message window, according to an exemplary embodiment of the present invention;

FIG. 14 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a spreadsheet, according to an exemplary embodiment of the present invention;

FIG. 15 is a screen shot illustrating a retrieved address in a spreadsheet, according to an exemplary embodiment of the present invention; and

FIG. 16 is a schematic illustration of a general purpose computer for performing the processes of the present invention, according to an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

In an embodiment of the present invention, single button addressing is achieved by providing an input device, such as a touch screen, keyboard, icon, menu, voice command device, etc. (hereinafter called "button"), in a computer program, such as a word processing program, spreadsheet program, etc. (hereinafter called "word processor"), for executing address handling therein.

Accordingly, 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, such as a touch screen button, keyboard button, icon, menu choice, voice command device, etc. A program then executes and retrieves the typed information from the document, and searches an information management source, such as a database, file, database program, contact management program, etc. (hereinafter called "database") to determine if the information, such as the name or part of the name typed and searched by the program exists in the database. If the program does not find stored information, such as a name, corresponding to the name or part of the name typed, the user is asked by the program whether the information, such as the name that was not found, should be added to the database. In addition, the user may enter any other information besides the name, such as addresses, businesses, telephone numbers, fax numbers, e-mail address, etc., so that this other information can be stored in the database for later use.

If the program finds name(s) and address(es) corresponding to the part of the addressee's name typed, this additional information is automatically entered into the user's word processor, optionally with a confirmation from the user that this is the correct data. If the typed address information does

4

not correspond to data already stored in the database, after clicking on the button, the program, for example, lets the user decide: (1) if this is new data (e.g., a new address) for an existing contact; (2) if the stored data should be changed to what the user just typed; (3) if this is a new contact with the same name as the one already entered into the database; or (4) if the typed address is only to be used once, and therefore not to be stored in the database at all. If, later, for example, a name with several address stored in the database is recalled, all addresses for this contact will be displayed, so that the correct address can be selected by the user.

The program may be extended to also store and retrieve other information, such as telephone numbers, fax numbers, e-mail addresses, etc. Once the program recalls the telephone numbers, fax numbers, email addresses, etc., the user can command the program to send e-mails, faxes, etc. Similarly, if the user types in the name of a mailing list, the program create merge letters, group emails, etc.

Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views, and more particularly to FIGS. 1 and 2 thereof, there is illustrated flow charts of single button addressing, according to exemplary embodiments of the present invention.

In FIG. 1, after the user has inserted the address in the word processor, the user commands the button at step 2 and the program analyzes what the user has typed in the document at step 4. At step 6, the program decides what was found in the document and if the program found nothing in the document or what it found was un-interpretable the program goes to step 8 and outputs an appropriate message to the user and then quits at step 16. The program analyzes what the user has typed in the document at step 4, for example, by analyzing (i) paragraph/line separations/formatting, etc.; (ii) street, avenue, is drive, lane, boulevard, city, state, zip code, country designators and abbreviations, etc.; (iii) Mr., Mrs., Sir, Madam, Jr., Sr. designators and abbreviations, etc.; (iv) Inc., Ltd., P.C., L.L.C, designators and abbreviations, etc.; and (v) a database of common male/female names, etc.

If the program find an e-mail address mailing list/category name telephone number or other information, at step 10 an appropriate action is performed by the program and then the program execution quits at step 16. If the program only finds a name or initials, or the like, the program looks up the name in the database at step 12 and at step 18 the program determines what was found. If the program finds more than one possible contact/address match, at step 20 the program displays menu choices to the user to let him choose an appropriate answer. Then at step 22 the program inserts a correct address and name in the document and then at step 16 the program quits execution. If the program finds one match exactly, i.e., one contact with one address, the program inserts the correct address and name in the document at step 22 then quits and then quits execution at step 16. If the program does not find a name in the database, at step 24 the program prompts the user to specify an address and then quits execution at step 16. If the program at step 6 finds a name and an address, at step 14 the name is looked up in the database. Then, at step 26, if no match is found, at step 28 the program inserts an address and a name which are possibly corrected by the user into the database and then quits execution at step 16. If at step 26, the name and address is found, at step 32 the program either takes no action or displays the data for the user to edit. If at step 26, the name is found but not the address, the program prompts the user for a decision at step 30. If the user decides that this is another contact with a same name, the program goes to step 28. If the is user decides that this is a one

US 7,496,854 B2

5

time occurrence, no action is taken and the program quits at step 16. If the user decides that the contact has, for example, moved and that this is a new address, at step 34 one of the old addresses for the contact is replaced with the new one and the program with the new one and the program quits at step 16. If the user decides that this is an additional address for the contact, at step 36 the additional address is inserted into the database for that contact and execution quits at step 16.

The flowchart shown in FIG. 2 is similar to the flowchart in FIG. 1, except for some additional steps which will now be discussed. At step 6, if the program only finds a name or a similar name then the name is looked up in the database at step 12, then at step 18 if the program found more than one possible contact/address match, the program displays choices to the user to let him choose an address at step 20. Then at step 21 the user decides whether to insert the selected address into the document. If the user does not decide to select the address into the document the program quits execution at step 16. If the user decides to insert the selected address into the document the program inserts the address and name into the document at step 22 and then quits at step 16.

If the program finds a name and address in the database at step 6, then at step 14 the program looks up the name in the database and at step 26 the program determines what it has found. If the program does not find the name at step 26, at step 27 the program prompts the user for a decision and review and whether to insert the contact and address. If the user does not decide to insert the contact address, the program quits at step 16. If the user decides to insert the contact address, at step 28 the program inserts the address and name which may be possibly corrected by the user for program in the database then execution quits at step 16.

If at step 26 the program finds a name and not an address, then at step 29 the name is looked up in the database. Then at step 31 the program decides whether this contact has another address. If the contact does not have another address, at step 33 the program prompts the user for a decision and review and whether to add the address. If the user does not want to add the address at step 33, the program quits at step 16. If the user wants to add the address at step 33 because this is an additional address for the contact, at step 36 the address is inserted in the database for the contact and execution quits at step 16.

At step 30, if the user decides that this is another contact with the same name, then the program goes to step 28. If at step 30 the user decides that this is a one time occurrence, then the program quits at step 16. If at step 30, the user decides that the contact has, for example, moved, the program goes to step 34. If at step 30, the user decides that this is an additional address for the contact, at step 36 the program inserts the address in the database for the contact and then quits at step 16.

Various exemplary screen shots which are generated during execution of the program, according to the present invention, will now be described with reference to FIGS. 3-15 and examples 1-7 as follows.

Example 1

Retrieving an Existing Address from the Database

FIG. 3 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name 40. The user hits the button 42, for example, marked "OneButton" and the program according to the present invention retrieves the name 40 from the document, searches a

6

database for the name 40, and inserts the retrieved address 44 associated with the name 40 into the document as shown in, for example, FIG. 4.

The above example corresponds to steps 2, 4, 6, 12, 18, 22 and 16 in the flow charts of FIGS. 1 and 2.

Example 2

Adding a New Contact to the Database

FIG. 5 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name and address of a new contact 46. The user commands the button 42, for example, marked "OneButton," and the program according to the invention retrieves the new contact 46 from the document, searches a database for the name of the new contact 46 and generates a screen as shown in, for example, FIG. 6. This screen includes a message 50 informing the user that the new contact does not exist in the database, a message 52 including the address retrieved from the document, an address type selection 54, such as home, business, etc., and "OK," "Details," and "Cancel" buttons 56, 58, and 60, respectively.

At this point, the user can cancel the operation by commanding the Cancel button 60, ask the program to store data in the database and return the document by commanding the OK button 56, or check details before storing data into the database by commanding the Details button 58. If the user commands the Details button 58, as shown in, for example, FIG. 7, a message screen is provided so that the user can review and edit data 62 and the selection 54, store the data 62 and 54 in the database by commanding a "Add and Choose" button 64, see more options by commanding an "Options" button 66, or cancel the operation by commanding the Cancel button 60.

The above example corresponds to steps 2, 4, 6, 14, 26, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 27, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 26, 27, 28 and 16 in the flow chart of FIG. 2.

Example 3

Try to Retrieve Existing Address, But Contact is not in Database

FIG. 3 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name of a contact 40. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the name 40 from the document, searches a database for the name of the contact 40 and generates a screen as shown in, for example, FIG. 8. This screen includes a message 68 informing the user that the contact does not exist in the database and to specify an address, and "OK" buttons 56. At this point when the user commands the OK button 56, the user returns to the document so that the contact's address can be included as in Example 2 above.

The above example corresponds to steps 2, 4, 6, 12, 18, 24, and 16 in the flow charts of FIGS. 1 and 2.

Example 4

Adding a New Address for an Existing Contact (Short Version)

FIG. 4 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed

US 7,496,854 B2

7

a name and new address of an existing contact 44. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact 44 from the document, searches a database for the same name of the existing contact 44 and generates a screen as shown in, for example, FIG. 9. This screen includes a message 70 informing the user that the contact already exists in the database with an existing address, a message 72 including the existing address, add new contact with same name selection 74, change existing address selection 76, use existing address in document selection 78, add the new address contact selection 80, the address type selection 54, such as home, business, etc., and the "OK," "Details," and "Cancel" buttons 56, 58 and 60 respectively. At this point, the user may select one of the four options 74-80, and command the OK button 56 to execute the selected options. The user can also cancel the operation by commanding the Cancel button 60, or check details before storing data into the database by commanding the Details button 58.

The above example corresponds to steps 2, 4, 6, 14, 26, 28, 30, 34, 36, and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 29, 31, 30, 28, 34, 36, and 16 in the flow chart of FIG. 2.

Example 5

Selecting Between Several Possible Matching Addresses

FIG. 3 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and possibly address of at least one existing contact 40. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact 40 from the document, searches a database for the name of the existing contact 40 and generates a screen as shown in, for example, FIG. 10. This screen includes a message informing the user the name corresponds to several addresses and possible contacts which already exist in the database, with existing contacts and addresses for selection 82, a message 84 including the full name and address for the contact that the user selects in 82, the Options button 66, a "Choose" button 86, a "Full details" button 88, a "More>>>" button 90, and the Cancel button 60. The above screen indicates to the user that at least one contact with the same name exists, and that there are more than one addresses and/or contacts that match.

At this point, the user may command the Choose button 86 to use the selected address and return to the document, or the user may command the More>>> button 90 to view how the program interpreted what he user typed in the word processor, and possibly change this data, wherein the program generates an updated screen as shown in, for example, FIG. 11. The updated screen includes the data 62 which displays the name for example, FIG. 11. The updated screen includes the data 62 which displays the name typed in the word processor as interpreted by the program, address fields, and the fields for the address type selection 54, such as home, business, etc., which may be changed by the user before the program stores it in the database, the Add and Choose button 64, a "<<<Less" button 90 corresponding to the More>>> button 90 for returning to the screen of FIG. 10, and an "Add this address to the selected contact above" button 92. The user might then command the Add this address to the selected contact above button 92 and the result in the word processor is illustrated in FIG. 4. The user can also cancel the operations by commanding the Cancel button 60, or command the add choose button

8

64 to add this name and address as a new contact and address, or open the database before storing data into the database by commanding a "Full details" button 88 as will be later described.

The above example corresponds to steps 2, 4, 6, 12, 18, 20, 22, and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 12, 18, 20, 21, 22, and 16 in the flow chart of FIG. 2.

Example 6

Adding a New Address for an Existing Contact (Long Version)

FIG. 4 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and new address of an existing contact 44. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact 44 from the document, searches a database for the name of the existing contact 44 and generates a screen as shown in, for example, FIG. 9. As previously described the screen includes a message 70 informing the user that the contact already exists in the database with an existing address, and the user may command the Details button 58 to see the details of the new address for potentially modify the details before they are stored in the database and the program generates a screen as shown in, for example, FIG. 10. From this screen, the user may choose to use another address than the one he typed, and return to the document, or the user may command the "Full details" button 88 to enter a database program, such as OUTLOOK™, directly as shown in, for example, FIG. 12. In FIG. 12, the database program, such as OUTLOOK™, may include portions 94-104 for allowing the user to modify various pieces of data before they are stored in the database.

Alternatively, in the screen shown in FIG. 10, the user may command the More>>> button 90 at which time the program generates the screen as shown in, for example, FIG. 11 and as previously described. In this screen, the user might then command the Add this address to the selected contact above button 92. If the address typed is already in use, the program generates a screen including a message 106, and "Yes" and "No" buttons, 108 and 110, respectively, as shown in, for example, FIG. 13. If the user hits the Yes button 108 the program overwrites the contact address with the address specified by the user (e.g., if the contact has moved) and the result in the word processor is shown in, for example FIG. 4.

The above example corresponds to steps 2, 4, 6, 12, 14, 26, 28, 30, 34, 36, and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 12, 14, 26, 29, 31, 30, 28, 34, 36, and 16 in the flow chart of FIG. 2.

Example 7

Spreadsheet Application

FIG. 14 illustrates a starting point in word processor document, such as an EXCEL spreadsheet, wherein the user has typed a name 112. The user hits the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the name 112 from the spreadsheet, searches a database for the name 112, and inserts the retrieved address 114 into the spreadsheet as shown in, for example, FIG. 15. Accordingly, the examples 1-6 apply not only to word processor documents, such as WORD™ documents, etc., but to other word processor documents, and spreadsheets, such as EXCEL™ spreadsheets, etc.

US 7,496,854 B2

9

The above example corresponds to steps 2, 4, 12, 18, 22, and 16 in the flow of charts FIGS. 1 and 2.

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.

As stated above, the system includes at least one computer readable medium or alternatively, the computer readable medium may be accessed through various paths, such as networks, internet, drives, etc. Examples of computer readable media are compact discs, hard disks, floppy disks, tape, magneto-optical disks, PROMs (EPROM, EEPROM, Flash, EPROM), DRAM, SRAM, SDRAM, etc. Stored on any one or on a combination of computer readable media, the present invention includes software for controlling both the hardware of the computer 200 and for enabling the computer 200 to interact with a human user. Such software may include, but is not limited to, device drivers, operating systems and user applications, such as development tools. Such computer readable media further includes the computer program product of the present invention for performing any of the processes according to the present invention, described above (see, e.g., FIGS. 1-15). The computer code devices of the present invention can be any interpreted or executable code mechanism, including but not limited to scripts, interpreters, dynamic link libraries, Java classes, and complete executable programs, etc.

The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

Address handling, according to this invention, is a significant simplification relative to existing methods, and requires little or no training on the part of a user, as correct addresses are retrieved with a minimal number of user commands, "clicks," keystrokes, etc. In addition, a program according to the present invention, can be programmed and created in most existing programming languages and be connected to most modern word processors. Therefore, according to the present invention, the process of creating and updating records in an address database is significantly simplified, since this may now be performed directly from the word processor.

Although the present invention is defined in terms of word processing documents, such as WORD™ documents and EXCEL™ spreadsheets, the present invention is applicable to all types of word processing documents such as NOTE-

10

PAD™, WORDPAD™, WORDPERFECT™, QUATRO-PRO™, AMIPRO™, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of information management or is database programs, such as OUTLOOK™, etc., the present invention is applicable to all types of information management or database programs such as ACCESS™, ORACLE™, DBASE™, RBASE™, CARD-FILE™, including "flat files," etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of providing an input device, such as a button 42 in a word processor for address handling therein, the present invention may be practiced with all types of input devices, such as touch screen, keyboard button, icon, menu, voice command device, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving information from a document before searching a database, the user may select the information in the document to be searched by the program in the database (e.g., by highlighting, selecting, italicizing, underlining, etc.), as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving a name or portion thereof from a document before searching a database the program may retrieve an address or portion thereof from the document before searching the database and insert, correct, complete, etc., the retrieved address based on the information found in the database corresponding to the retrieved address or portion thereof, as will be readily apparent to those skilled in the art.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of appended claims, the invention may be practiced otherwise than as specifically described herein.

This application claims priority and contains subject matter related to Norwegian patent application No. 984066 filed on Sep. 3, 1998, the entire contents of which are hereby incorporated by reference.

What is claimed is:

1. A method for information handling within a document created using a first application program comprising the steps of:

entering a first information in the first application program; marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

2. The method of claim 1 wherein the user selection further comprises an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

3. The method of claim 1, wherein the step of inserting the second information into the document further comprises the steps of:

initializing the second application program; searching, using the second application program, for the second information associated with the first information; and retrieving the second information.

4. The method of claim 3, wherein when the second application program includes second information associated with the first information, performing the further step of displaying the second information.

US 7,496,854 B2

11

5. The method of claim 4, further comprising the step of completing at least one of the first and second information in the document.

6. The method of claim 1, wherein the first information comprises a name.

7. A computer readable medium, including program instructions related to information handling within a document created using a first application program and for performing the steps of:

entering a first information in the first application program; marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

8. The computer readable medium of claim 7, wherein the user selection further comprises an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

9. The computer readable medium of claim 7, wherein the step of inserting the second information into the document further comprises the steps of:

initializing the second application program;

searching, using the second application program, for the second information associated with the first information; and

retrieving the second information.

10. The computer readable medium of claim 9, wherein when the second application program includes second information associated with the first information, performing the further step of displaying the second information.

11. The computer readable medium of claim 10, further comprising the step of completing at least one of the first and second information in the document.

12. The computer readable medium of claim 7, wherein the first information comprises a name.

13. A computer system related to information handling within a document created using a first application program, comprising:

means for entering a first information in the first application program;

means for marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

means for responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

14. The computer system of claim 13, wherein the means for the user selection further comprises:

means for an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

15. The computer system of claim 13, wherein the means for inserting the second information into the document further comprises:

means for initializing the second application program;

means for searching, using the second application program, for the second information associated with the first information; and

means for retrieving the second information.

16. The computer system of claim 15, wherein when the second application program includes second information

12

associated with the first information, and comprising means for performing the further step of displaying the second information.

17. The computer system of claim 16, further comprising means for completing at least one of the first and second information in the document.

18. The computer system of claim 13, wherein the first information comprises a name.

19. A method for information handling within a document created by a first application program comprising the steps of: entering a first information in the first application program; marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

responding to a user selection by performing an operation related to a second information, the second information associated with the first information from the second application program.

20. The method of claim 19 wherein the operation comprises displaying the second information.

21. The method of claim 19, wherein the first information is a name, and the operation performed is selected from a group consisting of an electronic mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

22. The method of claim 19, wherein the operation performed is entering additional data into a database.

23. The method of claim 22, wherein the additional data is entered by a user.

24. The method of claim 22, wherein the additional data is located within the document.

25. A computer readable medium, including program instructions related to information handling within a document created by a first application program and for performing the steps of:

entering a first information in the first application program; marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

responding to a user selection by performing an operation related to a second information, the second information associated with the first information from the second application program.

26. The computer readable medium of claim 25 wherein the operation comprises displaying the second information.

27. The computer readable medium of claim 25, wherein the first information is a name, and the operation performed is selected from a group consisting of an electronic mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

28. The computer readable medium of claim 25, wherein the operation performed is entering additional data into a database.

29. The computer readable medium of claim 28, wherein the additional data is entered by a user.

30. The computer readable medium of claim 28, wherein the additional data is located within the document.

31. A computer system related to information handling within a document created by a first application program, comprising:

means for entering a first information in the first application program;

means for marking without user intervention the first information to alert the user that the first information can be utilized in a second application program; and

means for responding to a user selection by performing an operation related to a second information, the second

US 7,496,854 B2

13

information associated with the first information from the second application program.

32. The computer system of claim 31 wherein the operation comprises displaying the second information.

33. The computer system of claim 31, wherein the first information is a name, and the operation performed is selected from a group consisting of an electronic mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

34. The computer system of claim 31, wherein the operation performed is entering additional data into a database.

35. The computer system of claim 34, wherein the additional data is entered by a user.

36. A method for information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program, the method comprising the steps of:

identifying without user intervention or designation the first information; and

responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

37. The method of claim 36, wherein the user selection further comprises an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

38. The method of claim 36, wherein the step of inserting the second information into the document further comprises the steps of:

initializing the second application program;

searching, using the second application program, for the second information associated with the first information; and

retrieving the second information.

39. The method of claim 38 wherein the step of inserting the second information into the document further comprises adding the second information to the first information in the document.

40. The method of claim 38, wherein when the second application program includes second information associated with the first information, performing the further step of displaying the second information.

41. The method of claim 38, further comprising the step of completing at least one of the first and second information in the document.

42. The method of claim 36, wherein the first information comprises a name.

43. A computer readable medium, including program instructions related to information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program, and for performing the steps of:

identifying without user intervention or designation the first information; and

responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

44. The computer readable medium of claim 43, wherein the user selection further comprises an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

14

45. The computer readable medium of claim 43, wherein the step of inserting the second information into the document further comprises the steps of:

initializing the second application program;

searching, using the second application program, for the second information associated with the first information; and

retrieving the second information.

46. The method of claim 45 wherein the step of inserting the second information into the document further comprises adding the second information to the first information in the document.

47. The computer readable medium of claim 46, wherein when the second application program includes second information associated with the first information, performing the further step of displaying the second information.

48. The computer readable medium of claim 46, further comprising the step of completing at least one of the first and second information in the document.

49. The computer readable medium of claim 43, wherein the first information comprises a name.

50. A computer system related to information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program, comprising:

means for identifying without user intervention or designation the first information; and

means for responding to a user selection by inserting a second information into the document, the second information associated with the first information from a second application program.

51. The computer system of claim 50, wherein the means for user selection further comprises means for an activation of a device selected from a group consisting of a touch screen, a keyboard button, a screen button, an icon, a menu, and a voice command device.

52. The computer system of claim 50, wherein the means for inserting the second information into the document further comprises:

means for initializing the second application program;

means for searching, using the second application program, for the second information associated with the first information; and

means for retrieving the second information.

53. The computer system of claim 50 wherein the means for inserting the second information into the document further comprises means for adding the second information to the first information in the document.

54. The computer system of claim 50, wherein when the second application program includes second information associated with the first information, and comprising means for performing the further step of displaying the second information.

55. The computer system of claim 54, further comprising means for completing at least one of the first and second information in the document.

56. The computer system of claim 50, wherein the first information comprises a name.

57. A method for information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program the method comprising the steps of:

identifying without user intervention or designation the first information; and

US 7,496,854 B2

15

responding to a user selection by performing an operation related to a second information, the second information associated with the first information from the second application program.

58. The method of claim 57 wherein the operation comprises displaying the second information.

59. The method of claim 57, wherein the first information is a name, and the operation performed is selected from a group consisting of generating an electronic mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

60. The method of claim 57, wherein the operation performed is entering additional data into a database.

61. The method of claim 60, wherein the additional data is entered by a user.

62. The method of claim 60, wherein the additional data is located within the document.

63. The method of claim 57 wherein the user selection comprises an activation of a menu.

64. The method of claim 63, wherein the step of performing the operation further comprises the steps of:

initializing the second application program;
searching, using the second application program, for the second information associated with the first information; and
retrieving the second information.

65. The method of claim 64, wherein when the second information associated with first information exists, performing the further step of displaying the second information.

66. The method of claim 65, wherein the first information comprises a name.

67. The method of claim 66, wherein the activation of the menu comprises

selecting the menu indicator for the menu
opening the menu
selecting a choice in the menu; and
activating the selected choice in the menu.

68. The method of claim 67, wherein selection of the menu indicator comprises moving a mouse pointer to the menu indicator.

69. The method of claim 68, wherein the opening of a menu comprises clicking on the menu indicator with a mouse button.

70. The method of claim 57 wherein at least part of the identifying occurs after the user selection.

71. The method of claim 57 wherein the identifying occurs after the user selection.

72. The method of claim 57, wherein the second information is associated with only part of the identified first information.

73. A computer readable medium, including program instructions related to information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program, and for performing the steps of:
identifying without user intervention or designation the first information; and

responding to a user selection by performing an operation related to a second information, the second information associated with the first information from the second application program.

74. The computer readable medium of claim 73 wherein the operation comprises displaying the second information.

75. The computer readable media of claim 73, wherein the first information is a name, and the operation performed is selected from a group consisting of generating an electronic

16

mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

76. The computer readable media of claim 73, wherein the operation performed is entering additional data into a database.

77. The computer readable media of claim 76, wherein the additional data is entered by a user.

78. The computer readable media of claim 76, wherein the additional data is located within the document.

79. A computer system related to information handling within a document operated on by a first application program, the document containing first information that can be utilized in a second application program, comprising:

means for identifying without user intervention or designation the first information; and
means for responding to a user selection by performing an operation related to a second information, the second information associated with the first information from the second application program.

80. The computer system of claim 79 wherein the operation comprises displaying the second information.

81. The computer system of claim 79, wherein the first information is a name, and the operation performed is selected from a group consisting of generating an electronic mail, a telex, a facsimile or a letter addressed to the name indicated by the first information.

82. The computer system of claim 79, wherein the operation is entering additional data into a database.

83. The computer system of claim 82, wherein the additional data is entered by a user.

84. The computer system of claim 82, wherein the additional data is located within the document.

85. A method for information handling within a document operated on by a first application program, the document containing first information entered by a user, the method comprising the steps of:

identifying without user intervention or designation the first information that can be utilized in a second application program, the first information selected from a group consisting of a name and an address; and

responding to a user selection by performing an operation related to a second information, the second information associated with all or part of the first information from the second application program, wherein the step of responding to the user selection further comprises the steps of:

initializing the second application program;
searching, using the second application program, for the second information associated with the first information; and

if said second information exists, retrieving and displaying the second information;

wherein said user selection comprises:

selecting a menu indicator for the menu;
opening the menu;
selecting a choice in the menu; and
activating the selected choice in the menu;

wherein selecting the menu indicator comprises moving the mouse pointer to the menu indicator;

wherein opening the menu comprises clicking on the menu indicator with a mouse button; and

wherein the second information is associated with at least part of the identified first information.

86. A method for assisting a computer operator to retrieve contact related information from a database when a document includes a name, the method comprising of the steps of:

US 7,496,854 B2

17

(1) using a first computer program to analyze the document, without direction from the operator, to identify the name,
 (2) using the identified name and a second computer program to search the database and to locate contact related information associated with the name, and
 (3) inserting the contact related information into the document,
 wherein steps (1)-(3) require only a single execute command.
87. The method of claim **86** wherein the contact related information comprises an address.
88. The method of claim **86** wherein the contact related information comprises a telephone number.
89. The method of claim **86** wherein the name comprises a business name.
90. The method of claim **86** wherein the name comprises a personal name.
91. The method according to claim **86** wherein the execute command is a selection from a menu.
92. The method according to claim **91** wherein the operator enters the execute command before step (2).
93. A method for assisting a computer operator to retrieve information from a database that is related to text in a document, the method comprising the steps of:
 (1) using a first computer program to analyze the document, without direction from the operator, to identify text in the document that can be used to search for related information,
 (2) using a second computer program and the text identified in step (1) to search the database and to locate related information, and
 (3) inserting the information located in step (2) into the document.
94. The method according to claim **93** wherein at least steps (2)-(3) take place following entry a single execute command.
95. The method according to claim **94** wherein the execute command is a selection from a menu.
96. A computer readable medium for information handling within a document operated on by a first application program, the document containing first information entered by a user, the computer readable medium including program instructions for performing the steps of:
 identifying without user intervention or designation the first information that can be utilized in a second application program, the first information selected from a group consisting of a name and an address; and
 responding to a user selection by performing an operation related to a second information, the second information associated with all or part of the first information from the second application program, wherein the step of responding to the user selection further comprises the steps of:
 initializing the second application program;
 searching, using the second application program, for the second information associated with the first information; and
 if said second information exists, retrieving and displaying the second information;
 wherein said user selection comprises:
 selecting a menu indicator for the menu;
 opening the menu;
 selecting a choice in the menu; and
 activating the selected choice in the menu;
 wherein selection the menu indicator comprises moving the mouse pointer to the menu indicator;
 wherein the opening of a menu comprises clicking on the menu indicator with a mouse button; and

18

wherein the second information is associated with at least part of the identified first information.
97. A computer readable medium for assisting a computer operator to retrieve contact related information from a database when a document includes a name, the computer readable medium including program instructions for performing the steps of:
 (1) using a first computer program to analyze the document, without direction from the operator, to identify the name,
 (2) using the identified name and a second computer program to search the database and to locate contact related information associated with the name, and
 (3) inserting the contact related information into the document,
 wherein steps (1)-(3) require only a single execute command.
98. A computer readable medium for assisting a computer operator to retrieve information from a database that is related to text in a document, the computer readable medium including program instructions for performing the steps of:
 (1) using a first computer program to analyze the document, without direction from the operator, to identify text in the document that can be used to search for related information,
 (2) using a second computer program and the text identified in step (1) to search the database and to locate related information, and
 (3) inserting the information located in step (2) into the document.
99. A system for information handling within a document operated on by a first application program, the system comprising:
 means for identifying without user intervention or designation the first information that can be utilized in a second application program, the first information selected from a group consisting of a name and an address; and
 means for responding to a user selection by performing an operation related to a second information, the second information associated with all or part of the first information from the second application program, wherein the means for responding to the user selection further comprises:
 means for initializing the second application program;
 means for searching, using the second application program, for the second information associated with the first information; and
 if said second information exists, means for retrieving and displaying the second information;
 wherein said user selection comprises:
 selecting a menu indicator for the menu;
 opening the menu;
 selecting a choice in the menu; and
 activating the selected choice in the menu;
 wherein selection the menu indicator comprises moving the mouse pointer to the menu indicator;
 wherein the opening of a menu comprises clicking on the menu indicator with a mouse button; and
 wherein the second information is associated with at least part of the identified first information.
100. A system for assisting a computer operator to retrieve contact related information from a database when a document includes a name, the system comprising:
 (1) means for using a first computer program to analyze the document, without direction from the operator, to identify the name,

US 7,496,854 B2

19

(2) means for using the identified name and a second computer program to search the database and to locate contact related information associated with the name, and

(3) means for inserting the contact related information into the document,

wherein (1)-(3) require only a single execute command.

101. A system for assisting a computer operator to retrieve information from a database that is related to text in a document, the system comprising:

20

(1) means for using a first computer program to analyze the document, without direction from the operator, to identify text in the document that can be used to search for related information,

(2) means for using a second computer program and the text identified in (1) to search the database and to locate related information, and

(3) means for inserting the information located in (2) into the document.

* * * * *

EXHIBIT 4

(12) **United States Patent**
Hedloy

(10) **Patent No.:** **US 7,921,356 B2**
(45) **Date of Patent:** **Apr. 5, 2011**

(54) **METHOD, SYSTEM AND COMPUTER READABLE MEDIUM FOR ADDRESSING HANDLING FROM A COMPUTER PROGRAM**

FOREIGN PATENT DOCUMENTS
EP 0 093 250 A2 9/1983
(Continued)

(75) Inventor: **Atle Hedloy**, Madrid (ES)

OTHER PUBLICATIONS

(73) Assignee: **Arendi S.A.R.L.**, Luxembourg (LU)

Person et al., "Special Edition Using Microsoft Word 97" by Que Publisher Pub. date Dec. 16, 1996, pp. 475-514.*

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(Continued)

(21) Appl. No.: **12/841,302**

Primary Examiner — Doug Hutton
Assistant Examiner — Quoc A Tran

(22) Filed: **Jul. 22, 2010**

(74) *Attorney, Agent, or Firm* — Sunstein Kann Murphy & Timbers LLP

(65) **Prior Publication Data**

US 2010/0281354 A1 Nov. 4, 2010

Related U.S. Application Data

(63) Continuation of application No. 12/182,048, filed on Jul. 29, 2008, which is a continuation of application No. 09/923,134, filed on Aug. 6, 2001, now Pat. No. 7,496,854, which is a continuation of application No. 09/189,626, filed on Nov. 10, 1998, now Pat. No. 6,323,853.

(57) **ABSTRACT**

A method, system and computer readable medium for providing for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

(51) **Int. Cl.**

G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/230**; 715/825; 715/234; 715/752; 715/853; 715/711

(58) **Field of Classification Search** 715/230, 715/209, 231, 273, 277, 792, 783
See application file for complete search history.

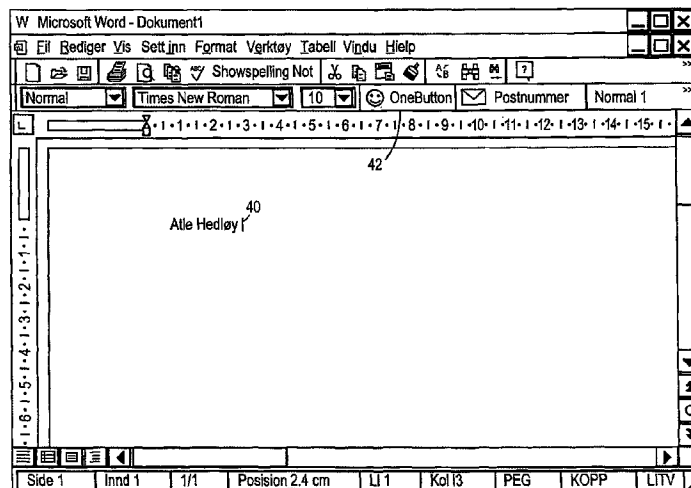
(56) **References Cited**

U.S. PATENT DOCUMENTS

3,350,695 A 10/1967 Kaufman et al. 340/172.5
4,270,182 A 5/1981 Asija 364/900

(Continued)

20 Claims, 14 Drawing Sheets



US 7,921,356 B2

Page 2

U.S. PATENT DOCUMENTS

4,674,065	A	6/1987	Lange et al.	364/900	6,092,074	A	7/2000	Rodkin et al.	707/102
5,226,117	A	7/1993	Miklos	395/157	6,108,686	A	8/2000	Williams	709/202
5,267,155	A	11/1993	Buchanan et al.	364/419.14	6,151,584	A	11/2000	Papierniak et al.	705/10
5,331,555	A	7/1994	Hashimoto et al.	364/419.07	6,178,411	B1	1/2001	Reiter	705/408
5,355,472	A	10/1994	Lewis	395/600	6,182,133	B1	1/2001	Horvitz	709/223
5,375,200	A	12/1994	Dugan et al.	395/159	6,192,380	B1	2/2001	Light et al.	707/505
5,388,150	A	2/1995	Schneyer et al.	379/67	6,208,975	B1	3/2001	Bull et al.	705/14
5,392,386	A	2/1995	Chalas	395/155	6,233,570	B1	5/2001	Horvitz et al.	706/11
5,416,901	A	5/1995	Torres	395/159	6,253,188	B1	6/2001	Witek et al.	705/14
5,491,783	A	2/1996	Douglas et al.	395/159	6,260,035	B1	7/2001	Horvitz et al.	706/60
5,491,784	A	2/1996	Douglas et al.	395/159	6,262,730	B1	7/2001	Horvitz et al.	345/337
5,491,785	A	2/1996	Robson et al.	395/162	6,285,987	B1	9/2001	Roth et al.	705/27
5,495,565	A	2/1996	Millard et al.	395/146	6,304,881	B1	10/2001	Halim et al.	707/201
5,500,859	A	3/1996	Sharma et al.	370/81	6,317,761	B1	11/2001	Landsman et al.	707/513
5,530,853	A	6/1996	Schell et al.	395/600	6,323,853	B1	11/2001	Hedloy	345/339
5,546,447	A	8/1996	Skarbo et al.	379/142	6,338,957	B2	1/2002	Onishi et al.	435/108
5,566,330	A	10/1996	Sheffield	395/600	6,373,940	B2	4/2002	Shaffer et al.	379/355.02
5,576,955	A	11/1996	Newbold et al.	395/795	6,377,965	B1	4/2002	Hachamovitch et al.	707/534
5,579,467	A	11/1996	Capps	395/768	6,385,592	B1	5/2002	Angles et al.	705/14
5,587,911	A	12/1996	Asano et al.	364/444.2	6,421,691	B1	7/2002	Nishiyama et al.	707/507
5,606,712	A	2/1997	Hidaka	395/800	6,421,693	B1	7/2002	Nishiyama et al.	707/507
5,634,121	A	5/1997	Tracz et al.	395/602	6,499,021	B1	12/2002	Abu-Hakima	706/10
5,640,565	A	6/1997	Dickinson	395/683	6,651,217	B1	11/2003	Kennedy et al.	715/507
5,666,502	A	9/1997	Capps	345/352	6,738,615	B1	5/2004	Chow et al.	455/415
5,708,804	A	1/1998	Goodwin et al.	395/603	6,826,407	B1	11/2004	Helferich	455/466
5,724,597	A	3/1998	Cuthbertson et al.	395/793	7,039,599	B2	5/2006	Merriman et al.	705/14
5,732,229	A	3/1998	Dickinson	395/334	7,051,019	B1	5/2006	Land et al.	707/4
5,732,395	A	3/1998	Alexander Silverman	704/260	7,136,853	B1	11/2006	Kohda et al.	707/6
5,734,871	A	3/1998	Kleinerman et al.	395/500	7,155,451	B1	12/2006	Torres	707/104.1
5,742,905	A	4/1998	Pepe et al.	455/461	7,216,292	B1	5/2007	Snapper et al.	715/507
5,754,636	A	5/1998	Bayless et al.	379/142	7,272,604	B1	9/2007	Hedloy	707/10
5,761,656	A	6/1998	Ben-Shachar	707/4	7,496,854	B2	2/2009	Hedloy	715/780
5,761,663	A	6/1998	Lagarde et al.	707/10	7,505,974	B2	3/2009	Gropper	707/10
5,774,887	A	6/1998	Wolff et al.	707/1	7,509,349	B2	3/2009	Celik	707/200
5,778,367	A	7/1998	Wesinger, Jr. et al.	707/10	2002/0188683	A1	12/2002	Lytle et al.	709/206
5,781,189	A	7/1998	Holleran et al.	345/335	2003/0033288	A1	2/2003	Shanahan et al.	707/3
5,787,451	A	7/1998	Mogilevsky	707/533	2006/0047644	A1	3/2006	Bocking et al.	707/4
5,793,972	A	8/1998	Shane	395/200.49	2007/0244907	A1	10/2007	Hedloy	707/10
5,794,228	A	8/1998	French et al.	767/2	2008/0313159	A1	12/2008	Hedloy	707/3
5,794,259	A	8/1998	Kikinis	707/507	2010/0211600	A2	8/2010	Hedloy	707/770
5,799,302	A	8/1998	Johnson et al.	707/7					
5,805,886	A	9/1998	Skarbo et al.	395/685					
5,805,911	A	9/1998	Miller	395/796					
5,815,830	A	9/1998	Anthony	707/6					
5,826,257	A	10/1998	Snelling, Jr.	707/4					
5,835,089	A	11/1998	Skarbo et al.	345/335					
5,859,636	A	1/1999	Pandit	715/204					
5,860,073	A	1/1999	Ferrel et al.	715/255					
5,864,848	A	1/1999	Horvitz et al.	707/6					
5,873,107	A	2/1999	Borovoy et al.	707/501					
5,873,108	A	2/1999	Goyal et al.	707/507					
5,884,309	A	3/1999	Vanechanos, Jr.	707/10					
5,893,093	A	4/1999	Wills	707/5					
5,896,321	A	4/1999	Miller et al.	365/189.01					
5,896,533	A	4/1999	Ramos et al.	395/680					
5,907,838	A	5/1999	Miyasaka et al.	707/4					
5,913,214	A	6/1999	Madnick et al.	707/10					
5,923,848	A	7/1999	Goodhand et al.	395/200.49					
5,924,074	A	7/1999	Evans	705/3					
5,924,090	A	7/1999	Krellenstein	707/5					
5,926,808	A	7/1999	Evans et al.	707/3					
5,930,471	A	7/1999	Milewski et al.	395/200.04					
5,930,764	A	7/1999	Melchione et al.	705/10					
5,940,484	A	8/1999	DeFazio et al.	379/142					
5,946,647	A	8/1999	Miller et al.	704/9					
5,946,679	A	8/1999	Ahuja et al.	707/3					
5,999,938	A	12/1999	Bliss et al.	707/102					
6,006,218	A	12/1999	Breese et al.	707/3					
6,006,351	A	12/1999	Peretz et al.	714/751					
6,012,075	A	1/2000	Fein et al.	707/540					
6,018,571	A	1/2000	Langlois et al.	379/207					
6,021,403	A	2/2000	Horvitz et al.	706/45					
6,026,398	A	2/2000	Brown et al.	707/5					
6,028,605	A	2/2000	Conrad et al.	345/354					
6,067,565	A	5/2000	Horvitz	709/218					
6,081,788	A	6/2000	Appleman et al.	705/14					
6,085,201	A	7/2000	Tso	707/505					
6,085,226	A	7/2000	Horvitz	709/203					

FOREIGN PATENT DOCUMENTS

EP	1 630 642	3/2006
GB	2 318 703	4/1998
GP	2196454	4/1988
WO	WO 97/21183	6/1997
WO	WO 98/16890	4/1998
WO	WO 98/19259	5/1998
WO	WO 98/34391	8/1998

OTHER PUBLICATIONS

Getting Results with Microsoft Office 97, 701 pages, © 1995-1997.
 Apple Data Detectors User's Manual, 16 pages, © 1997.
 CoStar User Manual for AddressMate and AddressMate Plus, 226 pages, © 1994-1995.
 Anind K. Dey, et al., "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services," *Knowledge-Based Systems*, vol. 11, No. 1, pp. 3-13, 1998.
 Bonnie A. Nardi, et al., Collaborative, Programmable Intelligent Agents, Website, pp. 1-11, Mar. 1998.
 Addressmate Automatic Envelope Addressing Program, User'S Manual, 24 pages, © 1991.
 Brown, Peter, Unix Guide, 1 page, 1995.
 N.D. Beitner, et al., Multimedia Support and Authoring in Microcosm: an Extended Model, 12 pages, no date available.
 McMahon, Lee E., SED—A Non-interactive Text Editor, Bell Laboratories, 10 pages, Aug. 15, 1978.
 SED(1), BSD Reference Manual Page, 6 pages, Dec. 30, 1993.
 "AddressMate Incorporates Correction Capability," The New York Law Publishing Company, 1 page, Jan. 1996.
 Mar. 1996 Reviews, website available at www.techweb.com/winmag/library/1996/0396/03rvh002.htm, 9 pages.
 Magid, Lawrence J., "Addressing the Matter of Labels," Los Angeles Times, 2 pages, Sep. 23, 1996.
 Important Note for New AddressMate Users, 1 page.
 Important Tips for LabelWriter Owners Using AddressMate, 3 pages, Jan. 1995.

US 7,921,356 B2

Page 3

- Addressing the Issues, 4 pages, Jun. 24, 1993.
- "Whye Do it the Olde Way" AddressMate Advertisement, 2 pages, no date available.
- Robinson, Philip, "The Envelope, Please: It's AddressMate," San Jose Mercury News, Sunday, 1 page, Jul. 19, 1992.
- Supkoff, Steve, AddressMate, PCM, 2 pages, Aug. 1992.
- Shannon, L.R., "Addressing Envelopes," The New York Times, 2 pages, Tuesday, Jul. 14, 1992.
- Schwabach, Bob, "Addressing for Success," On Computers Column, 1 page, Mar. 16, 1992.
- Infoworld, p. 15, Mar. 16, 1992.
- Hart, Ken, Simplify Envelope Printing with AddressMate, Computer Shopper, 1 page, no date available.
- "Colvin's Beta-Testing of AddressMate Sparks Developer's Appreciation," Westview, 1 page, Apr. 6, 1992.
- Press Release: AddressMate Software Automatically Addresses and Bar Codes Envelopes to Save Time and Money, 2 pages, Mar. 9, 1992.
- Long, George M., Letter to Mr. David Block, 1 page, Nov. 11, 1994.
- Damiano, Philip J., Letter to Mr. David Block, 2 pages, Jan. 5, 1994.
- Kahn, Ed, "Envelope Addressing Finally Simplified AddressMate Does Addressing and Much More," Microtimes, 3 pages, Nov. 27, 1995.
- Bernthal, Kristen, "LabelWriter XI Plus," PC Catalog, 1 page, Aug. 18, 1995.
- Plotkin, David, "Address for Success," Bay Area Computer Currents, p. 36-38, Mar. 21, 1995.
- Address Fixer for Microsoft Word and Office, Product Box, 10 pages, no date available.
- AddressMate Plus, Product Box and License Agreement, 16 pages, no date available.
- Getting Results with Microsoft Office for Windows 95, Version 7.0, 635 pages, © 1995-1997.
- Microsoft Word, User's Guide, Version 6.0, 449 pages, © 1993-1994.
- CTAGS(1) Manual Page, 3 pages, Oct. 11, 1993 (Last Change).
- Bernstein, Matt, An Apprentice that Discovers Hypertext Links, pp. 213-223, not date available.
- Contextual Menu Manager/Apple Data Detectors, 5 pages, no date available.
- Wilson, Eve, "Guiding Lawyers: Mapping Law into Hypertext," Artificial Intelligence Review 6, pp. 161-189, 1992.
- P.J. Brown, et al., A Help System Based on UNIX Help Manual, 7 pages, Feb. 1987.
- Charles H. Franke III, et al., Authoring a Hypertext Unix Help Manual, 8 pages, © 1995.
- Brown, P.J., Guide User Manual, 1985, sixteenth impression, 55 pages, Apr. 1995.
- Wilson, Eve, "Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System," Literary and Linguistic Computing, vol. 5, No. 2, 11 pages, 1990.
- John Robertson, et al., The Hypermedia Authoring Research Toolkit, ECHT '94 proceedings, pp. 177-185, Sep. 1994.
- What is Wild and Wild History webpages, available at wiki.org/wild.cgi?WhatIsWiki and www.c2.org/cgi/wild?WikiHistory, 8 pages, no date available.
- Wilson, Eve, Integrated Information Retrieval for Law in a Hypertext Environment, Annual ACM Conference on Research and Development in Information Retrieval, INRIA, 23 pages, 1988.
- Multimedia Hyperlinks Automatically Created for Reference Documents, Research Disclosure, 2 pages, Jun. 1993.
- Wild Wild Origin, <http://www.c2.com/cgi/wiki?WikiWikiOrigin>, 2 pages, no date available.
- Blake, Joy-Lyn, WikiWikiWeb, Computer World, 4 pages, Jan. 29, 2001.
- Microsoft's Supplemental Responses to Arendi's Interrogatories, in *Arendi USA et al. v. Microsoft Corporation*, Civil Action 02CV-343 (ECT) from the United States District Court for the District of Rhode Island, 4 pages, Apr. 7, 2003.
- "Apple Introduces Internet Address Detectors," Press Release, 3 pages, 2 pages, Sep. 8, 1997.
- Cunningham, Cara, "Apple Kicks Off Macworld with Talk of Revival," new software demos, InfoWorld Electric, Aug. 7, 1996.
- Staten, James, "Apple Looks to the Future," MacWeek, 2 pages, Aug. 7, 1996.
- Simmons, Mark, "Striking a Key Note," Mac Addict Online, 5 pages, Aug. 8, 1996.
- Miller, Jim, Email regarding Apple Data Detectors, 1 page, Jan. 8, 1997.
- Apple Data Detectors—Now Shipping web page, 5 pages, Jan. 6, 1997.
- The Apple Data Detectors FAQ, 8 pages, Jan. 6, 1997.
- Apple Data Detector Webpages, available on web.archive.org/web/20020601164217/www.apple.com/applescript/data_detectors, 15 pages, © 2002.
- Apple Data Detectors 1.0.2 Read Me, 3 pages, no date available.
- Developer's Guide to Apple Data Detectors, 34 pages, Dec. 1, 1997.
- AppleScript Editors, Utilities & Environments, available at www.applescriptsourcebook.com/links/applescripteditors.html, 4 pages, dated Jan. 11, 2004.
- eMailman Internet Address Detectors, 3 pages, © 1996-2000.
- Tannehill, Steve, News from Jul. 1997, 6 pages, Jul. 31, 1997.
- Control-Click! The Archive, 2 pages, © 1997-2000.
- Contextual Menus: One of the System 8's Greatest Features, in ApplePress, 10 pages, no date available.
- Trygve's CMM Plug-Ins Home, available at web.archive.org/web/19980130053511/www.bombaydigital.com/cmms, 7 pages, no date available.
- ADD Depot, available from [Web.archive.org/web/20000819091818/http://homepage.mac.com/Mathewmiller/add](http://web.archive.org/web/20000819091818/http://homepage.mac.com/Mathewmiller/add), 4 pages, Jan. 12, 2004.
- Press Release: Apple Introduces Internet Address Detectors, 4 pages, Sep. 8, 1997.
- Langberg, Mike, "Show of Potential Apple Breaks New Ground By Displaying What's on Its Drawing Board 'Innovation is at the Heart of What We Do'", San Jose Mercury News, 2 pages, Aug. 7, 1996.
- Apple Introduces Internet Address Detectors, Newsbytes, 11 pages, Sep. 29, 1997.
- Williams, Greg Strategy Mosaic: Understanding Apple's Dual OS Strategy, 7 pages, no date available.
- "Taking [control] of your Mac with System 8," The MacAuthority, 6 pages, Jan. 1998.
- Apple Data Detectors 1.0.2, TidBITS Updates, 1 page, Mar. 8, 1998.
- Apple Data Detectors 1.0.2, TidBITS #419, 1 page, Mar. 9, 1998.
- Engst, Tanya, More on Context on Contextual Menus, tidbits #399, 2 pages, Sep. 29, 1997.
- Engst, Tanya, Of Mice and Menus, TidBITS #399, 3 pages, Sep. 22, 1997.
- Whaley, Charles "Will This be Enough to Kick-Start Apple?," Computing Canada, 4 pages, Aug. 4, 1997.
- MacOS8.com—Mac OS 8 Indepth, 3 pages, no date available.
- "A Farewell to the Apple Advanced Technology Group," SIGCHI, vol. 30, No. 2, 3 pages, Apr. 1998.
- James R. Miller, et al., From Documents to Objects, in SIGCHI, vol. 30, No. 2, 11 pages, Apr. 1998.
- Thomas Bonura et al., Drop Zones, in SIGCHI, vol. 30, No. 2, 8 pages, Apr. 1998.
- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island in Case No. 02-CV-343, Judge Ernest C. Tones, Brief for the Defendant-Cross Appellant Frank E. Scherkenbach, 59 pages, Jul. 7, 2005.
- United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law that the '853 Patent is Invalid, 16 pages, Oct. 15, 2004.
- United States District Court for the District of Rhode Island, Civ. A. No. 02-CV-343 (ECT), Plaintiffs Reply Memorandum in Support of their Motion for New Trial, Francis A. Connor, 9 pages, Nov. 4, 2004.
- United States Court of Appeals for the Federal Circuit, Arendi Reply Brief, 37 pages, Sep. 2, 2005.
- United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Memorandum in Support of Defendant Microsoft Corporation's Opposition to Arendi's Motion for a New Trial, Patricia A. Sullivan, 15 pages, Oct. 27, 2004.
- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island,

US 7,921,356 B2

Page 4

- 02-CV-343 (ECT), Brief of Plaintiffs-Appellants Arendi U.S.A., Inc. and Arend Holding Limited, Donald R. Dunner, 61 pages, Apr. 25, 2005.
- United States Court of Appeals for the Federal Circuit, Appeal from the United States District Court for the District of Rhode Island, 02-CV-343 (ECT), Reply Brief for Defendant-Cross Appellant, Frank E. Scherkenbach, 27 pages, Oct. 3, 2005.
- United States District Court for the District of Rhode Island, Order Denying Defendant Microsoft Corporation's Motion for Judgment as a Matter of Law That the '853 Patent is Invalid, C.A. No. 02-343T, Ernest C. Tones, Chief Judge, 6 pages, Nov. 30, 2004.
- United States District Court for the District of Rhode Island, C.A. No. 02-343T, Order Denying Plaintiffs' Motion for New Trial, Ernest C. Tones, Chief Judge, 18 pages, Nov. 30, 2004.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from 2:00 Pm, 48 pages, Sep. 13, 2004, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 14, 2004, 165 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 15, 2004, 111 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 16, 2004, 142 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 17, 2004, 66 pages, Providence, RI.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 20, 2004, Providence, RI, 225 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 21, 2004, Providence, RI, 170 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 22, 2004, Providence, RI, 156 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 23, 2004, Providence, RI, 140 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 27, 2004, Providence, RI, 17 pages.
- Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.* C.A. No. 02-343T Court Transcript from Sep. 28, 2004, Providence, RI, 106 pages.
- Novell GroupWise User's Guide for Windows 16-BIT, Version 5.2, Novell, Inc., Orem, Utah, MS 125993, 235 pages, 1993.
- Novell GroupWise Webaccess User's Guide, Novell, Inc., Orem, Utah, MS 126785, 37 pages, 1998.
- Novell GroupWise User's Guide for Windows 32-BIT, Novell, Inc., Orem, Utah, MS 126463, 322 pages, 1998.
- Andrew Wood, et al., "CyberDesk: Automated Integration of Desktop and Network Services", GVVU Technical Support, OIT-GVVU-97-11, 5 pages, May 1997.
- Gregory D. Abowd, et al., "Applying Dynamic Integration as a Software Infrastructure for Context-Aware Computing," GVVU Technical Report, GIT-GVVU-97-18, 21 pages, Sep. 1997.
- Gregory D. Abowd, et al., "Context-awareness in Wearable and Ubiquitous Computing," GVVU Technical Report, GIT-GVVU-97-11, 13 pages, Mar. 1997.
- Apple Internet Address Detectors User's Manual, 16 pages, © 1997.
- Apple Introduces Internet Address Detectors, 4 pages, Sep. 8, 1997.
- Contextual Menu Manager/Apple Data Detectors, 5 pages, date unavailable.
- Anind K. Dey, et al., CyberDesk: The Use of Perception in Context-Aware Computing, PUI Workshop Submission, Proc. Of 1997 Workshop on Perceptual User Interfaces (PUI '97), 4 pages, Oct. 1997.
- Dey, Anind K., "Context-Aware Computing: The CyberDesk Project," Future Computing Environments, AAAI '98 Spring Symposium, Stanford University, pp. 51-55, Mar. 23-25, 1998.
- Anind K. Dey, et al., "CyberDesk: A Framework for Providing Self-Integrating Ubiquitous Software Services," GVVU Technical Report, GIT-GVVU-97-10, 18 pages, 1997.
- L. Nancy Garrett, et al., "Intermedia: Issues, Strategies, and Tactics in the Design of a Hyermedia Document System," pp. 163-174, © 1986.
- Wilson, Eve, "Links and Structures in Hypertext Databases for Law," Proceedings of the First European Conference on Hypertext, INRIA, 20 pages, Nov. 1990.
- Langberg, Mike, "Innovation Is at the Heart of What We Do", Apple Breaks New Ground by Displaying What's on It's Drawing Board," Mercury News, pp. 1-2, Aug. 7, 1996.
- Henry Lieberman, et al., Training Agents to Recognize Text by Example, Proc. of the Third Annual Conference on Autonomous Agents, Seattle, WA, 13 pages, 1999.
- Milind S. Pandit, et al., "The Selection Recognition Agent: Instance Access to Relevant Information and Operations," Proc. of Intelligent User Interfaces 1997, Orlando, FL, 11 pages, 1997.
- Spell, iSpell, Spellout, 71 pages, © 1994.
- Microsoft Word 97 Help File entitled "Automatically Check Spelling and Grammar as You Type," 1 page, 2003.
- Microsoft Word 97 Help File entitled "Turn Automatic Changes On or Off," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Automatically Correct Text," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Field Codes: Hyperlink Field," 1 page, no date available.
- Microsoft Word 97 Help File entitled "Change the contents of an AutoCorrect Entry," 1 page, no date available.
- Excerpt from Jury Charge Transcript for C.A. No. 02-343T, *Arendi USA, Inc. et al. vs. Microsoft Corporation, et al.*, 1 page, Sep. 28, 2004.
- Microsoft Corporation and Dell Inc.'s Answer, Affirmative Defenses and Counterclaims to Complaint, *Arendi Holding Ltd. vs. Microsoft Corp. et al.*, C.A. No. 09-119-JJF, from the United States District Court for the District of Delaware, pp. 1-17, Apr. 30, 2009.
- "More about The Database," <http://c2.com/cgi/wiki?MoreAboutTheDatabase>, pp. 1-3, last edited May 7, 2009.
- "Ward Cunningham," <http://c2.com/cgi/wiki?WardCunningham>, pp. 1-15, last edited Apr. 23, 2009.
- "Adding New Pages," <http://c2.com/cgi/wiki?AddingNewPages>, pp. 1-2, last edited Jun. 8, 2009.
- "Wiki Wiki Web," <http://c2.com/cgi/wild?WikiWikiWeb>, 1 page, last edited Jun. 1, 2009.
- Quinlan, Tom, "Apple System Software Branches Out," 3 pages, Mar. 16, 1992.
- "Introducing AddressMate Plus, The Address Book Manager for Microsoft Word," 4 pages, May 1996.
- "Corel Professional Office 7 Quick Results," 534 pages, no date available.
- "Getting Results with Microsoft Office for Windows 95," Microsoft, 636 pages, 1995.
- "AddressMate for Windows, Adds all the missing pieces to your word processor," 6 pages, no date available.
- Opposition against European Patent EP 1 171 836, Letter to the European Patent Office in Preparation for the Oral Proceedings scheduled for May 26, 2009, 31 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 1 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 2 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 3 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 4 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 5 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 6 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 7 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 8 Claims, 3 pages, Mar. 26, 2009.

US 7,921,356 B2

Page 5

- Opposition against European Patent EP 1 171 836, Auxiliary Request 9 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 10 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 11 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 12 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 13 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 14 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 15 Claims, 3 pages, Mar. 26, 2009.
- Opposition against European Patent EP 1 171 836, Auxiliary Request 16 Claims, 3 pages, Mar. 26, 2009.
- Expert Report of Professor Benjamin Goldberg, 43 pages, dated Mar. 10, 2008.
- Supplemental Expert Report of Professor Benjamin Goldberg, 15 pages, May 6, 2008.
- Second Supplemental Expert Report of Professor Benjamin Goldberg, 32 pages, Mar. 26, 2009.
- Expert Report of Professor Calvin Gidney, III, 3 pages, Mar. 26 2009.
- European Patent Office, Summons to Attend Oral Proceedings pursuant to Rule 115(1) EPC, 18 pages, Jan. 23, 2009.
- Affidavit of James Miller, 18 pages, Jul. 19, 2006.
- Affidavit of David Block, 23 pages, Jul. 21, 2006.
- Order Re Claim Construction, in *Arendi U.S.A. Inc. et al. v. Microsoft Corp.*, CA No. 02-343-T, from United States District Court for the District of Rhode Island, 4 pages, Sep. 27, 2004.
- Expert Report of W. Bruce Croft, Ph.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 66 pages, Mar. 1, 2004.
- Supplemental Expert Report of W. Bruce Croft, Ph.D., in *Arendi U.S.A., Inc. et al. v. Microsoft Corp.*, CA No. 02-CV-343 (ECT), from United States District Court for the District of Rhode Island, 42 pages, Jun. 9, 2004.
- Affirming Claim Construction Order, in *Arendi USA et al. v. Microsoft Corp.*, No. 05-1170, 05-1171, from Court of Appeals for the Federal Circuit, 1 page, Feb. 8, 2006.
- Miller, Jim, "Portfolio; Innovative Interfaces for Information Management," Website: <http://www.miramontes.com/portfolio/add1.html>, 1 page, no date available.
- Church, Kenneth et al., "Commercial Applications of Natural Language Processing," 30 pages, Mar. 13, 2003.
- Yankelovich, Nichole et al., "Intermedia: The concept and the Construction of a Seamless Information Environment," *Electronic Publishing Technologies*, pp. 81-96, Jan. 1988.
- Utting, Kenneth et al., "Context and Orientation in Hypermedia Networks," *AMC Transactions on Information Systems*, vol. 7, No. 1, pp. 58-84, Jan. 1989.
- Kahn, Paul, "Linking Together Books: Experiments in Adapting Published Material into Intermedia Documents," *Hypermedia*, vol. 1 No. 2, pp. 1-37, 1989.
- Combs, James, "Hypertext, Full-Text and Automatic Linking," pp. 83-98, 1990.
- Catlin, Karen Smith et al., "Hypermedia Templates: An Authors Tools," *Hypertext 1991 Proceedings*, pp. 147-160, Dec. 1991.
- Meyrowitz, Norman, "The Missing Link. Why We're All Doing Hypertext Wrong," *The Society of Text, Hypertext, Hypermedia, and the Social Construction of Information*, pp. 107-114, no date available.
- Halasz, Frank, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems," *Hypertext 87 Papers*, pp. 345-365, Nov. 1987.
- Haan, Bernard et al., "IRIS Hypermedia Services," pp. 36-51, Jan. 1992.
- McKnight, Dillon, "Review, The Textbook of the Future," *Hypertext: a Psychological Perspective*, pp. 19-51, 8 pages, Mar. 1993.
- Bouvin, Niels Olof, "Augmenting the Web Through Open Hypermedia," 95 pages, Nov. 2000.
- Halasz, Frank et al., "The Dexter Hypertext Reference Model," pp. 95-133, Dec. 7, 1989.
- Weinreich, Harald et al., "The Look of the Link—Concepts for the User Interface of Extended Hyperlinks," 10 pages, no date available.
- Fountain, Andrew et al., "Microcosm: An Open Model for Hypermedia with Dynamic Linking," *Hypertext: Concept Systems and Applications, Proceedings of the First European Conference on Hypertext*, France, 16 pages, Nov. 1990.
- Justus: Wilson, Eve, "Cases for Justus: Preparing a Case Database for a Hypertext Information Retrieval System," *Library and Linguistics Computing*, vol. 5, No. 2, pp. 119-128, 1990.
- "Claris Em@iler Getting Started," Claris Corporation, 57 pages, © 1995-1997.
- "OmniPage Pro for Windows 95," Version 7.0 Caere Corporation, 57 pages, © 1996.
- Moi Khor, Siew, Microsoft Office Excel 2003 Preview, 17 pages, Jun. 2003.
- Microsoft Corporation and Dell Inc.'s Invalidity Contentions, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 126, Aug. 14, 2009.
- Exhibit A, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-35, Aug. 14, 2009.
- Exhibit AA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Aug. 14, 2009.
- Exhibit AAA, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit B, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit BB, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Aug. 14, 2009.
- Exhibit C, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Aug. 14, 2009.
- Exhibit CC, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit D, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit DD, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-22, Aug. 14, 2009.
- Exhibit E, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit EE, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Aug. 14, 2009.
- Exhibit F, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit FF, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-40, Aug. 14, 2009.
- Exhibit G, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit GG, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit H, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-33, Aug. 14, 2009.
- Exhibit HH, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-32, Aug. 14, 2009.
- Exhibit I-1 through I-6, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-1800, Aug. 14, 2009.

US 7,921,356 B2

Page 7

- Plaintiffs Answer to Defendants' First Amended Counterclaims, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-19, Oct. 30, 2009.
- Microsoft Corp. and Dell, Inc.'s Second Amended Answer, Affirmative Defenses and Counterclaims to Complaint, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-34, Dec. 8, 2009.
- Opposition for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-2, Jul. 26, 2006.
- In Response to Notice of Opposition for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-21, Mar. 7, 2007.
- Decision Revoking European Patent No. EP-B- 1171836, European Patent Office, pp. 1-13, Jul. 9, 2009.
- Notice of Appeal for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-2, Nov. 1, 2009.
- Grounds of Appeal for European Patent No. EP-B- 1171836, in the European Patent Office, pp. 1-62, Nov. 19, 2009.
- Plaintiff's Answer to Microsoft Corporation's and Dell Inc.'s Second Amended Counterclaims, in *Arendi Holding Ltd. v. Microsoft Corp. and Dell Inc.*, CA No. 09-119-JJF-LPS, from United States District Court for the District of Delaware, pp. 120, Dec. 14, 2009.
- Submission of opponent (Microsoft) in opposition against European Patent No. 1 171 836, Appeal T1779/09-3501, in the European Patent Office, 32 pages, Mar. 25, 2010.
- Second Affidavit of David Block, 15 pages, May 15, 2009.
- Second Affidavit of James Miller, 36 pages, May 15, 2009.
- Hall, Wendy, et al., "Rethinking Hypermedia," Kluwer Academic Publishers, 87 pages, 1996.
- Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 196 pages, Aug. 13, 2004.
- Exhibits for the Deposition of David Block, *Arendi U.S.A., Inc. v. Microsoft Corp.*, Case No. 02-CV-343 (ECT), from United States District Court for the Northern District of California, 334 pages, Aug. 13, 2004.
- Writ of Summons to Arendi Holding Co., District Court in the Hague and English translation, 65 pages, Oct. 26, 2005.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, 55 pages, date unknown.
- Microsoft's Reply in the Main Action, Defense in the Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 51 pages, date unknown.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, 26 pages, Sep. 24, 2008.
- Microsoft's Reply in Counterclaim for Infringement, District Court in the Hague, No. 2006/0782, English Translation, 23 pages, Sep. 24, 2008.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising A Counterclaim, With Exhibit, District Court in the Hague, No. 2006/0782, 57 pages, Jan. 31, 2007.
- Statement of Defense in the Principal Action and With Regard to the Interlocutory Claim Also Comprising A Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 23 pages, Jan. 31, 2007.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the Counterclaim With Exhibits, District Court in the Hague, No. 2006/0782, 123 pages, Mar. 12, 2008.
- Statement of Rejoinder in the Principal Action (Incl. With Regard to the Interlocutory Claims) Also Comprising Statement of Reply in the Counterclaim, District Court in the Hague, No. 2006/0782, English Translation, 33 pages, Mar. 12, 2008.
- Letter from Abraham H. Spierer to David Block, Dec. 19, 1994.
- Claris Em@iler User's Manual, 187 pages, © 1995-1997.
- "Word Lookup Data Detectors," Dictionary.com, available at <http://dictionary.reference.com/mac/add.html>, 3 pages, © 2004.
- Apple.com internet web pages (marked "MS 118139-MS 118143"), 5 pages, © 2001.
- "Overview: What are Apple Data Detectors", Apple.com internet web pages (marked "MA 118078-MS 118092"), 15 pages, © 2002.
- Addressmate Plus User Manual Addendum, 2 pages, Nov. 1, 1995.
- Now Software, Inc., "Now Contact & Up*to*Date Quick Reference Guide", 18 pages, © 1992-95.
- Now Software, Inc., "Now Contact version 3.5 User's Guide", 229 pages, © 1992-95.
- Now Software, Inc., "Now Up*to*Date version 3.5 User's Guide", 210 pages, © 1992-95.
- Now Software, Inc., "Now Utilities Quick Reference Guide", 6 pages, no date available.
- Now Software, Inc., "What's New in Now Up*to*Date & Now Contact", 14 pages, © 1995.
- Reply Brief of Plaintiffs-Appellants Arendi USA, Inc. and Arendi Holding Limited, United States Court of Appeals for the Federal Circuit, 37 pages, Sep. 2, 2005.
- Apple Data Detectors/Internet Address Detectors, ADD-IAD_1.0.2_Info.txt, 1 page, Mar. 5, 1998.
- Cortinas, M., Data Ratchet extracts info, ZD.com, 2 pages, Apr. 3, 1997.
- Glucose Unveils Data Ratchet v.1.1!, Glucose Development Corp., 4 pages, 1997.
- Glucose Releases Data Ratchet Source, Glucose Press Release, 1 page, Dec. 18, 1999.
- Are you a document mechanic?, from Glucose Development Corp.'s webpage (<http://glulabs.com>), 2 pages, 2002.
- Glucose Open Source Tools, from Glucose Development Corp., 2 pages, 1999.
- E-Mail Net Differences—Eudora Pro and E-Mail Connection shine in their own ways . . . , Communications Week, 4 pages, Jan. 6, 1997.
- Internet Update Sep. 10, 1997, Newsbytes news Network, 5 pages, Sep. 10, 1997.
- Brouwer, Rene, ADD → Address Book, 1 page, Jun. 8, 1998.
- Byrne, Mike, Add → New Mail, 1 page, Aug. 23, 1998.
- MacCentral: Apple Macintosh News, 4 pages, Jan. 12-13, 2004.
- Turney, P., Learning to Extract Key Phrases from Text, National Research Council of Canada, 46 pages, Feb. 1999.
- Deposition of Atle Hedloy in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 162-213, Mar. 7, 2010.
- Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-67, Apr. 9, 2010.
- Exhibit 15, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware , pp. 1-26, Apr. 9, 2010.
- Exhibit 16, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.
- Exhibit 17, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 18, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-23, Apr. 9, 2010.
- Exhibit 19, in Deposition of David Block in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-13, Apr. 9, 2010.
- Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-115, Apr. 9, 2010.
- Exhibit 11, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-7, Apr. 9, 2010.
- Exhibit 12, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-2, Apr. 9, 2010.
- Exhibit 13, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-28, Apr. 9, 2010.
- Exhibit 14, in Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-18, Apr. 9, 2010.
- Exhibit 15, Deposition of James R. Miller in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-36, Apr. 9, 2010.

AHL0117383

US 7,921,356 B2

Page 8

- Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-166, Apr. 30, 2010.
- Exhibit A, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-30, Apr. 30, 2010.
- Exhibit B, in Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-15, Apr. 30, 2010.
- Exhibit C, Expert Report of Bruce Croft, in *Arendi Holding Ltd. v. Microsoft Corp.*, CA No. 09-119-JJF, from United States District Court for the District of Delaware, pp. 1-90, Apr. 30, 2010.
- "On Automating Hypertext Links in Knowledge-Based Decision Support Systems", Technical Report BCCS-90-08, pp. 1-22, Apr. 1990.
- Hypertext: Concepts, Systems and Applications: Proceedings of the First European Conference on Hypertext, INRIA, France, pp. 1-8, Nov. 1990.
- "Issues in Modeling a "Dynamic" Hypertext Interface", Technical Report BCCS-91-2, pp. 1-13, Apr. 1991.
- "Valuation Links: Extending the Computational Power of Hypertext", Technical Report BCCS-91-1, pp. 1-11, Apr. 1991.
- Template-Driven Hypertext: A Methodology for Integrating a Hypertext Interface into Information Systems, Technical Report BCCS-91-3, pp. 1-23, Jun. 1991.
- "On Generalizing the Concept of Hypertext", Technical report BCCS-91-5, pp. 1-11, Jun. 1991.
- "Bridge Laws in Hypertext a Logic Modeling Approach", Technical Report BCCS-91-4, pp. 1-30, Jun. 1991.
- "On Merging Hypertext Into Dynamic, Non-Hypertext Systems", Technical Report BCCS-91-14, pp. 1-24, Nov. 8, 1991.
- "Architectures for Volatile Hypertext", Hypertext '91 Proceedings, pp. 243-280, Dec. 1991.
- "Providing Information Systems With Full Hypermedia Functionality", pp. 1-15, Oct. 1992.
- Mail Address Change Notification, IBM Technical Disclosure Bulletin, vol. 36 No. 1, pp. 70-72, Jan. 1993.
- Towards CAAI: Computer Assisted Application Integration, Technical Report Series CS-93-17, University of Waterloo, pp. 1-23, Oct. 1993.
- Semiautomatic Generation of Dictionary Links in Hypertext, pp. 1-14, Feb. 1, 1995.
- "On Integrating Hypermedia into Decision Support and Other Information Systems", Decision Support Systems 14, pp. 251-267, 1995. Proceedings, Sixth Message Understanding Conference (MUC-6), pp. 1-64, Nov. 6-8, 1995.
- "A System to Read Names and Addresses on Tax Forms", Proceedings of the IEEE, vol. 84, No. 7, Jul. 1996.
- "Extraction and Integration of Data from Semi-structured Documents into Business Applications", Sloan WP#3979, CISL WP# 97-12, pp. 1-9, Sep. 1997.
- Borland, Russel, "Advance Self-Study Step by Step Microsoft Word 97 Advance Topics", Microsoft Press, 331 pages, 1997.
- Microsoft Word Language Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1133 pages, © 1997.
- Microsoft PowerPoint and Microsoft Outlook Language References, Official Technical Reference to Visual Basic for Applications, Microsoft Press, vol. 4, 539 pages, 1997.
- Microsoft Visual Basic for Applications and Shared Libraries References, vol. 5 of 5, Microsoft Press, 1005 pages, part No. 097-001685, © 1997.
- Microsoft Office 97 Resource Kit, Technical Information and Tools for the Support Professional, Microsoft Press, 1162 pages, © 1997.
- Microsoft Office97 Reviewer's Guide, Professional Edition, 546 pages, Aug. 1996.
- Running Microsoft Word for Windows 95, In Depth Reference and Inside Tips from the Software Experts, Russell Borland, Microsoft Press, 1040 pages, © 1995.
- Running Microsoft Office 97, Standard and Professional, Michael Halvorson and Michael Young, Microsoft Press, 1130 pages, © 1997.
- Microsoft Word 97, self-Study Kit, Covers Microsoft Word 97 for Windows 95 and Windows NT version 4.0, Microsoft Press, 314 pages, © 1997.
- Microsoft Outlook 97, Step by Step, Self-Study Kit, Microsoft Press, 359 pages, © 1997.
- VBA Developer's Handbook, Reusable Code, Visio Professional Demo Copy, and VB5CCE, 1997.
- Microsoft97/Visual Basic Programmer's Guide, Microsoft Press, 548 pages, © 1997.
- Microsoft Access Language Reference and Office Data Access Reference, Official Technical Reference to Visual Basic for Applications, Microsoft Press, 1287 pages, © 1997.
- Microsoft Office97 Step by Step, Visual Basic, Developer's Self-Study Guide, Microsoft Press, 359 pages, © 1997.
- Microsoft Office97 Developer's Handbook, Microsoft Press, Christine Solomon, 596 pages, © 1997.
- Microsoft Word Developer's Kit, Microsoft Professional Editions, Third Edition, Manufacturing Copy 1st, Microsoft Press, 1106 pages, © 1995.
- Inside Microsoft Word, vol. 4, No. 1, The Cobb Group, pp. 1-16, Jan. 1997.
- Inside Microsoft Word, vol. 4, No. 2, The Cobb Group, pp. 1-16, Feb. 1997.
- Inside Microsoft Word, vol. 4, No. 3, The Cobb Group, pp. 1-16, Mar. 1997.
- Inside Microsoft Word, vol. 4, No. 4, The Cobb Group, pp. 1-16, Apr. 1997.
- Inside Microsoft Word, vol. 4, No. 7, The Cobb Group, pp. 1-16, Jul. 1997.
- Inside Microsoft Word, vol. 5, No. 9, ZD Journals, pp. 1-16, Sep. 1998.
- Inside Microsoft Word, vol. 5, No. 10, ZD Journals, pp. 1-16, Oct. 1998.
- Inside Microsoft Word, vol. 5, No. 11, ZD Journals, pp. 1-16, Nov. 1998.
- Eudora Mail Pro, Version 3.0 for Windows, User Manual, QUALCOMM Incorporated, 198 pages, Jun. 1997.
- Eudora Mail Pro, Version 3.1 for Macintosh, User Manual, QUALCOMM Incorporated, 199 pages, Jun. 1997.
- Non-Confidential Redacted Jun. 11, 2010 Rebuttal Expert Report Richard Taylor, 311 pages, Jun. 11, 2010.
- Spell Checking for Microsoft Windows, Palantir Software, 36 pages, © 1986.
- Johannes, S., et al., "Active Messenger: Email Filtering and Mobile Delivery," *Massachusetts Institute of Technology*, Sep. 1999, pp. 1-122.
- Srihari, S, et al., "A System to Read Names and Addresses on Tax Forms," *Proceedings of the IEEE*, vol. 14, No. 7, Jul. 1996, pp. 1038-1049.
- Person et al., "Special Edition Using Microsoft Word 97" Que Publishing, pp. 475-514, Dec. 16, 1996.
- AddressMate for Windows, Version 2.0, Product Box and License Agreement, 24 pages, no date available.
- U.S. Appl. No. 12/963,744, filed Dec. 9, 2010.
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626 As Filed, 85 pages, Nov. 10, 1998.
- US Patent No. 6,323,853, Preliminary Amendment, U.S. Appl. No. 09/189,626, 20 pages, Apr. 29, 1999.
- US Patent No. 6,323,853, Office Action, U.S. Appl. No. 09/189,626, 9 pages, Apr. 26, 2000.
- US Patent No. 6,323,853, U.S. Appl. No. 09/189,626, Interview Summary, 2 pages, Jun. 14, 2000.
- US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Jul. 25, 2000.
- US Patent No. 6,323,853, Final Office Action, U.S. Appl. No. 09/189,626, 10 pages, Sep. 18, 2000.
- US Patent No. 6,323,853, Interview Summary, U.S. Appl. No. 09/189,626, 1 page, Oct. 17, 2000.
- US Patent No. 6,323,853, Response, U.S. Appl. No. 09/189,626, 5 pages, Dec. 18, 2000.
- US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 7 pages, Jan. 4, 2001.

US 7,921,356 B2

Page 9

US Patent No. 6,323,853, Notice of Allowance, U.S. Appl. No. 09/189,626, 2 pages, Oct. 2, 2001.

US Patent No. 7,496,854, Application As Filed, 100 pages, Aug. 6, 2001.

US Patent No. 7,496,854, Preliminary Amendment, U.S. Appl. No. 09/923,134, 7 pages, Dec. 17, 2001.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 10 pages, Oct. 15, 2002.

US Patent No. 7,496,854, Response, 4 pages, U.S. Appl. No. 09/923,134, Apr. 15, 2003.

US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 7 pages, Jun 27, 2003.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 6 pages, Aug. 21, 2003.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 4 pages, Nov. 19, 2003.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 5 pages, Feb. 10, 2004.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 4 pages, May 19, 2004.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 14 pages, Dec. 30, 2004.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 7 pages, Aug. 29, 2005.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 17 pages, Nov. 16, 2005.

US Patent No. 7,496,854, Final Office Action, U.S. Appl. No. 09/923,134, 10 pages, Feb. 6, 2006.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 22 pages, Aug. 4, 2006.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 8 pages, Oct. 18, 2006.

US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 3 pages, Jan. 23, 2007.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Apr. 18, 2007.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 8 pages, Jul. 13, 2007.

US Patent No. 7,496,854, Interview Summary, U.S. Appl. No. 09/923,134, 4 pages, Oct. 19, 2007.

US Patent No. 7,496,854, Office Action, U.S. Appl. No. 09/923,134, 9 pages, Nov. 9, 2007.

US Patent No. 7,496,854, Response, U.S. Appl. No. 09/923,134, 32 pages, Jan. 24, 2008.

US Patent No. 7,496,854, Notice of Allowance, U.S. Appl. No. 09/923,134, 7 pages, Jul. 2, 2008.

US Patent No. 7,496,854, Amendment Under Rule 312; U.S. Appl. No. 09/923,134, 27 pages, Jul. 8, 2008.

US Patent No. 7,496,854, Response to Amendment Under Rule 312, U.S. Appl. No. 09/923,134, 4 pages, Jan 21, 2009.

U.S. Appl. No. 12/182,048, Application as Filed, 47 pages, Jul. 29, 2008.

U.S. Appl. No. 12/182,048, Office Action, 34 pages, Oct. 28, 2010.

U.S. Appl. No. 12/182,048, Interview Summary, 4 pages, Dec. 1, 2010.

U.S. Appl. No. 12/182,048, Response, 36 pages, Dec. 8, 2010.

US Patent No. 7,272,604, U.S. Appl. No. 09/390,303 As Filed, 55 pages, Sep. 3, 1999.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 9 pages, May 8, 2002.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 7 pages, Aug. 8, 2002.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 11 pages, Sep. 27, 2002.

US Patent No. 7,272,604, Proposed Response, U.S. Appl. No. 09/390,303, 4 pages, Oct. 24, 2002.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 3 pages, Mar. 27, 2003.

US Patent No. 7,272,604, Final Office Action, U.S. Appl. No. 09/390,303, 9 pages, Apr. 21, 2003.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 16 pages, Oct. 21, 2003.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Nov. 17, 2003.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 6 pages, Apr. 6, 2004.

US Patent No. 7,272,604, Supplemental Notice of Allowance, U.S. Appl. No. 09/390,303, 2 pages, Mar. 7, 2006.

US Patent No. 7,272,604, Office Action, U.S. Appl. No. 09/390,303, 10 pages, Jun. 6, 2006.

US Patent No. 7,272,604, Response, U.S. Appl. No. 09/390,303, 19 pages, Dec. 5, 2006.

US Patent No. 7,272,604, Notice of Allowance, U.S. Appl. No. 09/390,303, 7 pages, Feb. 7, 2007.

U.S. Appl. No. 11/745,186, Application As Filed, 61 pages, May 7, 2007.

U.S. Appl. No. 11/745,186, Office Action, 8 pages, Mar. 6, 2008.

U.S. Appl. No. 11/745,186, Response, 18 pages, May 14, 2008.

U.S. Appl. No. 11/745,186, Office Action, 17 pages, Oct. 14, 2008.

U.S. Appl. No. 11/745,186, Response, 28 pages, Feb. 13, 2009.

U.S. Appl. No. 11/745,186, Final Office Action, 16 pages, Mar. 26, 2009.

U.S. Appl. No. 11/745,186, Interview Summary, 4 pages, May 7, 2009.

U.S. Appl. No. 11/745,186, Response, 22 pages, Jun. 2, 2009.

U.S. Appl. No. 11/745,186, Advisory Action, 3 pages, Jun. 16, 2009.

U.S. Appl. No. 11/745,186, Request for Continued Examination, 3 pages, Jun. 22, 2009.

U.S. Appl. No. 11/745,186, Office Action, 40 pages, Sep. 29, 2009.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Mar. 26, 2010.

U.S. Appl. No. 11/745,186, Response, 29 pages, Mar. 29, 2010.

U.S. Appl. No. 11/745,186, Final Office Action, 10 pages, Jun. 15, 2010.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Aug. 5, 2010.

U.S. Appl. No. 11/745,186, Response, 24 pages, Aug. 10, 2010.

U.S. Appl. No. 11/745,186, Advisory Action, 3 pages, Aug. 16, 2010.

U.S. Appl. No. 11/745,186, Response, 18 pages, Sep. 23, 2010.

U.S. Appl. No. 11/745,186, Interview Summary, 3 pages, Dec. 14, 2010.

U.S. Appl. No. 12/963,744, Application As Filed, 53 pages, Dec. 9, 2010.

U.S. Appl. No. 12/963,744, Accelerated Examination Support Document, 80 pages, Dec. 9, 2010.

* cited by examiner

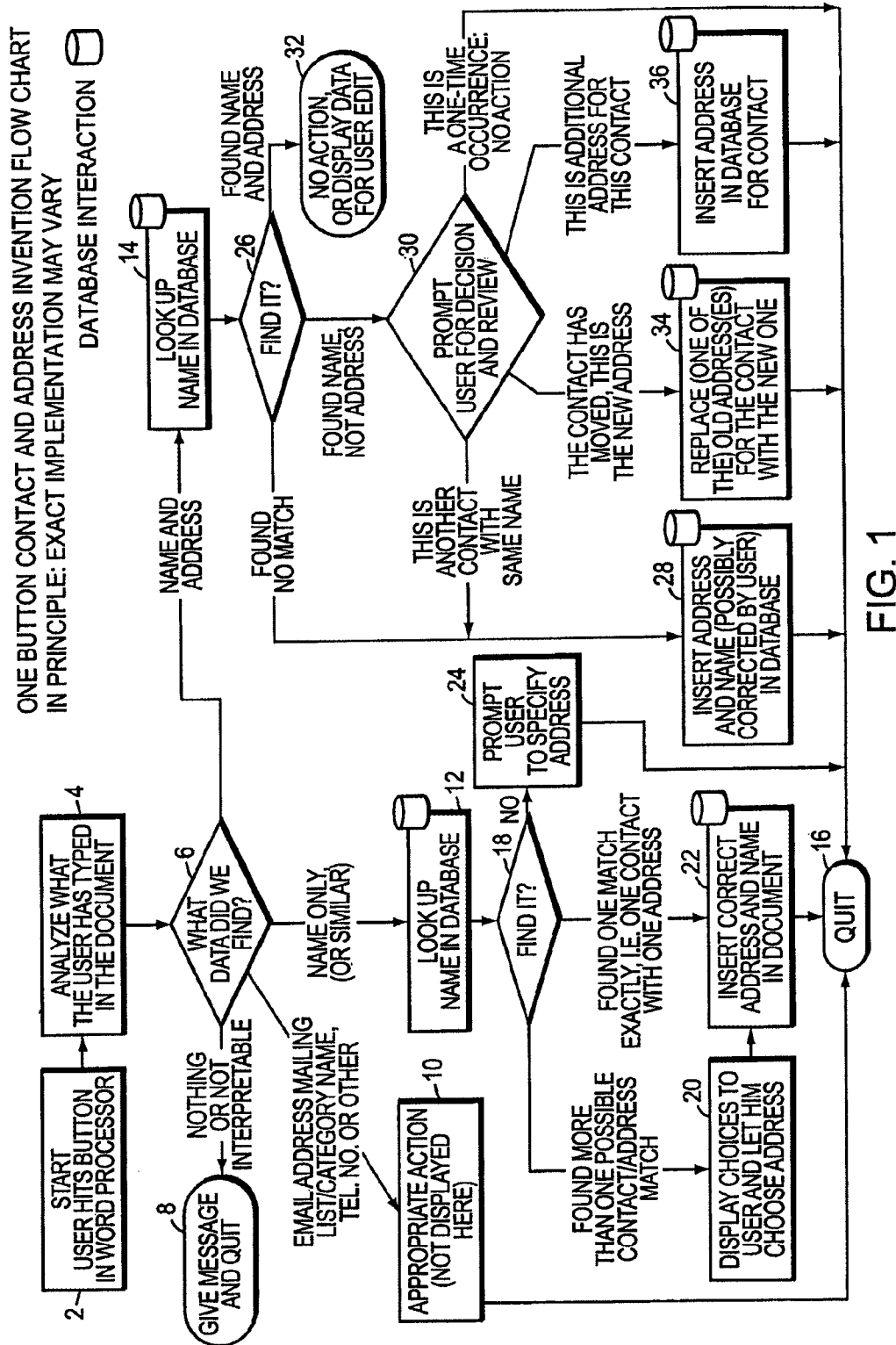


FIG. 1

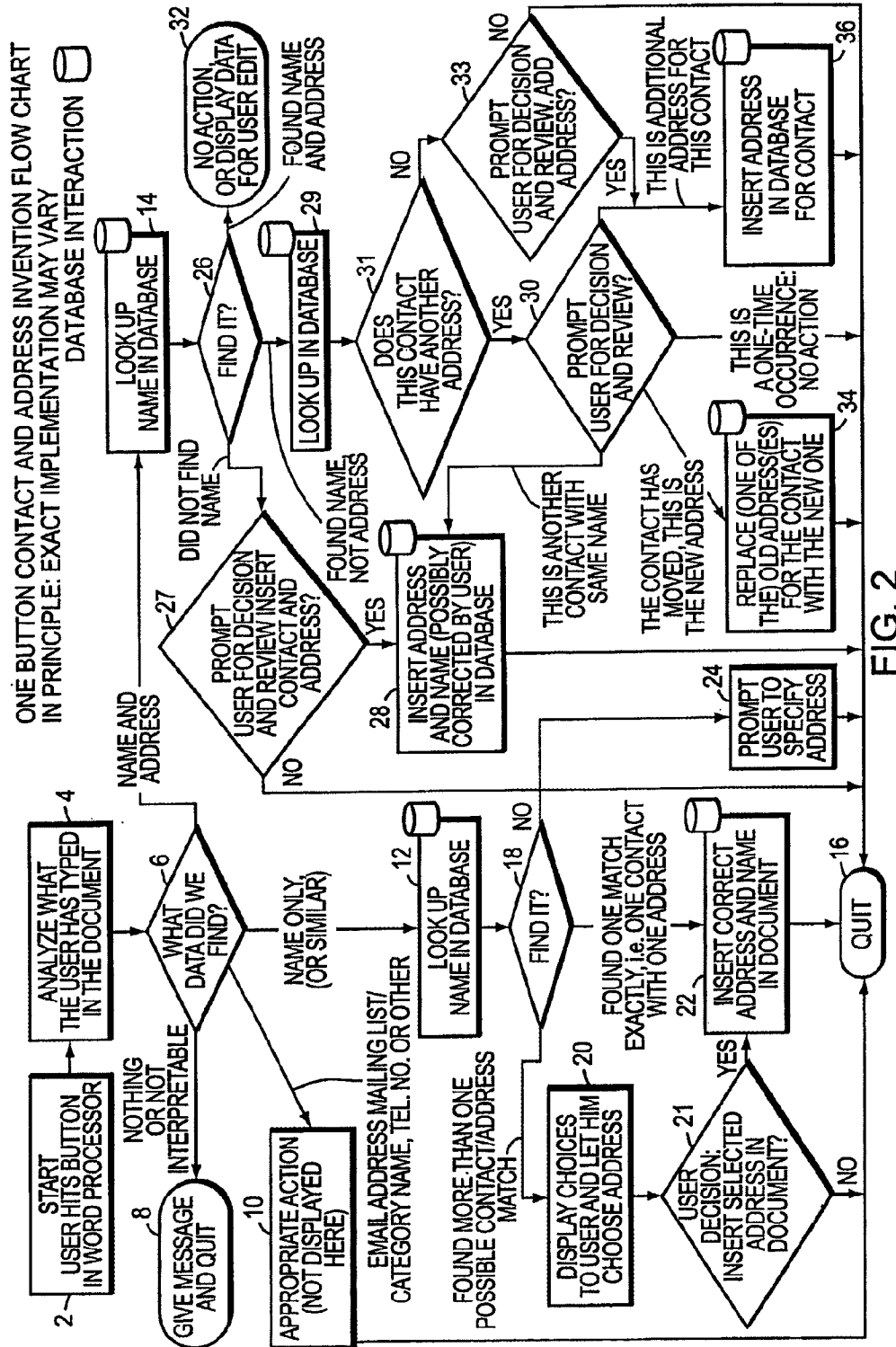


FIG. 2

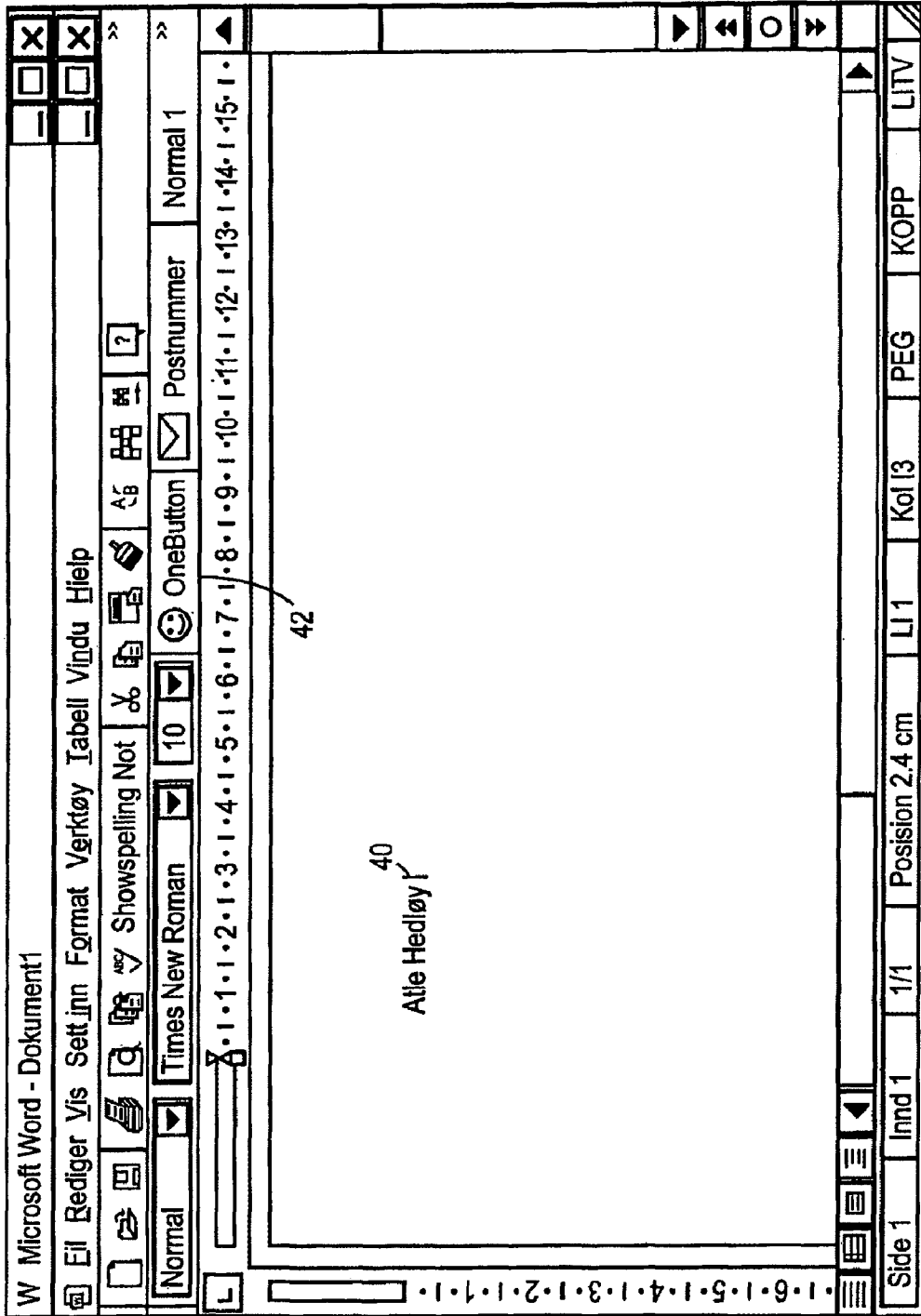


FIG. 3

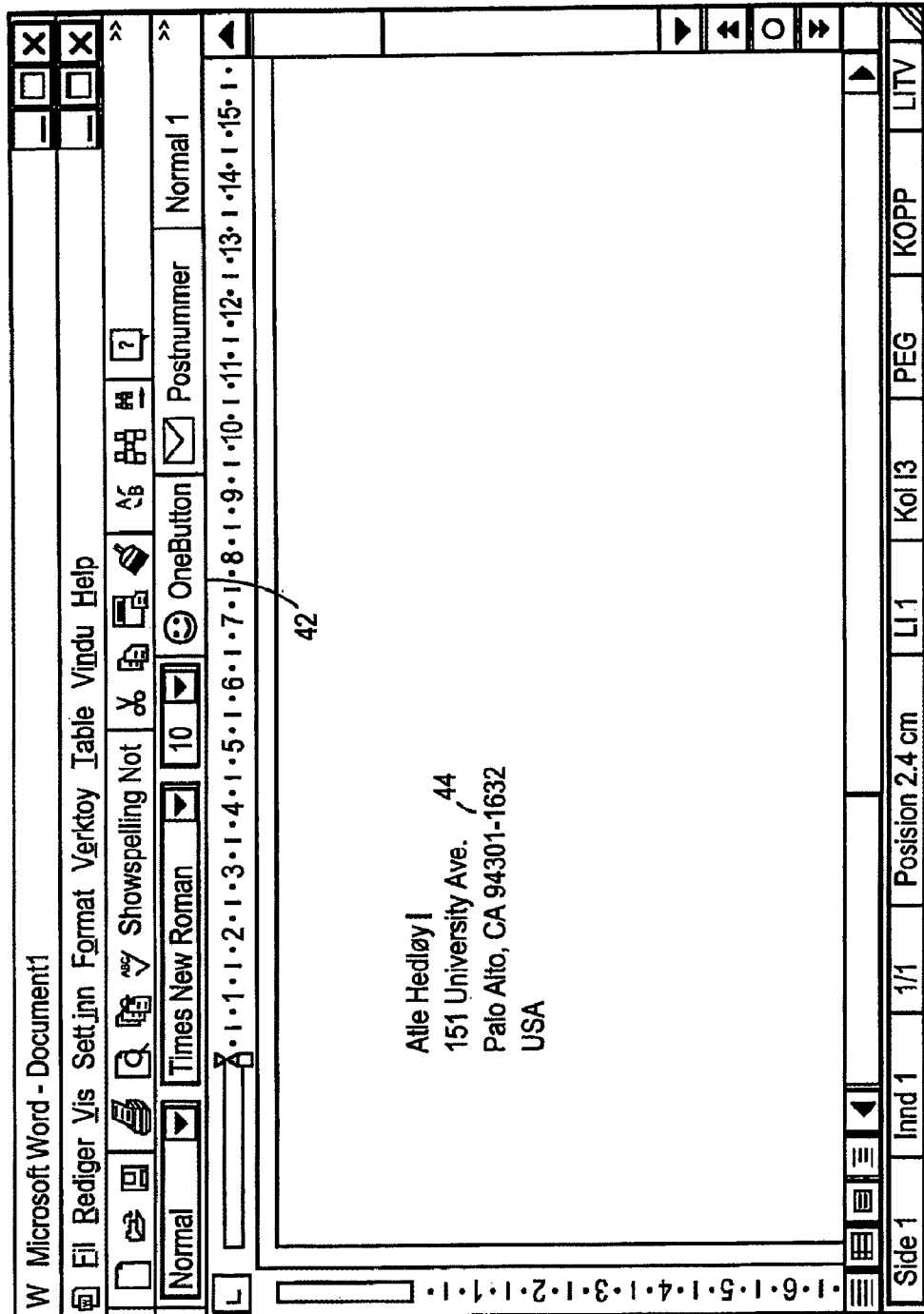


FIG. 4

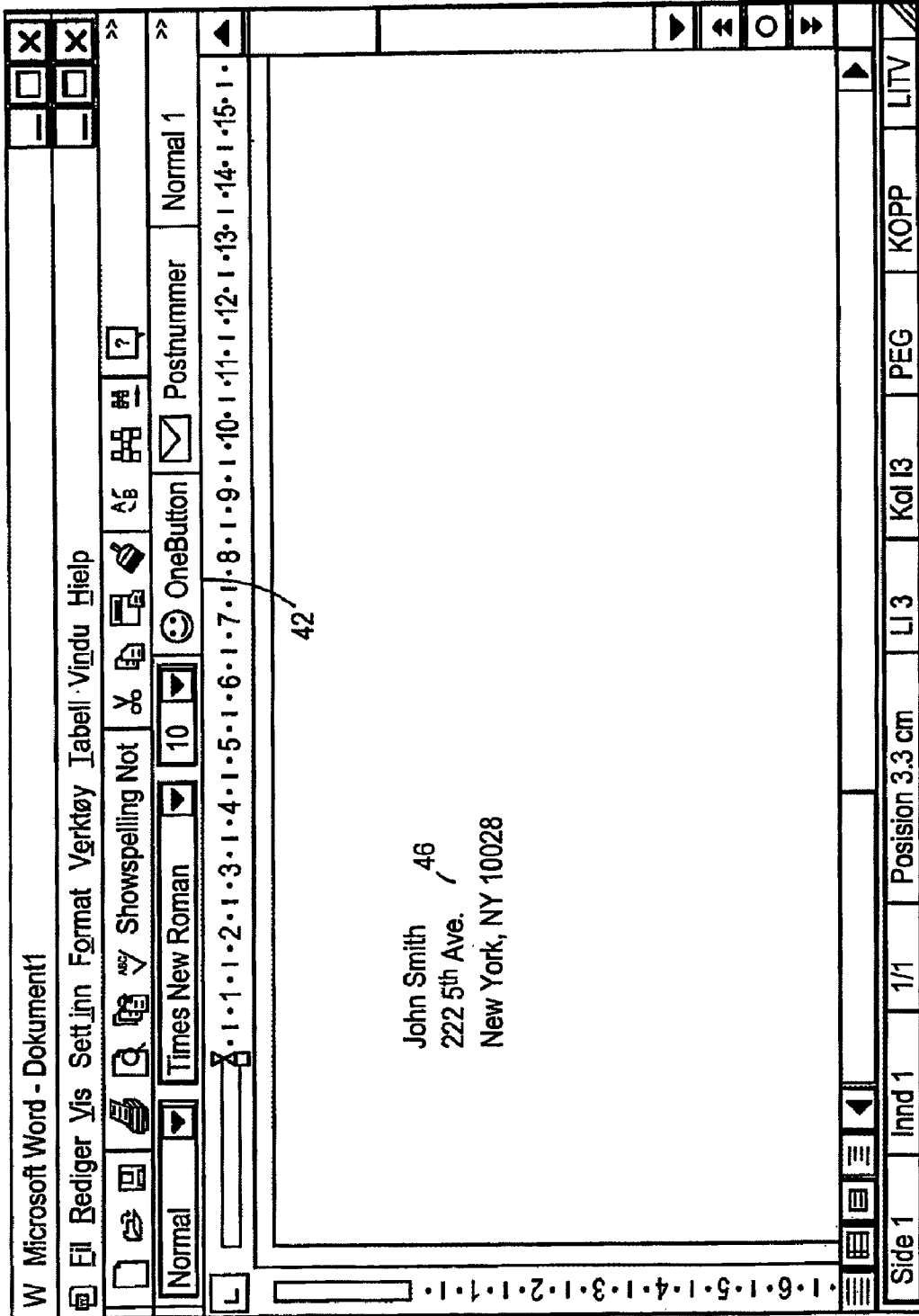


FIG. 5

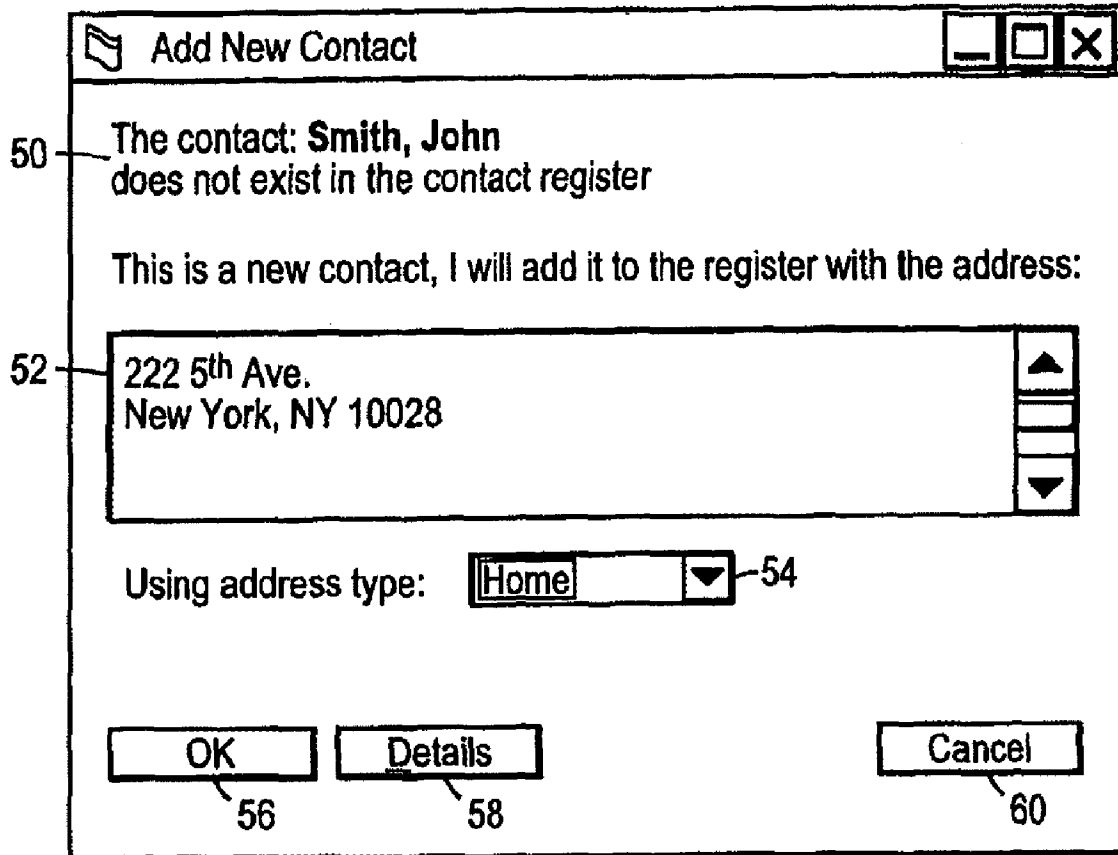


FIG. 6

The screenshot shows a software window titled "Arendi OneButton Contact Register". The window contains a form with the following fields and controls:

- Name:** A dropdown menu with a downward arrow.
- Title:** A text input field containing "John".
- First:** A text input field containing "Smith".
- Middle:** A text input field.
- Last:** A text input field.
- Suffix:** A dropdown menu with a downward arrow.
- Company:** A text input field.
- Address type:** A dropdown menu with "Home" selected.
- Street:** A text input field containing "222 5th Ave.".
- City:** A text input field containing "New York".
- State/Province:** A text input field containing "NY".
- ZIP/Postal:** A text input field containing "10028".
- Country:** A dropdown menu with a downward arrow.

At the bottom of the form area, there is a button labeled "Add and Choose" (64). To the right of the form area, there is a button labeled "Options..." (66) and a button labeled "Cancel" (60). At the bottom of the window, there is a status bar with the text "Dette er en test".

FIG. 7

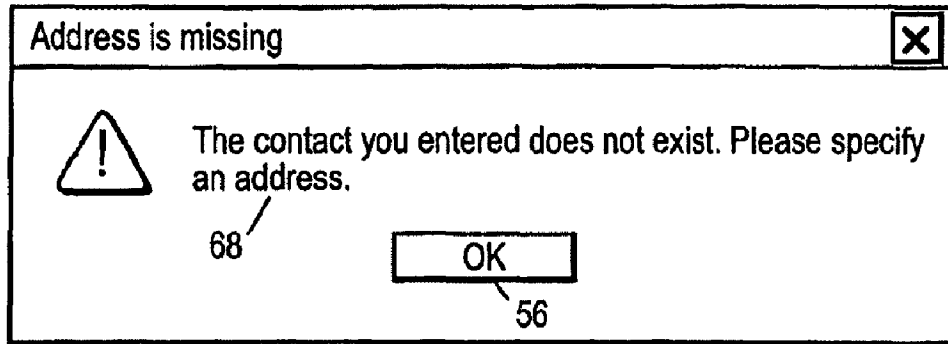


FIG. 8

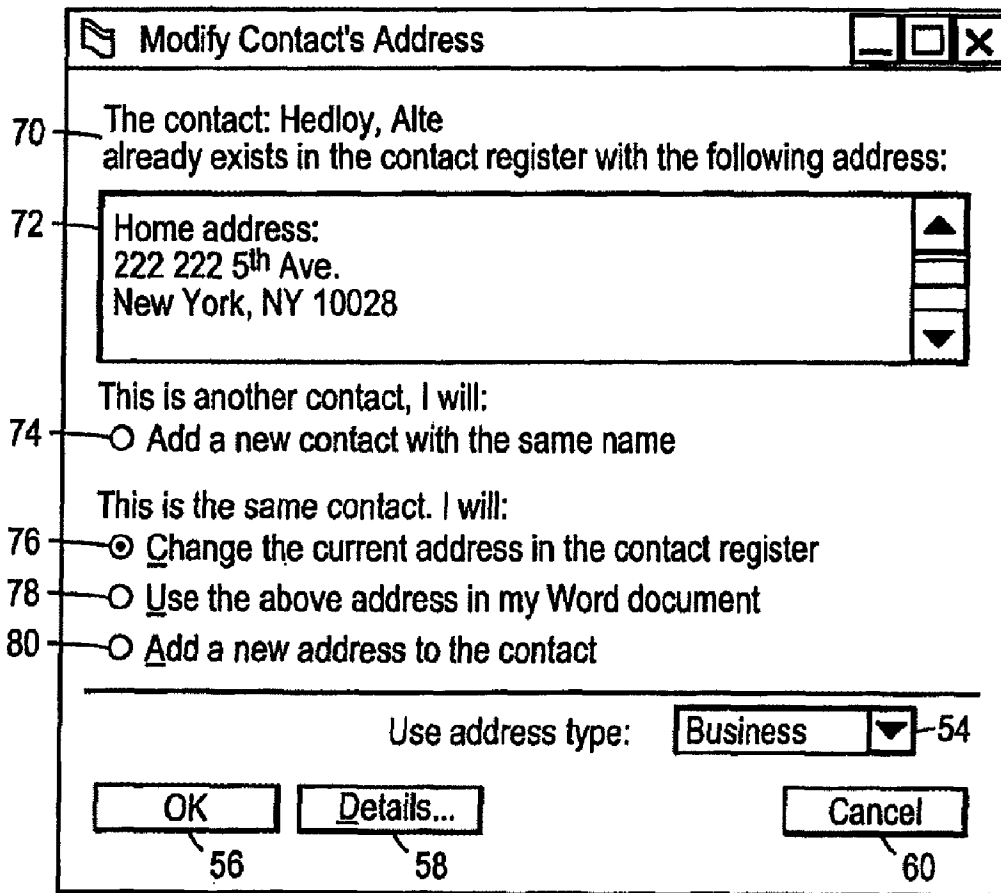


FIG. 9

The screenshot shows a window titled "Arendt OneButton Contact Register" with a close button (X) in the top right corner. Below the title bar, there is a text label "Existing addresses with the same name" followed by a list of addresses. The list has columns for Name, Address type, Address, Zip, City, and Country. The first entry is selected and highlighted in blue.

Name	Address type	Address	Zip	City	Country
[1] Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Below the list, there are two large text boxes. The left one is labeled "Name" and contains "Atle Hedløy". The right one is labeled "Address" and contains "113 Terrasse street", "New York, NY 12191-4292", and "United States of America".

To the right of the "Name" box is a button labeled "Choose" with reference numeral 86. To the right of the "Address" box is a button labeled "Full details..." with reference numeral 88. Below the "Address" box is a button labeled "More >>>" with reference numeral 90.

At the bottom left of the window is a button labeled "Options..." with reference numeral 66. At the bottom right is a button labeled "Cancel" with reference numeral 60. Below the "Options..." button is the text "Dette er en test".

FIG. 10

Arendi OneButton Contact Register

Existing addresses with the same name

Name	Address type	Address	Zip	City	Country
[1] Hedløy Atle	Business	113 Terrasse street	12191-4292	New York	United State of...
	Home	113 113 Jacob Aall street	12191-4292	New York	

Name: Atle Hedløy

Address: 113 Terrasse street
New York, NY 12191-4292
United States of America

Buttons: Choose (86), Full details... (88), More >>> (90)

Name: [Dropdown]

Title: [Dropdown]

First: Atle

Middle: [Dropdown]

Last: Hedløy

Suffix: [Dropdown]

Company: [Dropdown]

Buttons: Add and Choose (64), Options... (66)

Address type: [Dropdown] (54)

Street: 151 University Ave.

City: Palo Alto

State/Province: CA

ZIP/Postal: 94301-1632

Country: USA

Buttons: Add this address to the selected options above (92), Cancel (60)

Footer: Dette er en test

FIG. 11

The image shows a screenshot of a contact management application window titled "Atle Hedløy - Contact". The window has a menu bar with "File", "Edit", "View", "Insert", "Format", "Tools", "Contact", and "Help". Below the menu bar is a toolbar with icons for "Save and Close", "Print", "Copy", "Paste", "Find", "Home", "Back", "Forward", and "Help". The main content area is divided into several sections:

- General** (selected):
 - Full Name...:** Atle Hedløy
 - Company:** Hedløy, Atle
 - Address...:** 113 Terrasse street, New York, NY 12191-4292, United States of America
 - Business:** (dropdown menu)
 - Phone:** (dropdown menu)
 - Business:** (dropdown menu)
 - Home:** (dropdown menu)
 - Business Fax:** (dropdown menu)
 - Mobile:** (dropdown menu)
 - This is the mailing address
 - E-mail:** (dropdown menu)
 - Web page:** (text field)
- Details**: (tab)
- Journal**: (tab)
- All Fields**: (tab)
- Job title:** (text field)
- File as:** (text field)
- Categories...:** (text field)
- Private

FIG. 12

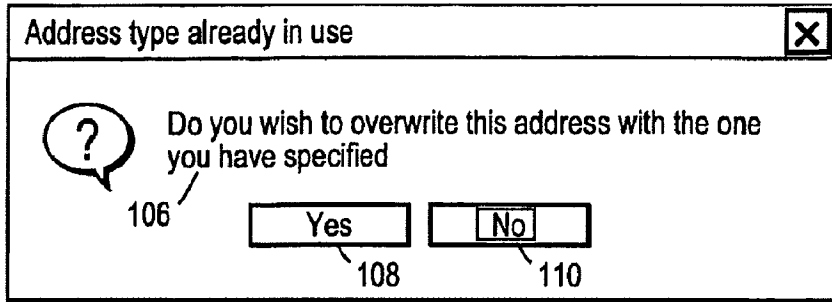


FIG. 13

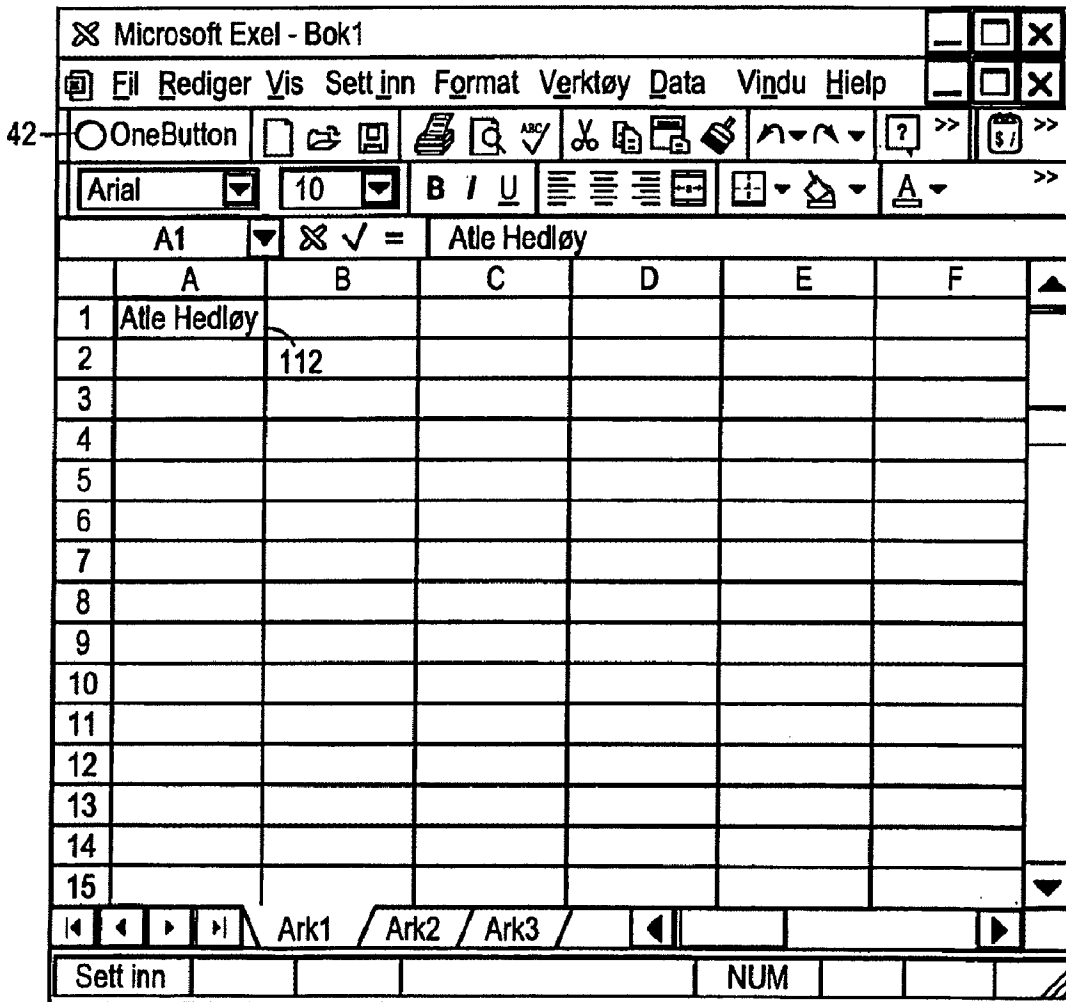


FIG. 14

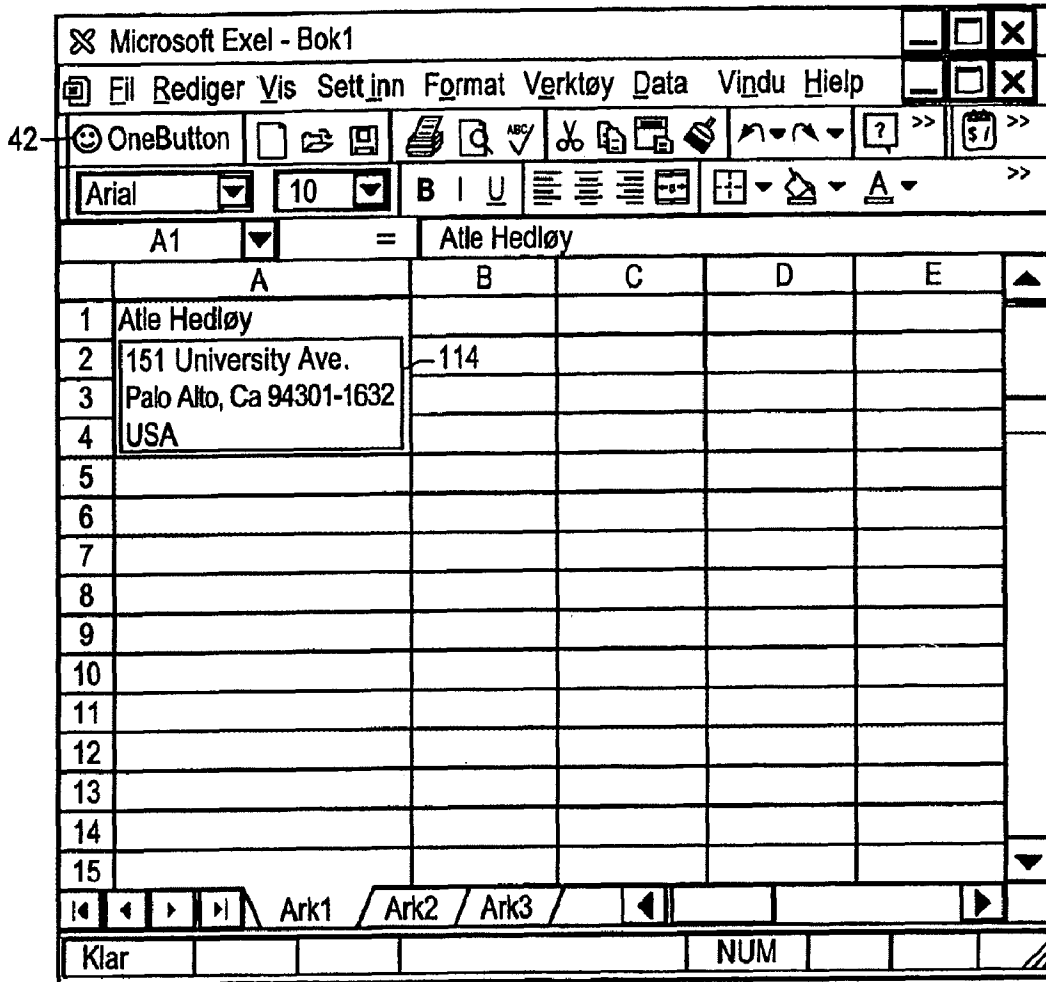


FIG. 15

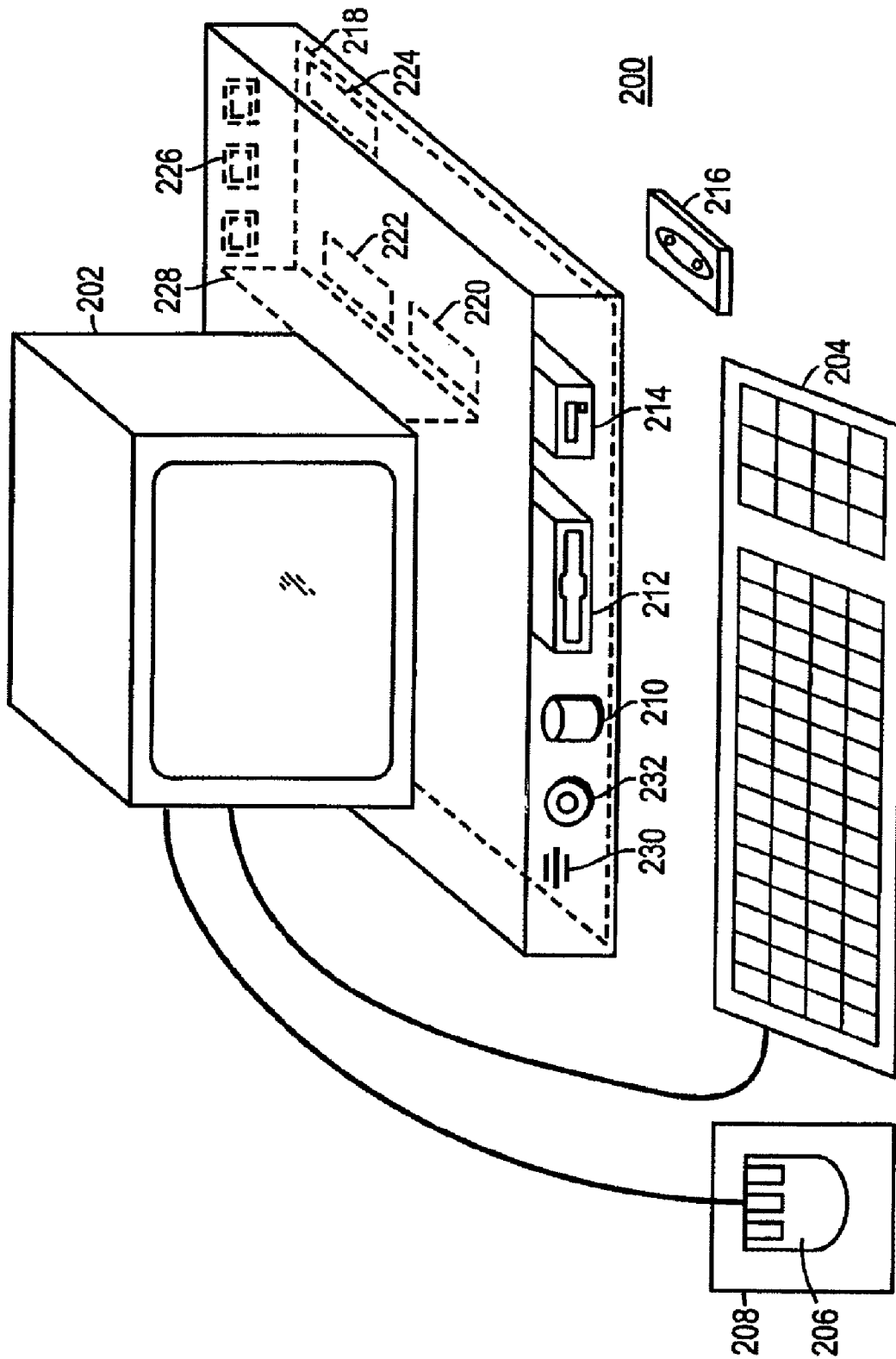


FIG. 16

US 7,921,356 B2

1

**METHOD, SYSTEM AND COMPUTER
READABLE MEDIUM FOR ADDRESSING
HANDLING FROM A COMPUTER PROGRAM**

**CROSS REFERENCES TO RELATED
APPLICATIONS**

The present application is a continuation of U.S. application Ser. No. 12/182,048 filed on Jul. 29, 2008, which in turn is a continuation of U.S. application Ser. No. 09/923,134 filed on Aug. 6, 2001 and issued as U.S. Pat. No. 7,496,854, which in turn is a continuation of U.S. application Ser. No. 09/189,626 filed on Nov. 10, 1998 and issued as U.S. Pat. No. 6,323,853, from which the present application claims priority. Each of the above-described applications is hereby incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

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, spread-sheets, 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™, QUATROPROT™, AMIPRO™, etc., and examples of such information management sources are ACCESS™, OUTLOOK™, ORACLE™, DBASE™, RBASE™, CARDFILE™, etc.

However, the information in the database must constantly be updated by the user. This requires the user to learn how to use and have access to the database. In this case, a change in the information, such as change in address or a name, etc., requires the user of the word processor to implement this change in the database, or alternatively, the change is made to the database centrally by a database administrator.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device provided in the computer program.

2

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device, such as a touch screen, keyboard button, icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source.

Another object of the present invention is to provide a method, system and computer readable medium for address handling within a computer program, such as a word processing program, spreadsheet program, etc., using an input device such as a touch screen, keyboard button icon, menu, voice command device, etc., provided in the computer program and coupled to an information management source, such as a database program, contact management program, etc.

The above and other objects are achieved according to the present invention by providing a novel method, system and computer readable medium for providing a function item, such as a key, button, icon, or menu, tied to a user operation in a computer, whereby a single click on the function item in a window or program on a computer screen, or one single selection in a menu in a program, 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. The click on the function item initiates a program connected to the button to search a database or file available on or through the computer, containing the person, company or address related data, in order to look up data corresponding to what the user types, or partly typed, e.g., name and/or address in the word processor, the correct data from the database, data related to the typed data, e.g., the name of the person, company, or the traditional or electronic address, or other person, or company, or address related data, and alternatively the persons, companies, or addresses, are displayed and possibly entered into the word processor, if such related data exists.

The present invention also includes a computer readable medium storing program instructions by which the method of the invention can be performed when the stored program instructions are appropriately loaded into a computer, and a system for implementing the method of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

FIG. 1 is a flow chart illustrating a method for address handling within a computer program, according to an exemplary embodiment of the present invention;

FIG. 2 is a flow chart illustrating a method for address handling within a computer program, according to another exemplary embodiment of the present invention;

FIG. 3 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 4 is a screen shot illustrating a retrieved address in a word processor, according to an exemplary embodiment of the present invention;

FIG. 5 is a screen shot illustrating the inputting of a name and address to be searched and an address handling button within a word processor, according to an exemplary embodiment of the present invention;

FIG. 6 is a screen shot illustrating an add new contact message window, according to an exemplary embodiment of the present invention;

US 7,921,356 B2

3

FIG. 7 is a screen shot illustrating a contact register message window, according to an exemplary embodiment of the present invention;

FIG. 8 is a screen shot illustrating an address missing message window, according to an exemplary embodiment of the present invention;

FIG. 9 is a screen shot illustrating a modify contact's address message window, according to an exemplary embodiment of the present invention;

FIG. 10 is a screen shot illustrating a select a contact address register message window, according to an exemplary embodiment of the present invention;

FIG. 11 is a screen shot illustrating a more detailed mode of registering an additional address for the contact register of FIG. 9, according to an exemplary embodiment of the present invention;

FIG. 12 is a screen shot illustrating a contact management program window in a full detailed mode, according to an exemplary embodiment of the present invention;

FIG. 13 is a screen shot illustrating an address already in use message window, according to an exemplary embodiment of the present invention;

FIG. 14 is a screen shot illustrating the inputting of a name to be searched and an address handling button within a spreadsheet, according to an exemplary embodiment of the present invention;

FIG. 15 is a screen shot illustrating a retrieved address in a spreadsheet, according to an exemplary embodiment of the present invention; and

FIG. 16 is a schematic illustration of a general purpose computer for performing the processes of the present invention, according to an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

In an embodiment of the present invention, single button addressing is achieved by providing an input device, such as a touch screen, keyboard, icon, menu, voice command device, etc. (hereinafter called "button"), in a computer program, such as a word processing program, spreadsheet program, etc. (hereinafter called "word processor"), for executing address handling therein.

Accordingly, 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, such as a touch screen button, keyboard button, icon, menu choice, voice command device, etc. A program then executes and retrieves the typed information from the document, and searches an information management source, such as a database, file, database program, contact management program, etc. (hereinafter called "database") to determine if the information, such as the name or part of the name typed and searched by the program exists in the database. If the program does not find stored information, such as a name, corresponding to the name or part of the name typed, the user is asked by the program whether the information, such as the name that was not found, should be added to the database. In addition, the user may enter any other information besides the name, such as addresses, businesses, telephone numbers, fax numbers, e-mail address, etc., so that this other information can be stored in the database for later use.

If the program finds name(s) and address(es) corresponding to the part of the addressee's name typed, this additional

4

information is automatically entered into the user's word processor, optionally with a confirmation from the user that this is the correct data. If the typed address information does not correspond to data already stored in the database, after clicking on the button, the program, for example, lets the user decide: (1) if this is new data (e.g., a new address) for an existing contact; (2) if the stored data should be changed to what the user just typed; (3) if this is a new contact with the same name as the one already entered into the database; or (4) if the typed address is only to be used once, and therefore not to be stored in the database at all. If, later, for example, a name with several address stored in the database is recalled, all addresses for this contact will be displayed, so that the correct address can be selected by the user.

The program may be extended to also store and retrieve other information, such as telephone numbers, fax numbers, e-mail addresses, etc. Once the program recalls the telephone numbers, fax numbers, email addresses, etc., the user can command the program to send e-mails, faxes, etc. Similarly, if the user types in the name of a mailing list, the program create merge letters, group emails, etc.

Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views, and more particularly to FIGS. 1 and 2 thereof, there is illustrated flow charts of single button addressing, according to exemplary embodiments of the present invention.

In FIG. 1, after the user has inserted the address in the word processor, the user commands the button at step 2 and the program analyzes what the user has typed in the document at step 4. At step 6, the program decides what was found in the document and if the program found nothing in the document or what it found was un-interpretable the program goes to step 8 and outputs an appropriate message to the user and then quits at step 16. The program analyzes what the user has typed in the document at step 4, for example, by analyzing (i) paragraph/line separations/formatting, etc.; (ii) street, avenue, drive, lane, boulevard, city, state, zip code, country designators and abbreviations, etc.; (iii) Mr., Mrs., Sir, Madam, Jr., Sr. designators and abbreviations, etc.; (iv) Inc., Ltd., P.C., L.L.C, designators and abbreviations, etc.; and (v) a database of common male/female names, etc.

If the program find an e-mail address mailing list/category name telephone number or other information, at step 10 an appropriate action is performed by the program and then the program execution quits at step 16. If the program only finds a name or initials, or the like, the program looks up the name in the database at step 12 and at step 18 the program determines what was found. If the program finds more than one possible contact/address match, at step 20 the program displays menu choices to the user to let him choose an appropriate answer. Then at step 22 the program inserts a correct address and name in the document and then at step 16 the program quits execution. If the program finds one match exactly, i.e., one contact with one address, the program inserts the correct address and name in the document at step 22 then quits and then quits execution at step 16. If the program does not find a name in the database, at step 24 the program prompts the user to specify an address and then quits execution at step 16. If the program at step 6 finds a name and an address, at step 14 the name is looked up in the database. Then, at step 26, if no match is found, at step 28 the program inserts an address and a name which are possibly corrected by the user into the database and then quits execution at step 16. If at step 26, the name and address is found, at step 32 the program either takes no action or displays the data for the user to edit. If at step 26, the name is found but not the address, the

US 7,921,356 B2

5

program prompts the user for a decision at step 30. If the user decides that this is another contact with a same name, the program goes to step 28. If the user decides that this is a one time occurrence, no action is taken and the program quits at step 16. If the user decides that the contact has, for example, moved and that this is a new address, at step 34 one of the old addresses for the contact is replaced with the new one and the program quits at step 16. If the user decides that this is an additional address for the contact, at step 36 the additional address is inserted into the database for that contact and execution quits at step 16.

The flowchart shown in FIG. 2 is similar to the flowchart in FIG. 1, except for some additional steps which will now be discussed. At step 6, if the program only finds a name or a similar name then the name is looked up in the database at step 12, then at step 18 if the program found more than one possible contact/address match, the program displays choices to the user to let him choose an address at step 20. Then at step 21 the user decides whether to insert the selected address into the document. If the user does not decide to select the address into the document the program quits execution at step 16. If the user decides to insert the selected address into the document the program inserts the address and name into the document at step 22 and then quits at step 16.

If the program finds a name and address in the database at step 6, then at step 14 the program looks up the name in the database and at step 26 the program determines what it has found. If the program does not find the name at step 26, at step 27 the program prompts the user for a decision and review and whether to insert the contact and address. If the user does not decide to insert the contact address, the program quits at step 16. If the user decides to insert the contact address, at step 28 the program inserts the address and name which may be possibly corrected by the user for program in the database then execution quits at step 16.

If at step 26 the program finds a name and not an address, then at step 29 the name is looked up in the database. Then at step 31 the program decides whether this contact has another address. If the contact does not have another address, at step 33 the program prompts the user for a decision and review and whether to add the address. If the user does not want to add the address at step 33, the program quits at step 16. If the user wants to add the address at step 33 because this is an additional address for the contact, at step 36 the address is inserted in the database for the contact and execution quits at step 16.

At step 30, if the user decides that this is another contact with the same name, then the program goes to step 28. If at step 30 the user decides that this is a one time occurrence, then the program quits at step 16. If at step 30, the user decides that the contact has, for example, moved, the program goes to step 34. If at step 30, the user decides that this is an additional address for the contact, at step 36 the program inserts the address in the database for the contact and then quits at step 16.

Various exemplary screen shots which are generated during execution of the program, according to the present invention, will now be described with reference to FIGS. 3-15 and examples 1-7 as follows.

Example 1

Retrieving an Existing Address from the Database

FIG. 3 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name 40. The user hits the button 42, for example, marked "OneButton" and the program according to the present inven-

6

tion retrieves is the name 40 from the document, searches a database for the name 40, and inserts the retrieved address 44 associated with the name 40 into the document as shown in, for example, FIG. 4.

The above example corresponds to steps 2, 4, 6, 12, 18, 22 and 16 in the flow charts of FIGS. 1 and 2.

Example 2

Adding a New Contact to the Database

FIG. 5 illustrates a starting point in word processor document such as WORD document, wherein the user has typed a name and address of a new contact 46. The user commands the button 42, for example, marked "OneButton," and the program according to the invention retrieves the new contact 46 from the document, searches a database for the name of the new contact 46 and generates a screen as shown in, for example, FIG. 6. This screen includes a message 50 informing the user that the new contact does not exist in the database, a message 52 including the address retrieved from the document, an address type selection 54, such as home, business, etc., and "OK," "Details," and "Cancel" buttons 56, 58, and 60, respectively.

At this point, the user can cancel the operation by commanding the Cancel button 60, ask the program to store data in the database and return the document by commanding the OK button 56, or check details before storing data into the database by commanding the Details button 58. If the user commands the Details button 58, as shown in, for example, FIG. 7, a message screen is provided so that the user can review and edit data 62 and the selection 54, store the data 62 and 54 in the database by commanding a "Add and Choose" button 64, see more options by commanding an "Options" button 66, or cancel is the operation by commanding the Cancel button 60.

The above example corresponds to steps 2, 4, 6, 14, 26, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 14, 26, 27, 28 and 16 in the flow chart of FIG. 1 and steps 2, 4, 6, 26, 27, 28 and 16 in the flow chart of FIG. 2.

Example 3

Try to Retrieve Existing Address, but Contact is not in Database

FIG. 3 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name of a contact 40. The user commands the button 42, for example, marked "OneButton," and the program according to the present invention retrieves the name 40 from the document, searches a database for the name of the contact 40 and generates a screen as shown in, for example, FIG. 8. This screen includes a message 68 informing the user that the contact does not exist in the database and to specify an address, and "OK" buttons 56. At this point when the user commands the OK button 56, the user returns to the document so that the contact's address can be included as in Example 2 above.

The above example corresponds to steps 2, 4, 6, 12, 18, 24, and 16 in the flow of charts of FIGS. 1 and 2.

Example 4

Adding a New Address for an Existing Contact

Short Version

FIG. 4 illustrates a starting point in word processor document, such as WORD document, wherein the user has typed

US 7,921,356 B2

7

a name and new address of an existing contact **44**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **44** from the document, searches a database for the same name of the existing contact **44** and generates a screen as shown in, for example, FIG. **9**. This screen includes a message **70** informing the user that the contact already exists in the database with an existing address, a message **72** including the existing address, add new contact with same name selection **74**, change existing address selection **76**, use existing address in document selection **78**, add the new address contact selection **80**, the address type selection **54**, such as home, business, etc., and the "OK," "Details," and "Cancel" buttons **56**, **58** and **60** respectively. At this point, the user may select one of the four options **74-80**, and command the OK button **56** to execute the selected options. The user can also cancel the operation by commanding the Cancel button **60**, or check details before storing data into the database by commanding the Details button **58**.

The above example corresponds to steps **2, 4, 6, 14, 26, 28, 30, 34, 36**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 14, 26, 29, 31, 30, 28, 34, 36**, and **16** in the flow chart of FIG. **2**.

Example 5

Selecting Between Several Possible Matching Addresses

FIG. **3** illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and possibly address of at least one existing contact **40**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **40** from the document, searches a database for the name of the existing contact **40** and generates a screen as shown in, for example, FIG. **10**. This screen includes a message informing the user the name corresponds to several addresses and possible contacts which already exist in the database, with existing contacts and addresses for selection **82**, a message **84** including the full name and address for the contact that the user selects in **82**, the Options button **66**, a "Choose" button **86**, a "Full details" button **88**, a "More>>>" button **90**, and the Cancel button **60**. The above screen indicates to the user that at least one contact with the same name exists, and that there are more than one addresses and/or contacts that match.

At this point the user may command the Choose button **86** to use the selected address and return to the document, or the user may command the More>>> button **90** to view how the program interpreted what the user typed in the word processor, and possibly change this data, wherein the program generates an updated screen as shown in, for example, FIG. **11**. The updated screen includes the data **62** which displays the name typed in the word processor as interpreted by the program, address fields, and the fields for the address type selection **54**, such as home, business, etc., which may be changed by the user before the program stores it in the database, the Add and Choose button **64**, a "<<< Less" button **90** corresponding to the More>>> button **90** for returning to the screen of FIG. **10**, and an "Add this address to the selected contact above" button **92**. The user might then command the Add this address to the selected contact above button **92** and the result in the word processor is illustrated in FIG. **4**. The user can also cancel the operations by commanding the Cancel button **60**, or command the add choose button **64** to add this name and address as a new contact and address, or open the database

8

before storing data into the database by commanding a "Full details" button **88** as will be later described.

The above example corresponds to steps **2, 4, 6, 12, 18, 20, 22**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 12, 18, 20, 21, 22**, and **16** in the flow chart of FIG. **2**.

Example 6

Adding a New Address for an Existing Contact

Long Version

FIG. **4** illustrates a starting point in word processor document, such as WORD document, wherein the user has typed a name and new address of an existing contact **44**. The user commands the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the existing contact **44** from the document, searches a database for the name of the existing contact **44** and generates a screen as shown in, for example, FIG. **9**. As previously described the screen includes a message **70** informing the user that the contact already exists in the database with an existing address, and the user may command the Details button **58** to see the details of the new address for potentially modify the details before they are stored in the database and the program generates a screen as shown in, for example, FIG. **10**. From this screen, the user may choose to use another address than the one he typed, and return to the document, or the user may command the "Full details" button **88** to enter a database program, such as OUTLOOK™, directly as shown in, for example, FIG. **12**. In FIG. **12**, the database program, such as OUTLOOK™, may include portions **94-104** for allowing the user to modify various pieces of data before they are stored in the database.

Alternatively, in the screen shown in FIG. **10**, the user may command the More>>> button **90** at which time the program generates the screen as shown in, for example, FIG. **11** and as previously described. In this screen, the user might then command the Add this address to the selected contact above button **92**. If the address typed is already in use, the program generates a screen including a message **106**, and "Yes" and "No" buttons, **108** and **110**, respectively, as shown in, for example, FIG. **13**. If the user hits the Yes button **108** the program overwrites the contact address with the address specified by the user (e.g., if the contact has moved) and the result in the word processor is shown in, for example FIG. **4**.

The above example corresponds to steps **2, 4, 6, 12, 14, 26, 28, 30, 34, 36**, and **16** in the flow chart of FIG. **1** and steps **2, 4, 6, 12, 14, 26, 29, 31, 30, 28, 34, 36**, and **16** in the flow chart of FIG. **2**.

Example 7

Spreadsheet Application

FIG. **14** illustrates a starting point in word processor document, such as an EXCEL spreadsheet, wherein the user has typed a name **112**. The user hits the button **42**, for example, marked "OneButton," and the program according to the present invention retrieves the name **112** from the spreadsheet, searches a database for the name **112**, and inserts the retrieved address **114** into the spreadsheet as shown in, for example, FIG. **15**. Accordingly, the examples 1-6 apply not only to word processor documents, such as WORD™ documents, etc., but to other word processor documents, and spreadsheets, such as EXCEL™ spreadsheets, etc.

The above example corresponds to steps 2, 4, 12, 18, 22, and 16 in the flow of charts FIGS. 1 and 2.

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 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.

As stated above, the system includes at least one computer readable medium or alternatively, the computer readable medium may be accessed through various paths, such as networks, internet, drives, etc. Examples of computer readable media are compact discs, hard disks, floppy disks, tape, magneto-optical disks, PROMs (EPROM, EEPROM, Flash, EPROM), DRAM, SRAM, SDRAM, etc. Stored on any one or on a combination of computer readable media, the present invention includes software for controlling both the hardware of the computer 200 and for enabling the computer 200 to interact with a human user. Such software may include, but is not limited to, device drivers, operating systems and user applications, such as development tools. Such computer readable media further includes the computer program product of the present invention for performing any of the processes according to the present invention, described above (see, e.g., FIGS. 1-15). The computer code devices of the present invention can be any interpreted or executable code mechanism, including but not limited to scripts, interpreters, dynamic link libraries, Java classes, and complete executable programs, etc.

The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

Address handling, according to this invention, is a significant simplification relative to existing methods, and requires little or no training on the part of a user, as correct addresses are retrieved with a minimal number of user commands, "clicks," keystrokes, etc. In addition, a program according to the present invention, can be programmed and created in most existing programming languages and be connected to most modern word processors. Therefore, according to the present invention, the process of creating and updating records in an address database is significantly simplified, since this may now be performed directly from the word processor.

Although the present invention is defined in terms of word processing documents, such as WORD™ documents and EXCEL™ spreadsheets, the present invention is applicable to all types of word processing documents such as NOTE-

PAD™, WORDPAD™, WORDPERFECT™, QUATRO-PROT™, AMIPRO™, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of information management or database programs, such as OUTLOOK™, etc., the present invention is applicable to all types of information management or database programs such as ACCESS™, ORACLE™, DBASE™, RBASE™, CARD-FILE™, including "flat files," etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of providing an input device, such as a button 42 in a word processor for address handling therein, the present invention may be practiced with all types of input devices, such as touch screen, keyboard button, icon, menu, voice command device, etc., as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving information from a document before searching a database, the user may select the information in the document to be searched by the program in the database (e.g., by highlighting, selecting, italicizing, underlining, etc.), as will be readily apparent to those skilled in the art.

Although the present invention is defined in terms of a program retrieving a name or portion thereof from a document before searching a database the program may retrieve an address or portion thereof from the document before searching the database and insert, correct, complete, etc., the retrieved address based on the information found in the database corresponding to the retrieved address or portion thereof, as will be readily apparent to those skilled in the art.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of appended claims, the invention may be practiced otherwise than as specifically described herein.

This application claims priority and contains subject matter related to Norwegian patent application No. 984066 filed on Sep. 3, 1998, the entire contents of which are hereby incorporated by reference.

What is claimed is:

1. At least one non-transitory computer readable medium encoded with instructions which, when loaded on a computer, establish processes for contact information handling, implemented by a document editing program running in the computer, the processes comprising:

- allowing a user to enter textual information into a document using the document editing program;
- displaying the textual information in the document electronically using the document editing program;
- allowing, in the document editing program, the user to select in the document at least a portion of the textual information while the textual information is displayed;
- following user selection of textual information in the document, analyzing, by the document editing program, the selected textual information to determine if the selected textual information is regarded by the document editing program as contact information and what type or types of contact information the selected textual information is;

- providing an input device configured by the document editing program to allow the user to initiate an operation, such operation being of a type depending at least in part on the type or types of contact information of the selected textual information, the operation comprising identifying at least part of the selected textual information to use as a search term in order to find second

US 7,921,356 B2

11

information, of a specific type or types, associated with the search term in an information source external to the document;

after identifying at least part of the selected information to use as a search term, and in consequence of receipt by the document editing program of an execute command from the input device, performing the operation, wherein the operation further comprises:

causing an electronic search in the information source, by an information management program external to the document editing program, for the search term in order to find whether the search term is included in the information source; and performing an action having a type,

wherein the type of action depends at least in part on whether the search term is included in the information source, and if the search term is so included, and if the information source includes the second information, the action comprises causing insertion of at least part of the second information into the document.

2. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein:

when the information source does not include the search term, the action comprises causing indication to the user that the information source does not include the search term.

3. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein:

when (i) the information source includes the search term, (ii) the selected textual information includes a name, (iii) the information source further includes the second information, and (iv) the second information includes an address, the action further comprises causing insertion of at least part of the address into the document.

4. At least one non-transitory computer readable medium according to claim 3, wherein the instructions establish processes wherein, when the second information includes a plurality of addresses, the operation further comprises allowing the user to choose one of the plurality of addresses to use for insertion into the document.

5. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein, when the selected textual information includes information that is not in the information source, the operation further comprises:

allowing the user to cause storage of at least some of the selected textual information in the information source.

6. At least one non-transitory computer readable medium according to claim 5, wherein the instructions establish processes wherein, when the selected textual information includes a name and an address and the information source does not include the name, allowing the user to cause storage comprises allowing the user to cause storage of at least part of the selected information in the information source as a new contact.

7. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein the information source is available over a network.

8. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein the information source includes an e-mail address and the operation includes causing an e-mail to be sent to the e-mail address.

12

9. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein, when the type or types of contact information includes a name, the operation includes causing display by the information management program of at least part of a contact information record in the information source corresponding to the name.

10. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein the document editing program is a spreadsheet program.

11. At least one non-transitory computer readable medium according to claim 1, wherein the instructions establish processes wherein the type of operation includes updating the document with information from the information source.

12. A method, for contact information handling, implemented by running a document editing program in a computer, the method comprising:

allowing a user to enter textual information into a document using the document editing program;

displaying the textual information in the document electronically using the document editing program;

allowing, in the document editing program, the user to select in the document at least a portion of the textual information while the textual information is displayed;

following user selection of textual information in the document, analyzing, by the document editing program, the selected textual information to determine if the selected textual information is regarded by the document editing program as contact information and what type or types of contact information the selected textual information is;

providing an input device configured by the document editing program to allow the user to initiate an operation, such operation being of a type depending at least in part on the type or types of contact information of the selected textual information, the operation comprising identifying at least part of the selected textual information to use as a search term in order to find second information, of a specific type or types, associated with the search term in an information source external to the document;

after identifying at least part of the selected information to use as a search term, and in consequence of receipt by the document editing program of an execute command from the input device, performing the operation, wherein the operation further comprises:

causing an electronic search in the information source, by an information management program external to the document editing program, for the search term in order to find whether the search term is included in the information source; and performing an action having a type,

wherein the type of action depends at least in part on whether the search term is included in the information source, and if the search term is so included, and if the information source includes the second information, the action comprises causing insertion of at least part of the second information into the document.

13. A method according to claim 12, wherein:

when the information source does not include the search term, the action comprises causing indication to the user that the information source does not include the search term; and

when (i) the information source does include the search term, (ii) the selected textual information includes a name, (iii) the information source further includes the

US 7,921,356 B2

13

second information, and (iv) the second information includes an address, the action further comprises causing insertion of at least part of the address into the document.

14. A method according to claim 12, wherein, when the second information includes a plurality of addresses, the operation further comprises allowing the user to choose one of the plurality of addresses to use for insertion into the document.

15. A method according to claim 12, wherein, when the selected textual information includes information that is not in the information source, the operation further comprises:

- allowing the user to cause storage of at least some of the selected textual information in the information source;
- wherein, when the selected textual information includes a name and an address and the information source does not include the name, allowing the user to cause storage comprises allowing the user to cause storage of at least part of the selected information in the information source as a new contact.

16. A method according to claim 12, wherein the information source is available over a network.

17. A method according to claim 12, wherein the information source includes an e-mail address and the operation includes causing an e-mail to be sent to the e-mail address.

18. A method according to claim 12, wherein, when the type or types of contact information includes a name, the operation includes causing display by the information management program of at least part of a contact information record in the information source corresponding to the name.

19. A method according to claim 12, wherein the type of operation includes updating the document with information from the information source.

20. An apparatus for contact information handling, comprising:

- a processor; and
- a memory storing instructions executable by the processor to perform processes that include:
 - allowing a user to enter textual information into a document using a document editing program;

14

displaying the textual information in the document electronically using the document editing program; allowing, in the document editing program, the user to select in the document at least a portion of the textual information while the textual information is displayed;

following user selection of textual information in the document, analyzing, by the document editing program, the selected textual information to determine if the selected textual information is regarded by the document editing program as contact information and what type or types of contact information the selected textual information is;

providing an input device configured by the program to allow the user to initiate an operation, such operation being of a type depending at least in part on the type or types of contact information of the selected textual information, the operation comprising identifying at least part of the selected textual information to use as a search term in order to find second information, of a specific type or types, associated with the search term in an information source external to the document;

after identifying at least part of the selected information to use as a search term, and in consequence of receipt by the document editing program of an execute command from the input device, performing the operation, wherein the operation further comprises:

- causing an electronic search in the information source, by an information management program external to the document editing program, for the search term in order to find whether the search term is included in the information source; and performing an action having a type,

wherein the type of action depends at least in part on whether the search term is included in the information source, and if the search term is so included, and if the information source includes the second information, the action comprises causing insertion of at least part of the second information into the document.

* * * * *

EXHIBIT 5



US006411941B1

(12) **United States Patent**
Mullor et al.

(10) **Patent No.:** **US 6,411,941 B1**
 (45) **Date of Patent:** **Jun. 25, 2002**

(54) **METHOD OF RESTRICTING SOFTWARE OPERATION WITHIN A LICENSE LIMITATION**

(75) Inventors: **Miki Mullor; Julian Valiko**, both of Ramat Hasharon (IL)

(73) Assignee: **Beeble, Inc.**, Newport Beach, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/164,777**

(22) Filed: **Oct. 1, 1998**

(30) **Foreign Application Priority Data**

May 21, 1998 (IL) 124571

(51) **Int. Cl.**⁷ **G06F 17/60**

(52) **U.S. Cl.** **705/59; 705/50; 705/51; 705/53; 705/57**

(58) **Field of Search** **705/51, 54, 56, 705/57, 58, 59, 1, 50, 52, 53; 713/187, 189, 200**

(56) **References Cited**

U.S. PATENT DOCUMENTS

- 4,866,769 A 9/1989 Karp
- 4,903,296 A 2/1990 Chandra et al.
- 4,924,378 A 5/1990 Hershey et al.
- 5,386,369 A 1/1995 Christiano
- 5,390,297 A 2/1995 Barber et al.
- 5,479,639 A * 12/1995 Ewertz et al. 395/430
- 5,490,216 A * 2/1996 Richadson, III 380/4
- 5,671,412 A 9/1997 Christiano
- 5,684,951 A * 11/1997 Goodman et al. 395/188.01
- 5,754,763 A 5/1998 Bereiter
- 5,758,068 A 5/1998 Brandt et al.
- 5,758,069 A 5/1998 Olsen
- 5,790,664 A 8/1998 Coley et al.
- 5,826,011 A 10/1998 Chou et al.
- 5,892,900 A * 4/1999 Ginter et al. 395/186
- 5,905,860 A 5/1999 Olsen et al.

- 6,000,030 A * 12/1999 Steinberg et al. 713/200
- 6,006,190 A 12/1999 Baena-Arnaiz et al.
- 6,021,438 A 2/2000 Duvvoori et al.
- 6,023,763 A 2/2000 Grumpstrup et al.
- 6,052,600 A * 4/2000 Fette et al. 455/509
- 6,055,503 A 4/2000 Horstmann
- 6,067,582 A * 5/2000 Smith et al. 710/5
- 6,073,256 A 6/2000 Sesma
- 6,078,909 A 6/2000 Knutson
- 6,128,741 A 10/2000 Goetz et al.
- 6,173,446 B1 1/2001 Khan et al.
- 6,189,146 B1 * 2/2001 Misra et al. 717/11
- 6,192,475 B1 2/2001 Wallace
- 6,198,875 B1 * 3/2001 Edenson et al. 386/94
- 6,226,747 B1 5/2001 Larsson et al.
- 6,233,567 B1 5/2001 Cohen
- 6,243,468 B1 6/2001 Pearce et al.
- 6,272,636 B1 8/2001 Neville et al.
- 6,298,138 B1 10/2001 Gotoh et al.

FOREIGN PATENT DOCUMENTS

JP 408286906 A * 11/1996 G06F/9/06

OTHER PUBLICATIONS

Dornbusch et al., Destop management software: no need to adjust your set., Infoworld, v17, n37, p60.*

* cited by examiner

Primary Examiner—Hyung-Sub Sough

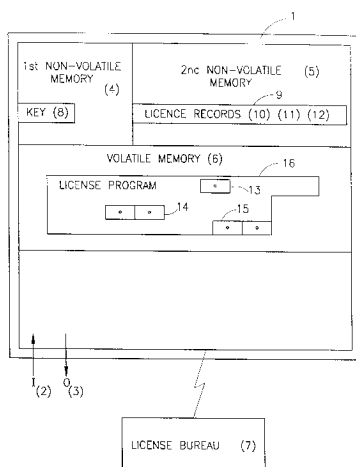
Assistant Examiner—Calvin L Hewitt

(74) *Attorney, Agent, or Firm*—Venable; Robert Kinberg; Jeffri A. Kaminski

(57) **ABSTRACT**

A method of restricting software operation within a license limitation that is applicable for a computer having a first non-volatile memory area, a second non-volatile memory area, and a volatile memory area. The method includes the steps of selecting a program residing in the volatile memory, setting up a verification structure in the non-volatile memories, verifying the program using the structure, and acting on the program according to the verification.

19 Claims, 2 Drawing Sheets



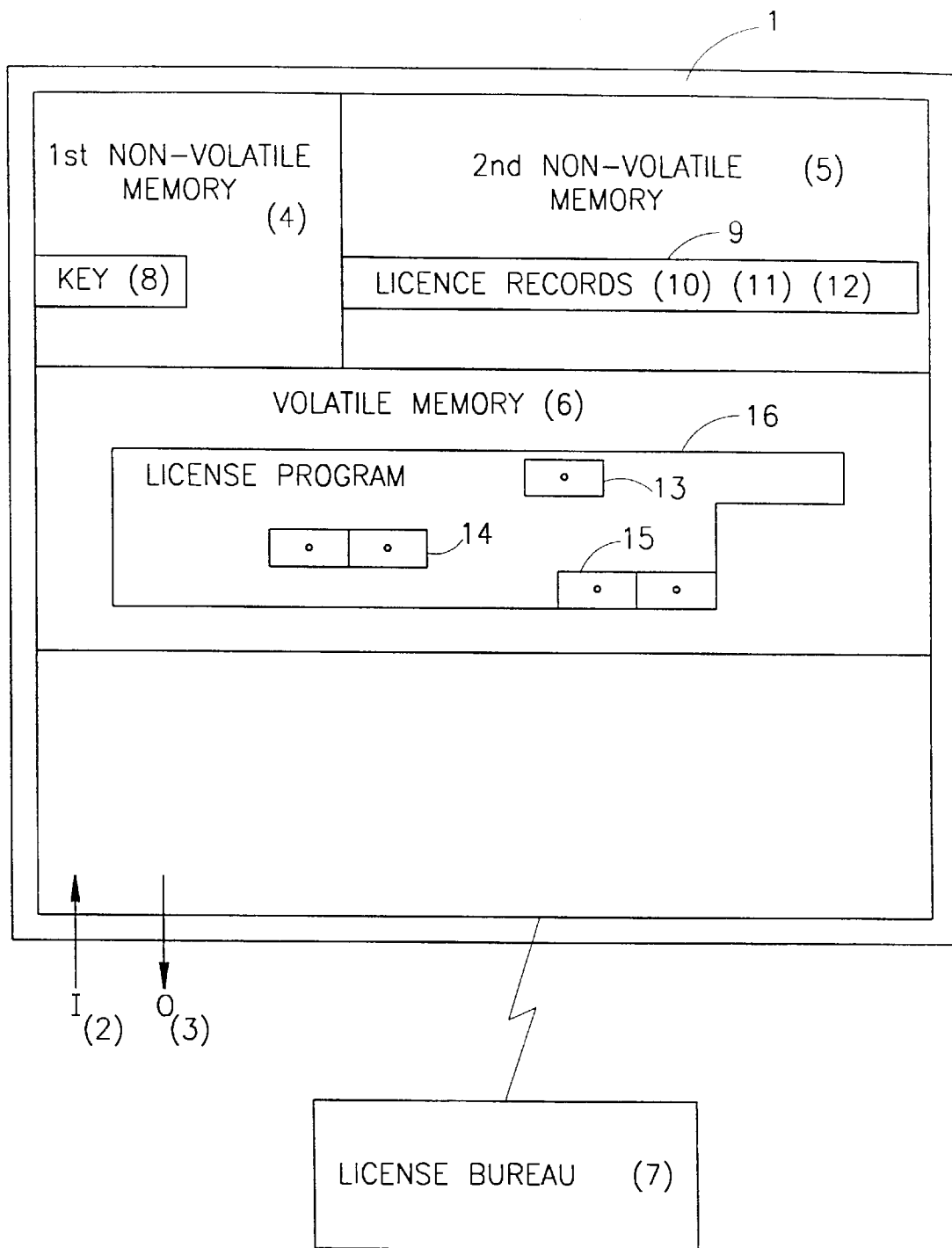


FIG. 1

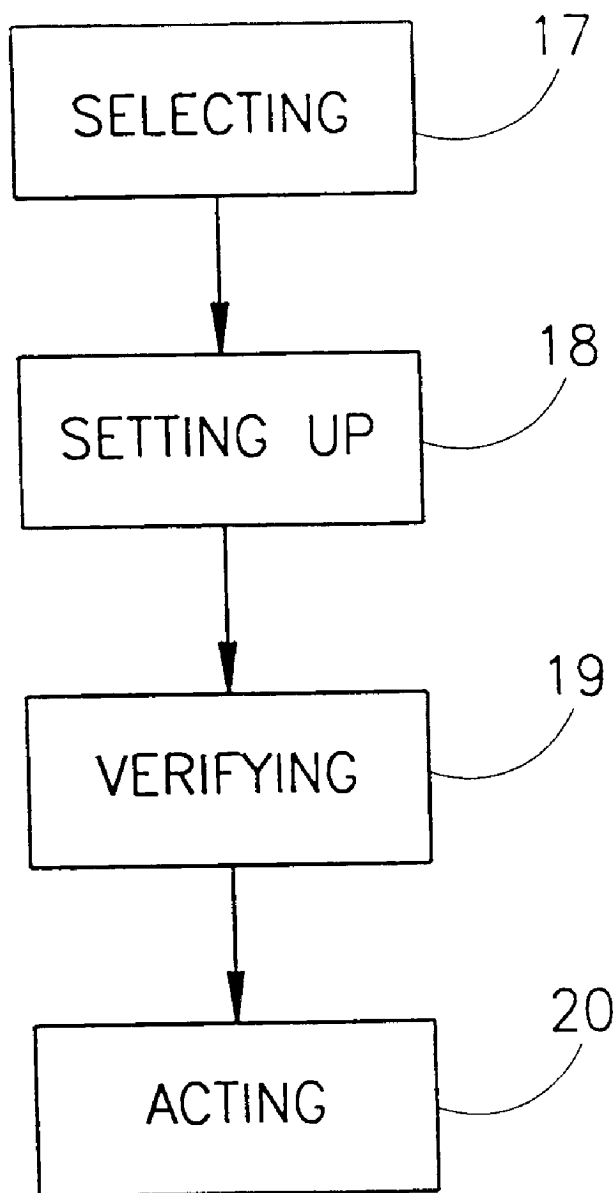


FIG.2

US 6,411,941 B1

1

**METHOD OF RESTRICTING SOFTWARE
OPERATION WITHIN A LICENSE
LIMITATION**

FIELD OF THE INVENTION

This invention relates to a method and system of identifying and restricting an unauthorized software program's operation.

BACKGROUND OF THE INVENTION

Numerous methods have been devised for the identifying and restricting of an unauthorized software program's operation. These methods have been primarily motivated by the grand proliferation of illegally copied software, which is engulfing the marketplace. This illegal copying represents billions of dollars in lost profits to commercial software developers.

Software based products have been developed to validate authorized software usage by writing a license signature onto the computer's volatile memory (e.g. hard disk). These products may be appropriate for restricting honest software users, but they are very vulnerable to attack at the hands of skilled system's programmers (e.g. "hackers"). These license signatures are also subject to the physical instabilities of their volatile memory media.

Hardware based products have also been developed to validate authorized software usage by accessing a dongle that is coupled e.g. to the parallel port of the P.C. These units are expensive, inconvenient, and not particularly suitable for software that may be sold by downloading (e.g. over the internet).

There is accordingly a need in the art to provide for a system and method that substantially reduce or overcome the drawbacks of hitherto known solutions.

SUMMARY OF THE INVENTION

The present invention relates to a method of restricting software operation within a license limitation. This method strongly relies on the use of a key and of a record, which have been written into the non-volatile memory of a computer.

For a better understanding of the underlying concept of the invention, there follows a specific non-limiting example. Thus, consider a conventional computer having a conventional BIOS module in which a key was embedded at the ROM section thereof, during manufacture. The key constitutes, effectively, a unique identification code for the host computer. It is important to note that the key is stored in a non-volatile portion of the BIOS, i.e. it cannot be removed or modified.

Further, according to the invention, each application program that is to be licensed to run on the specified computer, is associated with a license record; that consists of author name, program name and number of licensed users (for network). The license record may be held in either encrypted or explicit form.

Now, there commences an initial license establishment procedure, where a verification structure is set in the BIOS so as to indicate that the specified program is licensed to run on the specified computer. This is implemented by encrypting the license record (or portion thereof) using said key (or portion thereof) exclusively or in conjunction with other identification information) as an encryption key. The resulting encrypted license record is stored in another (second) non-volatile section of the BIOS, e.g. E²PROM (or the

2

ROM). It should be noted that unlike the first non-volatile section, the data in the second non-volatile memory may optionally be erased or modified (using E²PROM manipulation commands), so as to enable to add, modify or remove licenses. The actual format of the license may include a string of terms that correspond to a license registration entry (e.g. lookup table entry or entries) at a license registration bureau (which will be further described as part of the preferred embodiment of the present invention).

Having placed the encrypted license record in the second non-volatile memory (e.g. the E²PROM), the process of verifying a license may be commenced. Thus, when a program is loaded into the memory of the computer, a so called license verifier application, that is a priori running in the computer, accesses the program under question, retrieves therefrom the license record, encrypts the record utilizing the specified unique key (as retrieved from the ROM section of the BIOS) and compares the so encrypted record to the encrypted records that reside in the E²PROM. In the case of match, the program is verified to run on the computer. If on the other hand the sought encrypted data record is not found in the E²PROM database, this means that the program under question is not properly licensed and appropriate application define action is invoked (e.g. informing to the user on the unlicensed status, halting the operation of the program under question etc.)

Those versed in the art will readily appreciate that any attempt to run a program at an unlicensed site will be immediately detected. Consider, for example, that a given application, say Lotus 123, is verified to run on a given computer having a first identification code (k1) stored in the ROM portion of the BIOS thereof. This obviously requires that the license record (LR) of the application after having been encrypted using k1 giving rise to (LR)_{k1} is stored in the E²PROM of the first computer.

Suppose now that a hacker attempts to run the specified application in a second computer having a second identification code (k2) stored in the ROM portion of the BIOS thereof. All or a portion the database contents (including of course (LR)_{k1}) that reside in the E²PROM portion in the first computer may be copied in a known per se means to the second computer. It is important to note that the hacker is unable to modify the key in the ROM of the second computer to K1, since, as recalled, the contents of the ROM is established during manufacture and is practically invariable.

Now, when the application under question is executed in the second computer, the license verifier retrieves said LR from the application and, as explained above, encrypts it using the key as retrieved from the ROM of the second computer, i.e. k2 giving rise to encrypted license record (LR)_{k2}. Obviously, the value (LR)_{k2} does not reside in the E²PROM database section of the second computer (since it was not legitimately licensed) and therefore the specified application is invalidated. It goes without saying that the data copied from the first (legitimate) computer is rendered useless, since comparing (LR)_{k2} with the copied value (LR)_{k1} results, of course, in mismatch.

The example above is given for clarity of explanation only and is by no means binding.

In its broadest aspect, the invention provides for a method of restricting software operation within a license limitation including; for a computer having a first non-volatile memory area, a second non-volatile memory area, and a volatile memory area; the steps of: selecting a program residing in the volatile memory, setting up a verification structure in the

US 6,411,941 B1

3

non-volatile memories, verifying the program using the structure, and acting on the program according to the verification.

An important advantage in utilizing non-volatile memory such as that residing in the BIOS is that the required level of system programming expertise that is necessary to intercept or modify commands, interacting with the BIOS, is substantially higher than those needed for tampering with data residing in volatile memory such as hard disk. Furthermore, there is a much higher cost to the programmer, if his tampering is unsuccessful, i.e. if data residing in the BIOS (which is necessary for the computer's operability) is inadvertently changed by the hacker. This is too high of a risk for the ordinary software hacker to pay. Note that various recognized means for hindering the professional-like hacker may also be utilized (e.g. anti-debuggers, etc.) in conjunction with the present invention.

In the context of the present invention, a "computer" relates to a digital data processor. These processors are found in personal computers, or on one or more processing cards in multi-processor machines. Today, a processor normally includes a first non-volatile memory, a second non-volatile memory, and data linkage access to a volatile memory. There are also processors having only one non-volatile memory or having more than two non-volatile memories; all of which should be considered logically as relating to having a first and a second non-volatile memory areas. There are also computational environments where the volatile memory is distributed into numerous physical components, using a bus, LAN, etc.; all of which should logically be considered as being a volatile memory area.

According to the preferred embodiment of the present invention, there is further provided a license authentication bureau which can participate in either or both of:

- (i) establishing the license record in the second non-volatile memory; and
- (ii) verifying if the key and license record in the non-volatile memory(s) is compatible with the license record information as extracted from the application under question.

The bureau is a telecommunications accessible processor where functions such as formatting, encrypting, and verifying may be performed. Performing these or other functions at the bureau helps to limit the understanding of potential software hackers; since they can not observe how these functions are constructed. Additional security may also be achieved by forcing users of the bureau to register, collecting costs for connection to the bureau, logging transactions at the bureau, etc.

According to one example of using the bureau, setting up a verification structure further includes the steps of: establishing, between the computer and the bureau, a two-way data-communications linkage; transferring, from the computer to the bureau, a request-for-license including an identification of the .computer and the license-record's contents from the selected program; forming an encrypted license-record at the bureau by encrypting parts of the request-for-license using part of the identification as the encryption key; and transferring, from the bureau to the computer, the encrypted license-record.

According to another example of using the bureau, verifying the program further includes the steps of: establishing, between the computer and the bureau, a two-way data-communications linkage; transferring, from the computer to the bureau, a request-for-license-verification including an identification of the computer, the encrypted license-record

4

for the selected program from the second non-volatile memory, and the licensed-software-program's license-record contents; enabling the comparing at the bureau; and transferring, from the bureau to the computer, the result of the comparing.

The actual key that serves for identifying the computer may be composed of the pseudo-unique key exclusively, or, if desired, in combination with information, e.g. information related to the registration of the user such as e.g. place, telephone number, user name, license number, etc. In the context of the present invention, a "pseudo-unique" key may relate to a bit string which uniquely identifies each first non-volatile memory. Alternately the "pseudo-unique" key may relate to a random bit string (or to an assigned bit string) of sufficient length such that: there is an acceptably low probability of a successful unauthorized transfer of licensed software between two computers, where the first volatile memories of these two computers have the same key.

It should be noted that the license bureau might maintain a registry of keys and of licensed programs that have been registered at the bureau in association with these keys. This registry may be used to help facilitate the formalization of procedures for the transfer of ownership of licensed software from use on one computer to use on another computer.

Constructing the key in the manner specified may hinder the hacker in cracking the proposed encryption scheme of the invention, in particular when the establishment of the license record or the verification thereof is performed in the bureau. Those versed in the art will readily appreciate that the invention is by no means bound by the data, the algorithms, or the manner of operation of the bureau. It should be noted that the tasks of establishing and/or verifying a license record may be shared between the bureau and the computer, done exclusively at the computer, or done exclusively at the bureau. The pseudo-unique key length needs to be long enough to hinder encryption attack schemes. The establishing of the key may be done at any time from the non-volatile memory's manufacture until an attempted use of an established license-record in the non-volatile memory. The key is used for encryption or decryption operations associated with license-records. In principle, the manufacturer of the licensed-software-program may specify the license-record format and therefore different formats may, if desired, be used for respective applications.

According to the preferred embodiment of the present invention, the pseudo-unique key is a unique-identification bit string that is written onto the first non-volatile memory by the manufacturer of the is memory media.

According to one, non-limiting, preferred embodiment of the present invention, the first non-volatile memory area is a ROM section of a BIOS; the second non-volatile memory area is a E²PROM section of a BIOS; and the volatile memory is a RAM e.g. hard disk and/or internal memory of the computer.

The present invention also relates to a non-volatile memory media used as a BIOS of a computer, for restricting software operation within a license limitation, wherein a pseudo-unique key is established.

According to the preferred embodiment of the non-volatile memory media of the present invention, the pseudo-unique key is established in a ROM section of the BIOS.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to understand the invention and to see how it may be carried out in practice, a preferred embodiment will now be described, by way of non-limiting example only, with reference to the accompanying drawings, in which:

US 6,411,941 B1

5

FIG. 1 is a schematic diagram of a computer and a license bureau; and

FIG. 2 is a generalized flow chart of the sequence of operations performed according to one embodiment of the invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

A schematic diagram of a computer and a license bureau is shown in FIG. 1. Thus, a computer processor (1) is associated with input operations (2) and with output operations (3). This computer (processor) internally contains a first non-volatile memory area (4) (e.g. the ROM section of the BIOS), a second non-volatile memory area (5) (e.g. the E²PROM section of the BIOS), and a volatile memory area (6) (e.g. the internal RAM memory of the computer).

The computer processor is in temporary telecommunications linkage with a license bureau (7).

The first non-volatile memory includes a pseudo-random identification key (8), which exclusively or in combination with other information (e.g. user name), is sufficient to uniquely differentiate this first non-volatile memory from all other first non-volatile memories. As specified before, said key constitutes unique identification of the computer.

The second non-volatile memory includes a license-record-area (9) e.g. which contains at least one encrypted license-record (e.g. three records 10–12). The volatile memory accommodates a license program (16) having license record fields (13–15) appended thereto. By way of example said fields stand for Application names (e.g. Lotus 123), Vendor name (Lotus inc.), and number of licensed copies (1 for stand alone usage, >1 for number of licensed users for a network application).

Those versed in the art will readily appreciate that the license record is not necessarily bound to continuous fields. In fact, the various license content components of the data record may be embedded in various locations in the application. Any component may, if desired, be encrypted.

Each one of the encrypted license records (10–12) is obtained by encrypting the corresponding license record as extracted from program 16, utilizing for encryption the identification key (8).

In a typical, yet not exclusive, sequence of operation, a transaction/request is sent, by the computer to the bureau. This transaction includes the key (8), the encrypted license-records (10–12), contents from the license program used in forming a license record (e.g. fields 13–15), and other items of information as desired.

The bureau forms the proposed license-record from the contents, encrypts (utilizing predetermined encryption algorithm) the so formed license-record using the key (8), and compares the so formed encrypted license-record with the license-record (10–12). The bureau generates an overlay according to the result of the comparison indicating successful comparison, non-critical failure comparison and the critical failure comparison.

The bureau returns the overlay which will direct the computer in subsequent operation. Thus, a success overlay will allow the license program to operate. A non-critical failure overlay will ask for additional user interactions. A critical failure overlay will cause permanent disruption to the computer's BIOS operations. Thus, software operation of the program is methodologically according to a license limitation restriction.

Those versed in the art will readily appreciate that the implementation as described with reference to FIG. 1 is by

6

no means binding. Thus, by way of non-limiting example, the bureau, instead of being external entity may form part of the computer.

Attention is now directed to FIG. 2, showing a generalized flow chart of the sequence of operations performed according to one embodiment of the invention.

Thus, selecting (17) a program includes the step of: establishing a licensed-software-program in the volatile memory of the computer wherein the licensed-software-program includes contents used to form a license-record. These contents, be they centralize or decentralized, may include terms, identifications, specifications, or limitations related to the manufacturer of a software product, the distributor of a software product, the purchaser of a software product, a licensor, a licensee, items of computer hardware or components thereof, or to other terms and conditions related to the aforesaid.

Setting up (18) the verification structure includes the steps of: establishing or certifying the existence of a pseudo-unique key in the first non-volatile memory area; and establishing at least one license-record location in the first or the second nonvolatile memory area.

Establishing a license-record includes the steps of: forming a license-record by encrypting of the contents used to form a license-record with other predetermined data contents, using the key; and establishing the encrypted license-record in one of the at least one established license-record locations (e.g. 10–12 in FIG. 1).

Verifying (19) the program includes the steps of: encrypting the licensed-software-program's license-record contents from the volatile memory area or decrypting the license-record in the first or the second non-volatile memory area, using the key; and comparing the encrypted licensed-software-program's license-record contents with the encrypted license-record in the first or the second non-volatile memory area, or comparing the licensed-software-program's license-record contents with the decrypted license-record in the first or the second non-volatile memory area.

Acting (20) on the program includes the step of: restricting the program's operation with predetermined limitations if the comparing yields non-unity or insufficiency. In this context "non-unity" relates to being unequal with respect to a specific equation (e.g. $A=B+1$); and "insufficiency" relates to being outside of a relational bound (e.g. $A>B+1$). "Restricting the program's operation with predetermined limitations" may include actions such as erasing the software in volatile memory, warning the license applicant/user, placing a fine on the applicant/user through the billing service charges collected at the license bureau (if applicable), or scrambling sections of the BIOS of the computer (or of functions interacting therewith).

The present invention has been described with a certain degree of particularity but it should be understood that various modifications and alterations may be made without departing from the scope or spirit of the invention as defined by the following claims.

What is claimed is:

1. A method of restricting software operation within a license for use with a computer including an erasable, non-volatile memory area of a BIOS of the computer, and a volatile memory area; the method comprising the steps of: selecting a program residing in the volatile memory, using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record,

7

verifying the program using at least the verification structure from the erasable non-volatile memory of the BIOS, and

acting on the program according to the verification.

2. A method according to claim 1, further comprising the steps of:

establishing a license authentication bureau.

3. A method according to claim 2, wherein setting up a verification structure further comprising the steps of: establishing, between the computer and the bureau, a two-way data-communications linkage; transferring, from the computer to the bureau, a request-for-license including an identification of the computer and the license-record's contents from the selected program; forming an encrypted license-record at the bureau by encrypting parts of the request-for-license using part of the identification as an encryption key; transferring, from the bureau to the computer, the encrypted license-record; and storing the encrypted license record in the erasable non-volatile memory area of the BIOS.

4. A method according to claim 2, wherein verifying the program further comprises the steps of: establishing, between the computer and the bureau, a two-way data-communications linkage; transferring, from the computer to the bureau, a request-for-license verification including an identification of the computer, an encrypted license-record for the selected program from the erasable, non-volatile memory area of the BIOS, and the program's license-record; enabling the comparing at the bureau; and transferring, from the bureau to the computer, the result of the comparing.

5. A method according to claim 3 wherein the identification of the computer includes the unique key.

6. A method according to claim 1 wherein selecting a program includes the steps of: establishing a licensed-software-program in the volatile memory of the computer wherein said licensed-software-program includes contents used to form the license-record.

7. A method according to claim 6 wherein using an agent to set up the verification structure includes the steps of: establishing or certifying the existence of a pseudo-unique key in a first non-volatile memory area of the computer; and establishing at least one license-record location in the first nonvolatile memory area or in the erasable, non-volatile memory area of the BIOS.

8. A method according to claim 6 wherein establishing a license-record includes the steps of: forming a license-record by encrypting of the contents used to form a license-record with other predetermined data contents, using the key; and establishing the encrypted license-record in one of the at least one established license-record locations.

9. A method according to claim 7 wherein verifying the program includes the steps of: encrypting the licensed-software-program's license-record contents from the volatile memory area or decrypting the license-record in the erasable, non-volatile memory area of the BIOS, using the pseudo-unique key; and comparing the encrypted licenses-software-program's license-record contents with the encrypted license-record in the erasable, non-volatile memory area of the BIOS, or comparing the license-software-program's license-record contents with the decrypted license-record in erasable non-volatile memory area of the BIOS.

10. A method according to claim 9 wherein acting on the program includes the step: restricting the program's operation with predetermined limitations if the comparing yields non-unity or insufficiency.

8

11. A method according to claim 1 wherein the volatile memory is a RAM.

12. The method of claim 1, wherein a pseudo-unique key is stored in the non-volatile memory of the BIOS.

13. The method of claim 1, wherein a unique key is stored in a first non-volatile memory area of the computer.

14. The method according claim 13, wherein the step of using the agent to set up the verification record, including the license record, includes encrypting a license record data in the program using at least the unique key.

15. The method according to claim 14, wherein the verification comprises:

extracting the license record from the software program; encrypting the license record using the unique key stored in the first non-volatile memory area of the computer to form second encrypted license information; and

comparing the encrypted license information stored in the erasable, non-volatile memory area of the BIOS of the computer with the second encrypted license information.

16. The method according to claim 13, wherein the step of verifying the program includes a decrypting the license record data accommodated in the erasable second non-volatile memory area of the BIOS using at least the unique key.

17. The method according to claim 13, wherein the step of verifying the program includes encrypting the license record that is accommodated in the program using at least the unique key.

18. A method for accessing an application software program using a pseudo-unique key stored in a first non-erasable non-volatile memory area of a computer, the first non-volatile memory area being unable to be programmatically changed, the method, comprising:

loading the application software program residing in a non-volatile memory area of the computer;

using an agent to perform the following steps:

extracting license information from software program; encrypting license information using the pseudo-unique key stored in the first non-volatile memory area;

storing the encrypting license information in a second erasable, writable, non-volatile memory area of the BIOS of the computer;

subsequently verifying the application software program based on the encrypted license information stored in the second erasable, writable, non-volatile memory area of the BIOS; and

acting on the application software program based on the verification.

19. The method of claim 18, wherein the verification comprises:

extracting the license information from the software program;

encrypting the license information using the pseudo-unique key stored in the first non-volatile memory area of the computer to form second encrypted license information; and

comparing the encrypted license information stored in the second erasable, writable, non-volatile memory area of the BIOS of the computer with the second encrypted license information.

EXHIBIT 6



US005590259A

United States Patent [19]

[11] Patent Number: **5,590,259**

Anderson et al.

[45] Date of Patent: ***Dec. 31, 1996**

[54] **SYSTEM AND METHODS FOR IMPROVED SPREADSHEET INTERFACE WITH USER-FAMILIAR OBJECTS** 5,347,295 9/1994 Agulnick et al. 345/156
 5,416,895 5/1995 Anderson et al. 395/148

FOREIGN PATENT DOCUMENTS

[75] Inventors: **Charles R. Anderson**, Santa Cruz; **Robert W. Warfield**, Aptos; **Istvan Cseri**, Scotts Valley; **Murray K. Low**; **Weikuo Liaw**, both of Santa Cruz; **Alan M. Bush**, Menlo Park, all of Calif.
 90310897 5/1991 European Pat. Off. .
 90123564 6/1991 European Pat. Off. .
 90125095 8/1991 European Pat. Off. .
 PCT/US91/06461 3/1992 WIPO .
 PCT/US91/08077 5/1992 WIPO .

OTHER PUBLICATIONS

[73] Assignee: **Borland International, Inc.**, Scotts Valley, Calif.

[*] Notice: The term of this patent shall not extend beyond the expiration date of Pat. No. 5,416,895.

Warfield et al, "Quattro Pro for Windows", PC Magazine, vol. 11 No. 22, p. 136. Dec. 12, 1992.

Stinson, "Diversity Blossoms Amid the Rows and Columns", PC Magazine, vol. 11 No. 19, pp. 111-1170. Nov. 10, 1992.

[21] Appl. No.: **440,184**

Dreyfuss, "The Best of 1992", PC Magazine, vol. 12. No. 1, pp. 110 and 148. Jan. 12, 1993.

[22] Filed: **May 12, 1995**

Anonymous, "Product of the year: a new approach", Infoworld, pp. 16-17. Apr. 26, 1993.

Related U.S. Application Data

[63] Continuation of Ser. No. 866,658, Apr. 8, 1992, Pat. No. 5,416,895.

Barton, "Quattro Pro for Windows: Both Evolutionary and Revolutionary", CPA Software News. Nov. 1992.

[51] Int. Cl.⁶ **G06F 17/40**

Miller, "New Spreadsheet Thinking", PC Magazine, pp. 81-82. Nov. 10, 1992.

[52] U.S. Cl. **395/764; 395/938; 395/939; 395/344**

(List continued on next page.)

[58] **Field of Search** 395/148-161, 395/600; 364/401-408; 345/24-25, 156

Primary Examiner—Joseph H. Feild
Attorney, Agent, or Firm—John A. Smart; David N. Slone

[56] **References Cited**

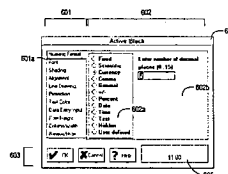
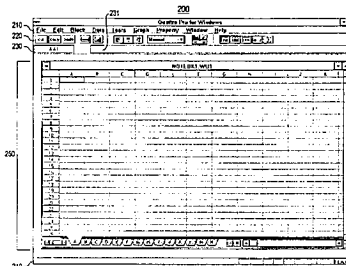
[57] **ABSTRACT**

U.S. PATENT DOCUMENTS

An electronic spreadsheet system includes a notebook interface having a plurality of notebook pages, each of which may contain a spread of information cells, or other desired page type (e.g., Graphs page). Methods are provided for rapidly accessing and processing information on the different pages, including displaying a plurality of page identifiers for selecting individual pages, and further including a preferred syntax for referencing information. Additional methods are provided for editing cells and blocks of cells.

5,021,973	6/1991	Hernandez et al.	364/518
5,033,009	7/1991	Dubnoff	364/523
5,055,998	10/1991	Wright et al.	364/200
5,072,412	12/1991	Henderson, Jr. et al.	395/159
5,093,907	3/1992	Hwong et al.	395/152
5,095,429	3/1992	Harris et al.	364/408
5,121,499	6/1992	McCaskill et al.	395/700
5,140,678	8/1992	Torres	395/159
5,182,793	1/1993	Alexander et al.	395/13
5,231,577	7/1993	Koss	364/419
5,237,651	8/1993	Randall	395/148
5,283,864	2/1994	Knowlton	395/158

56 Claims, 35 Drawing Sheets



5,590,259

Page 2

OTHER PUBLICATIONS

Stinson, "Quattro Pro for Windows Shows Spreadsheet Smarts", PC Magazine, pp. 37-39. Sep. 29, 1992.

Scoville, "Quattro Pro for Windows", PC World, pp. 102-105. Jul. 1992.

Delonas, "Spreadsheets Under Scrutiny", BYTE's Essential Guide to Windows, pp. 70-80. 1992.

Scoville et al, "The Windows Spreadsheet Challenge", PC World, pp. 146-164. Jan. 1993.

Seymour, "No doubt about it, 1992 was The Year of Usability", PC Week, p. 65. Dec. 21, 1992.

Birkholz, "Die Datenbank in der Tabellenkalkulation," *CHIP Zeitschrift für Mikrocomputer-Technik*, No. 3, Mar. 1991, pp. 450-454.

Cakmakov, D. et al., "A Multimedia Cognitive-Based System Model," *Advanced Computer Technology, Reliable Systems and Applications*, 5th Annual European Computer Conference, Bologna, May 13-16, 1991, pp. 282-286.

"Hyperpaper," *IBM Technical Disclosure Bulletin*, vol. 33, No. 9, Feb. 1991, pp. 283-384.

Lotus Development Corporation, *Lotus User's Guide: Lotus 1-2-3 for Windows Release 1.0*, 1991, pp. 37-46, 307-317, 159-172.

Lucid 3-D Tutorial Guide, Ch. 16, 1987, pp. 1-18.

Microsoft Excel User's Guide, 1990, pp. 76-79.

Proceedings: Advanced Computer Technology, Reliable Systems and Applications, 5th Annual European Computer Conf., CompEuro '91, May 13, 1991, Italy, pp. 282-286.

The R-File: An Efficient Access Structure for Proximity Queries, Feb. 1990, pp. 372-379.

100

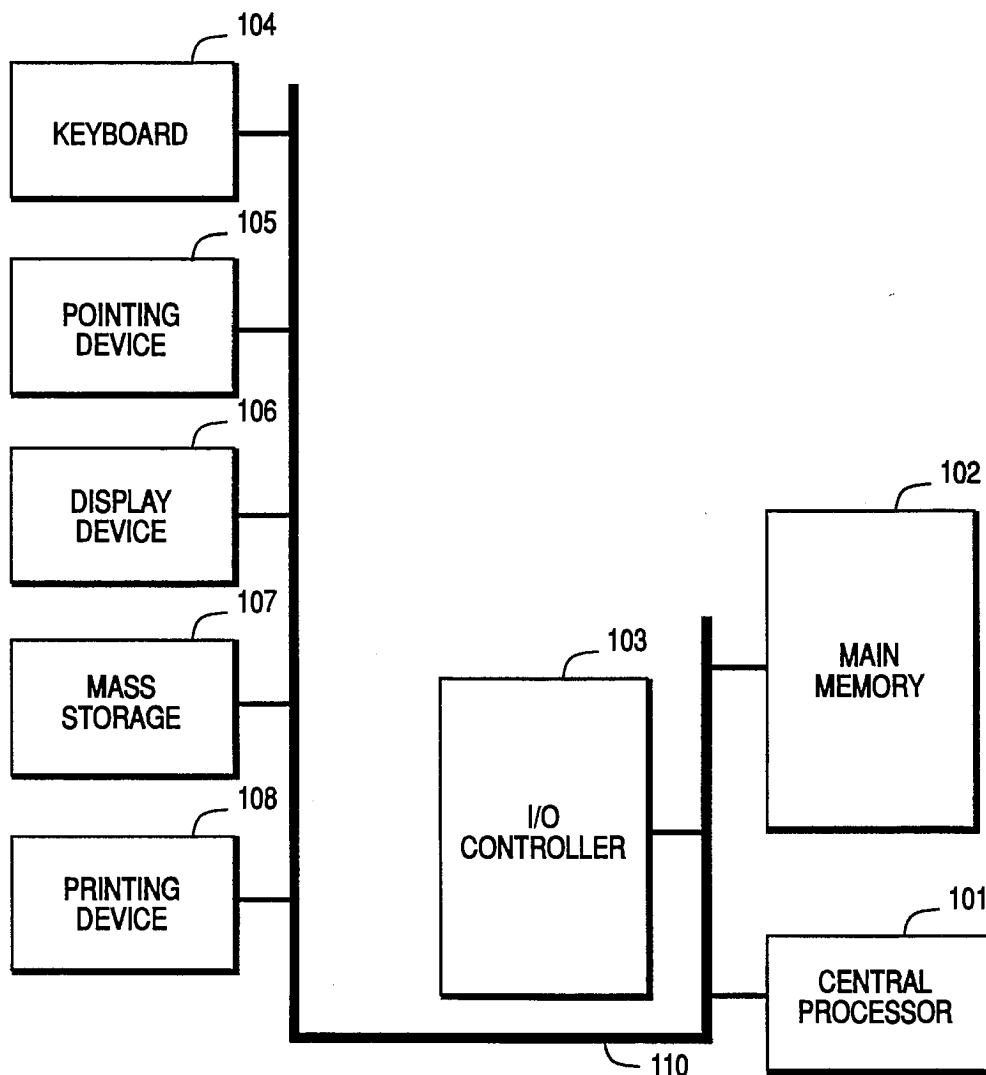


FIG. 1A

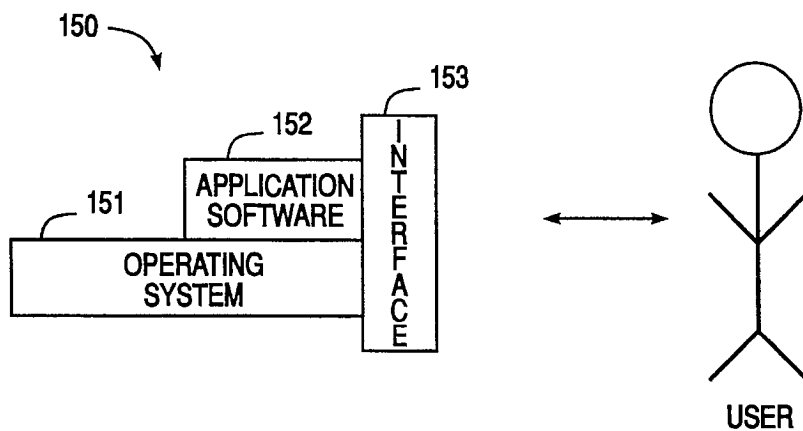


FIG. 1B

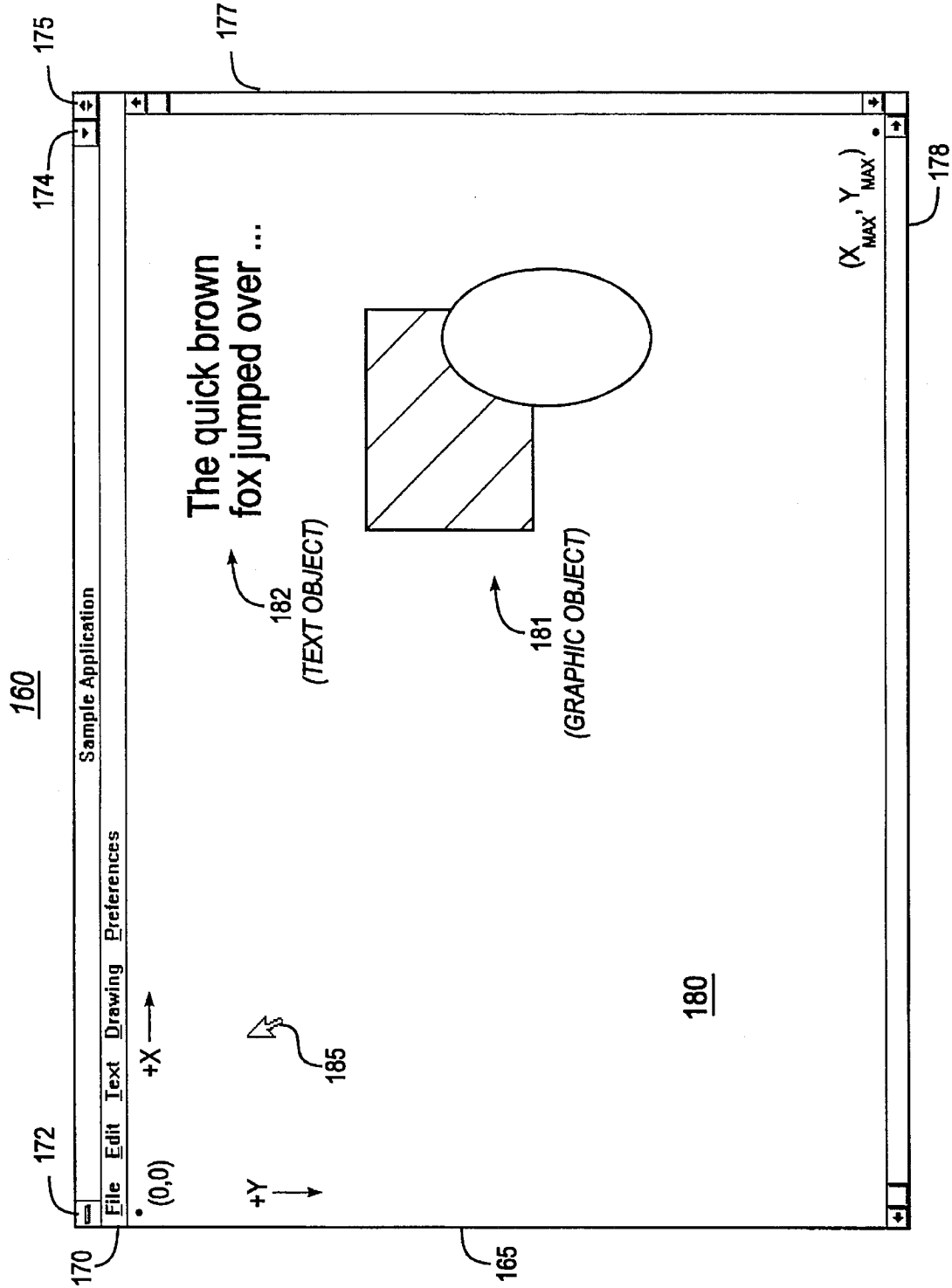


FIG. 1C

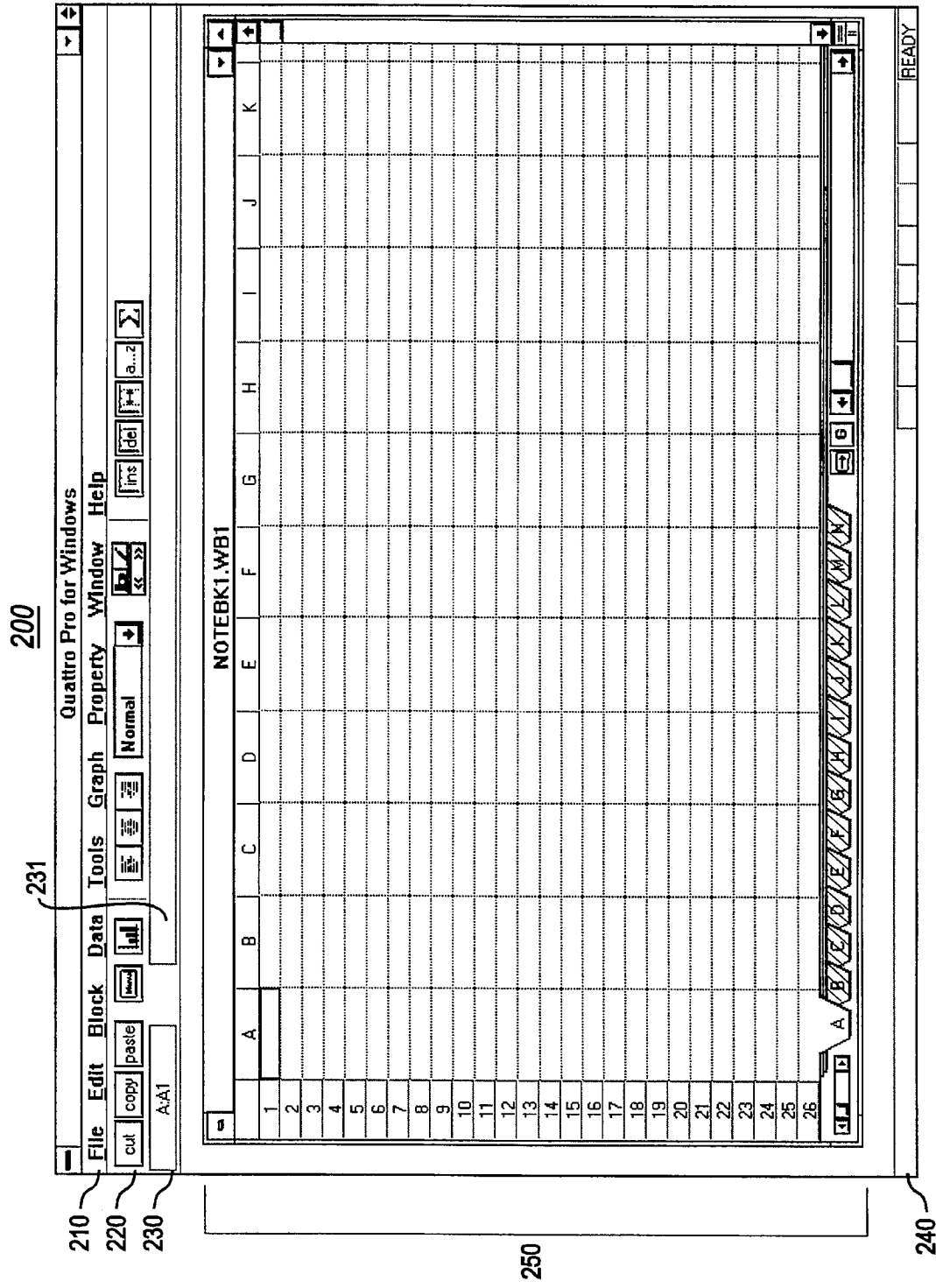


FIG. 2A

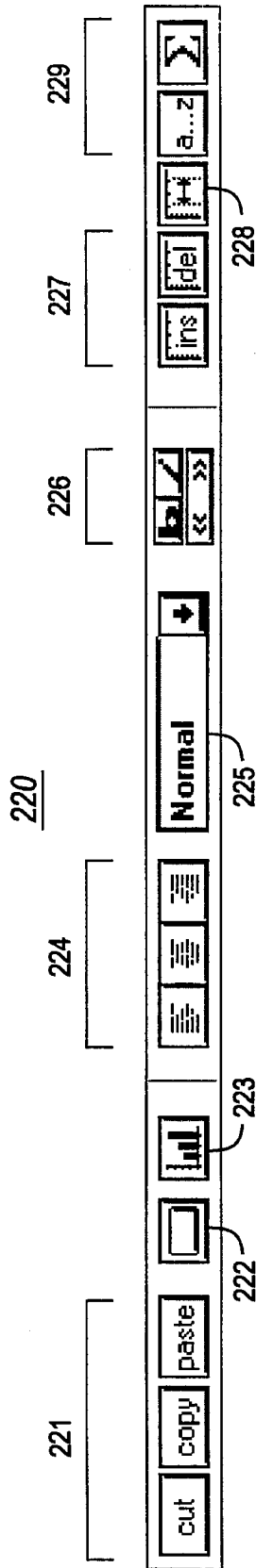


FIG. 2B

260a

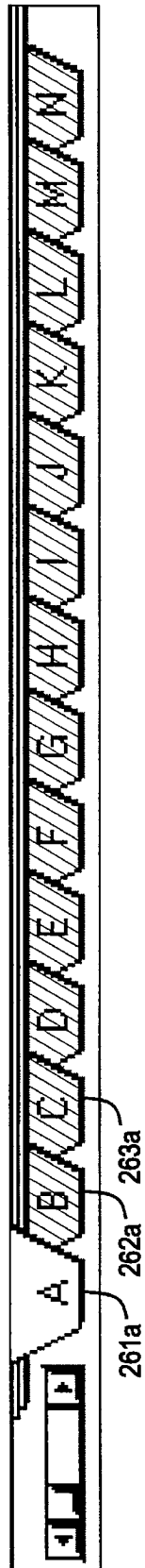


FIG. 2D

260b

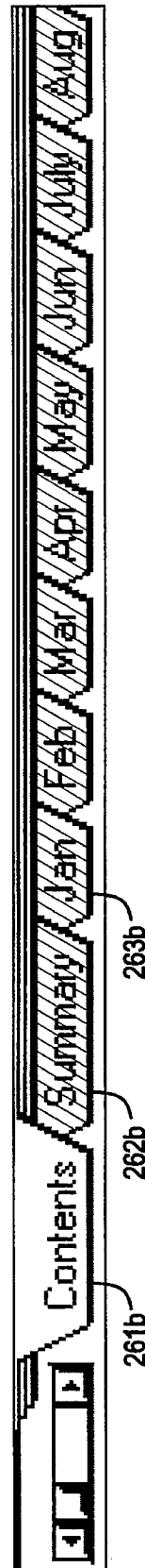


FIG. 2E

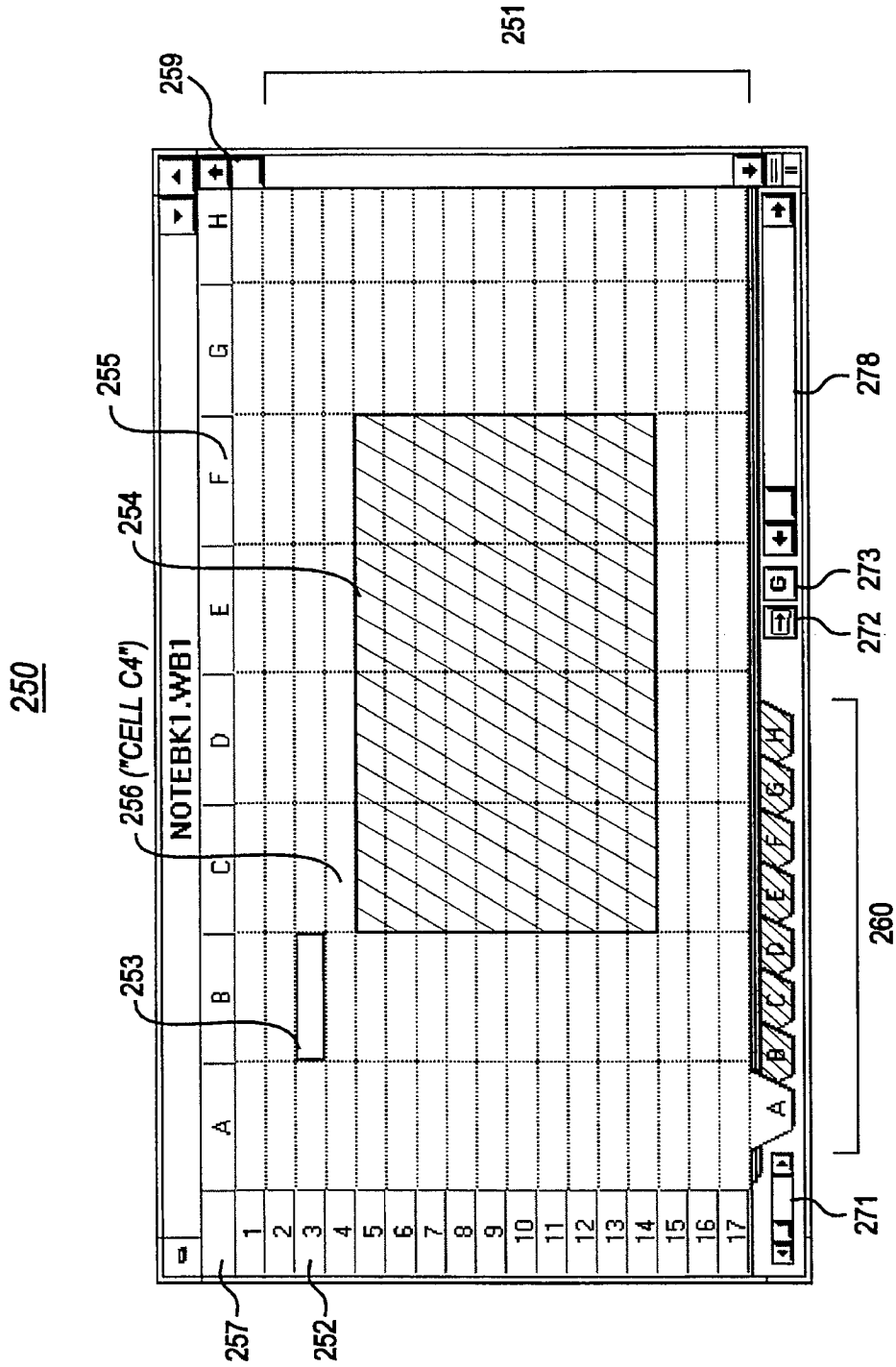
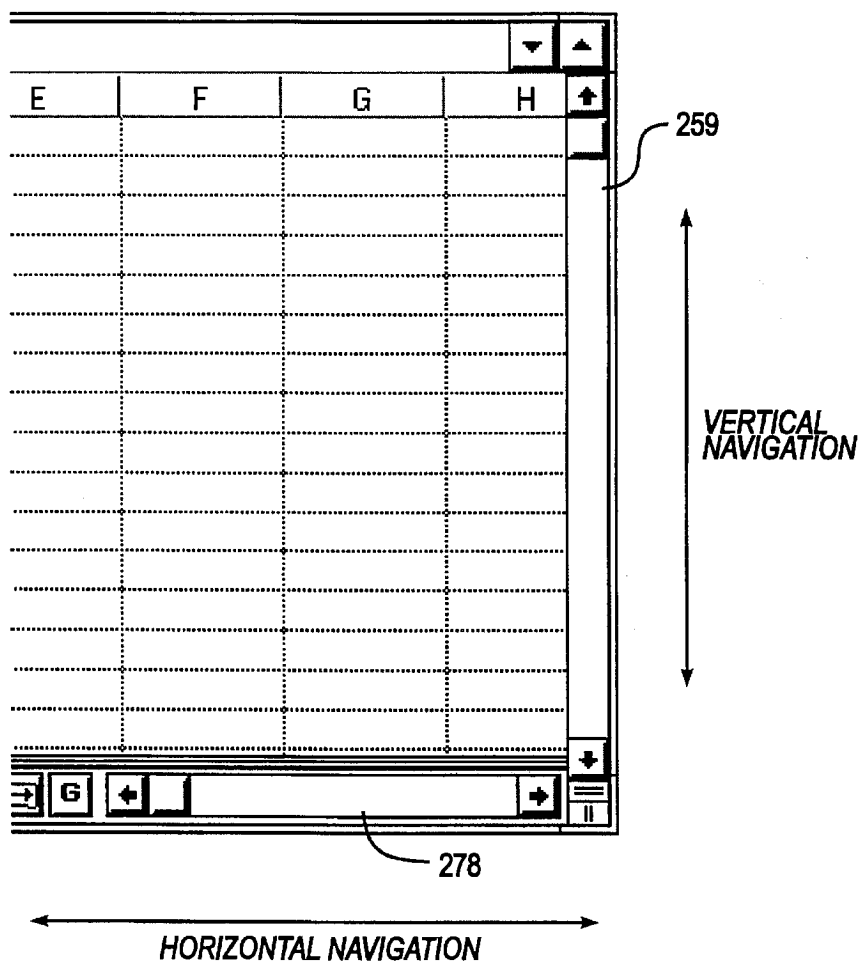
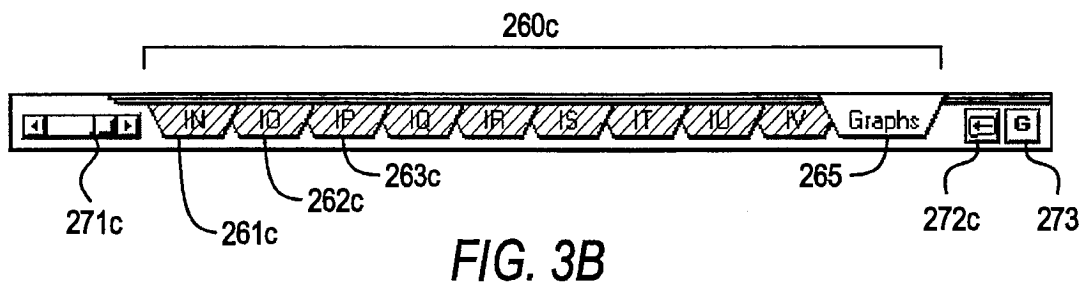
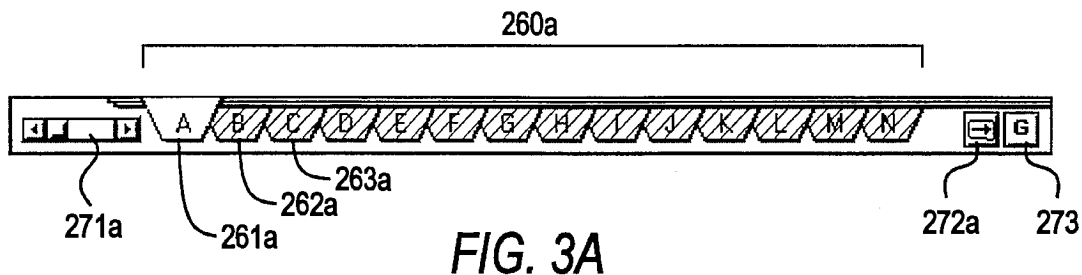


FIG. 2C



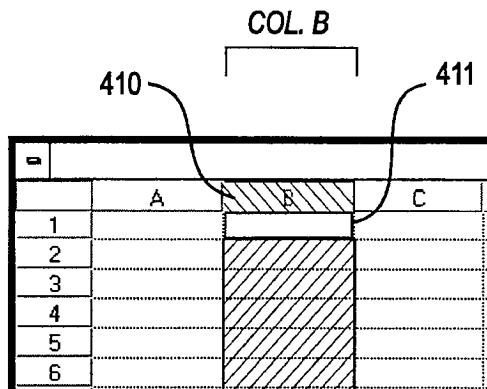


FIG. 4A

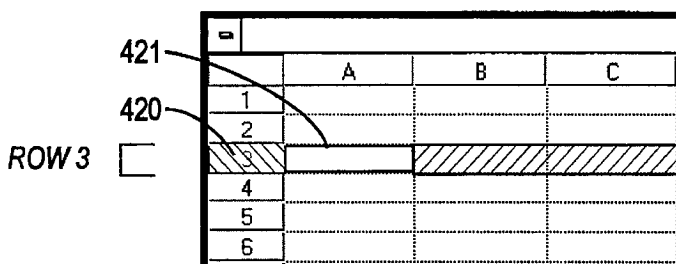


FIG. 4B

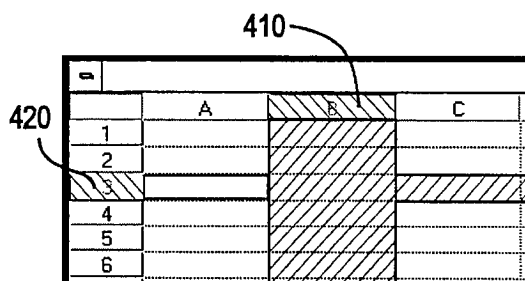


FIG. 4C

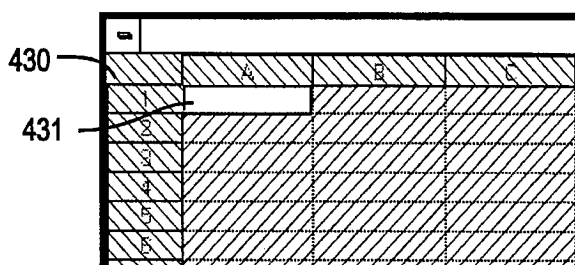


FIG. 4D

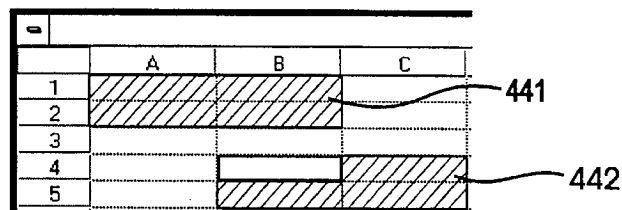


FIG. 4E

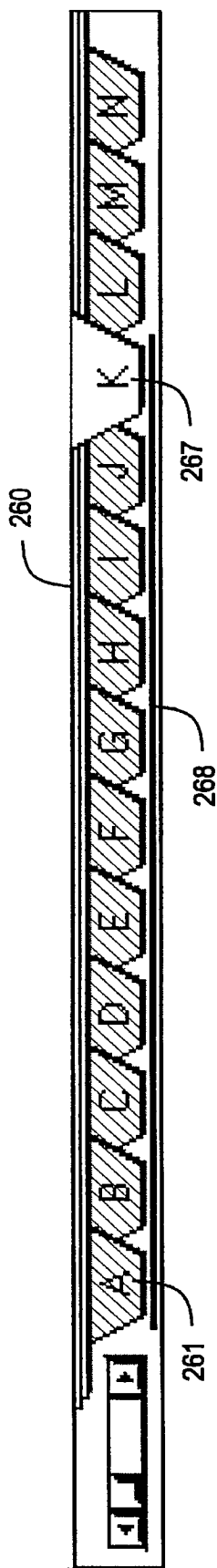


FIG. 4F

Quattro Pro for Windows - PROFITS.WB1

File Edit Block Data Tools Graph Property Window Help

cut copy paste ins del a..z Σ

Salads:B4 1

	A	B	C	D	E	F	G	H	I	J
1	PROFITS: SALADS									
2										
3		Large Caesar	Tossed	Spinach		Total Cost				
4		\$1.00	\$0.50	\$0.75	(1)	\$2.25				
5		Menu Price	\$2.75	\$2.75						
6		Profit	\$1.75	\$2.00						
7		% Profit	275.00%	366.67%	(2)					
8										
9										
10										
11		Avg Profit	2							
12		% Profit	397.22%	Last Updated						
13										
14										
15										
16										
17										
18										
19										
20										
21						\$1.00	\$0.50	\$0.75		
22						\$2.75	\$2.75	\$2.75		
23						\$1.75	\$2.25	\$2.00		
24						275.00%	550.00%	366.67%		
25										
26										
27										
28										
29										
30										

Salads Entrees Desserts

READY

FIG. 4G

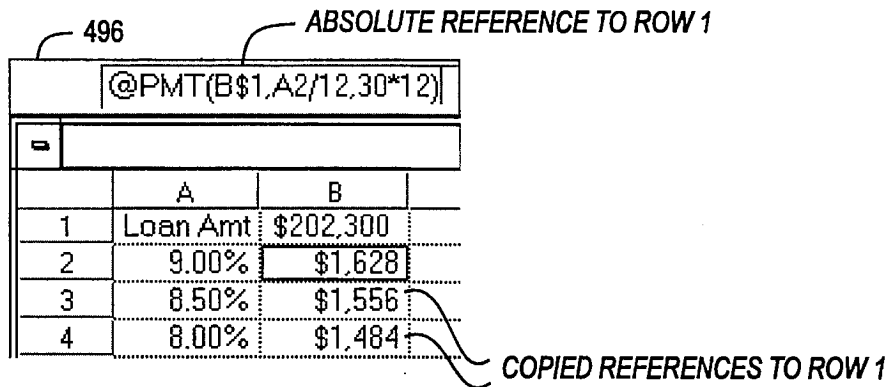


FIG. 4H

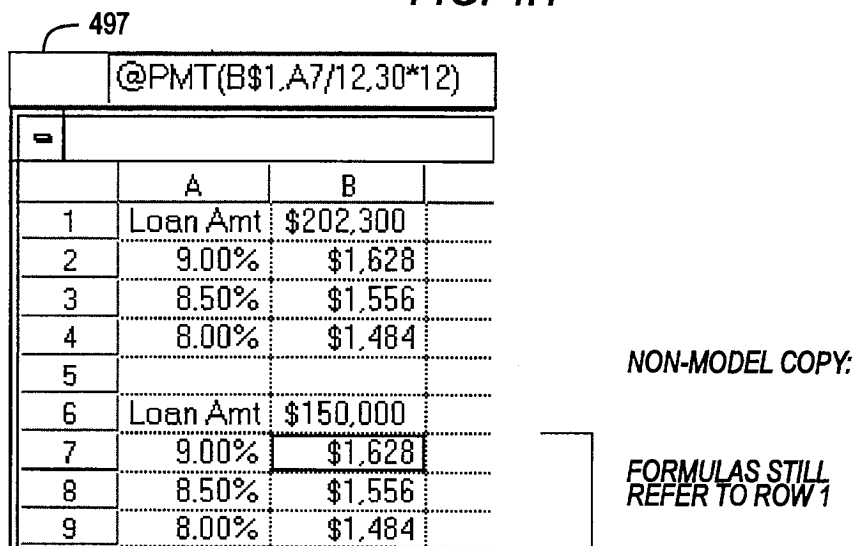


FIG. 4I

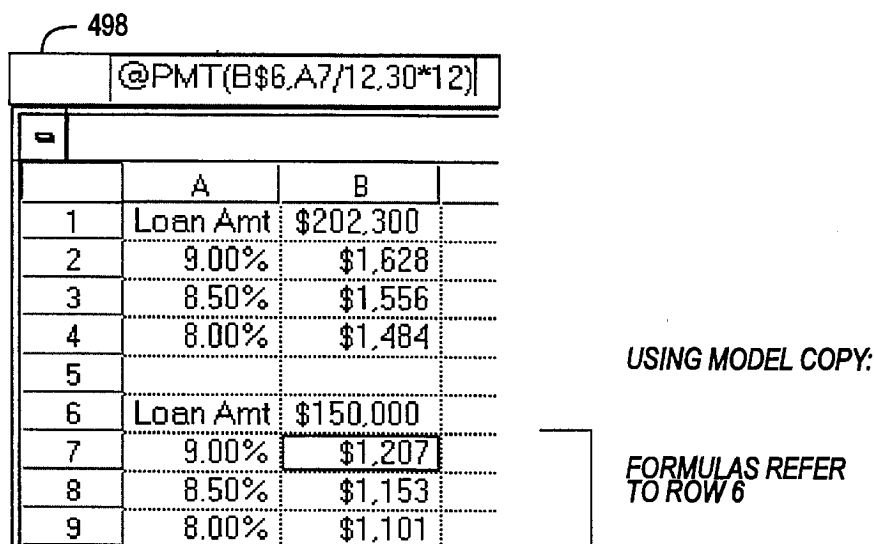


FIG. 4J

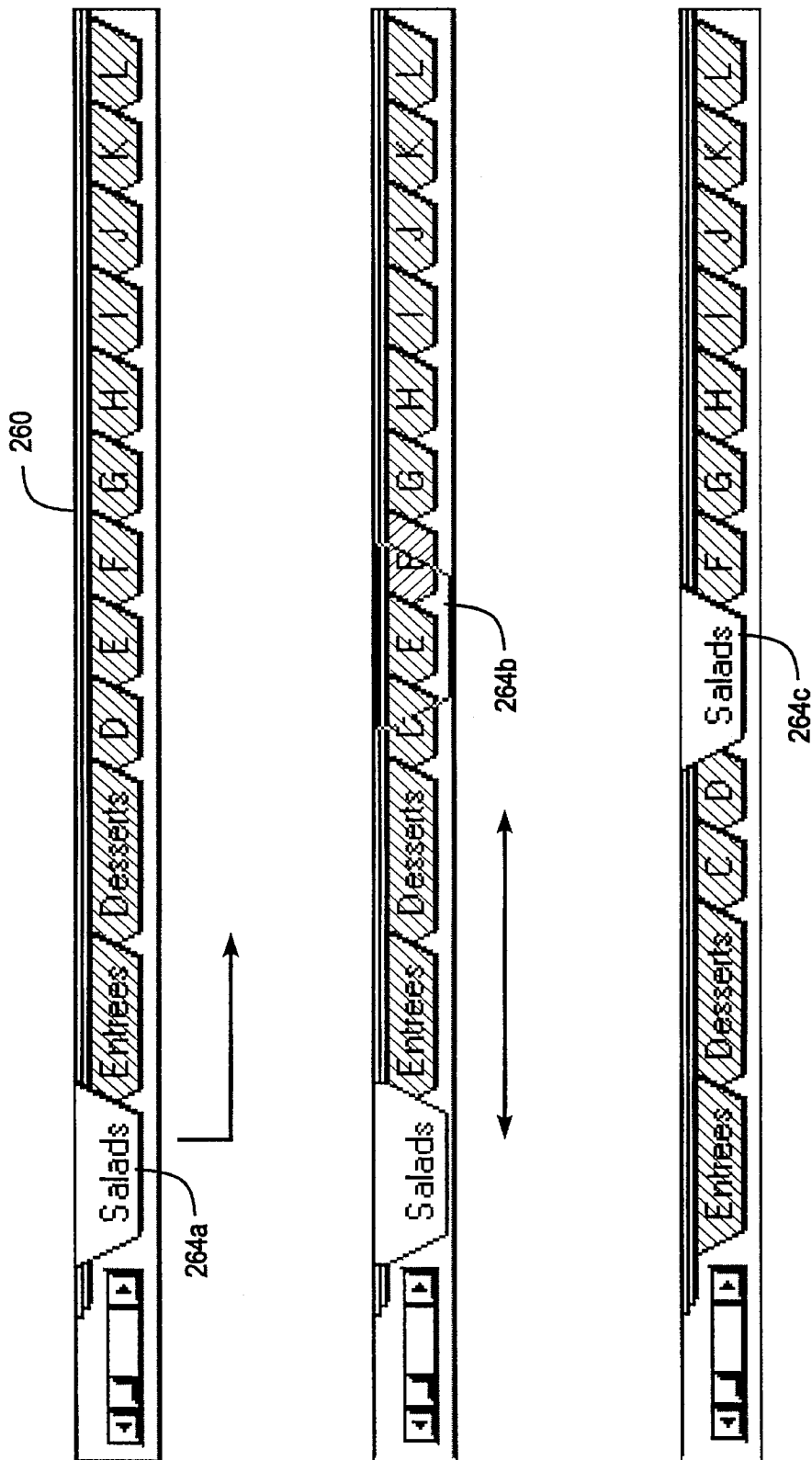


FIG. 4K

300

COL. B

STONEPG.WB1						
	A	B	C	D	E	F
1			Current	Back		
2	Part No.	Orders	Inventory	Orders		
3	P5-724-AB01	724	1001	0		
4	P5-801-AA02	134	85	49		
5	D5-714-AA04	267	250	17		
6	D7-824-AB09	340	340	0		
7						
8	Name	Address	City	ST	ZIP	
9	Roberta Alfred	5843 County Rd	Bloomdale	MS	29548	
10	Adam Stackable	990 Middlefiled St	Louisville	IN	30929	
11	Jerry Hurtado	224 Handley Dr	Canyon	AZ	12553	
12	Mel Grant	155 Miguel St	Atlanta	GA	30309	
13	Betty Rogers	629 Powers Dr	Kankakee	IL	60901	
14	Joyce Lupinetti	399 Glenview Way	Harrisburg	PA	13099	
15	Tom Jueneman	8398 Inlet Rd	La Jolla	CA	92651	

FIG. 4L

350

COL. B₁

SITWOPGS.WB1:1				
	A	B	C	D
1			Current	Back
2	Part No.	Orders	Inventory	Orders
3	P5-724-AB01	724	1001	0
4	P5-801-AA02	134	85	49
5	D5-714-AA04	267	250	17
6	D7-824-AB09	340	340	0
7				

351

352

COL. B₂

SITWOPGS.WB1:2					
	A	B	C	D	E
1	Name	Address	City	ST	ZIP
2	Roberta Alfred	5843 County Rd	Bloomdale	MS	29548
3	Adam Stackable	990 Middlefiled St	Louisville	IN	30929
4	Jerry Hurtado	224 Handley Dr	Canyon	AZ	12553
5	Mel Grant	155 Miguel St	Atlanta	GA	30309
6	Betty Rogers	629 Powers Dr	Kankakee	IL	60901
7	Joyce Lupinetti	399 Glenview Way	Harrisburg	PA	13099
8	Tom Jueneman	8398 Inlet Rd	La Jolla	CA	92651

FIG. 4M

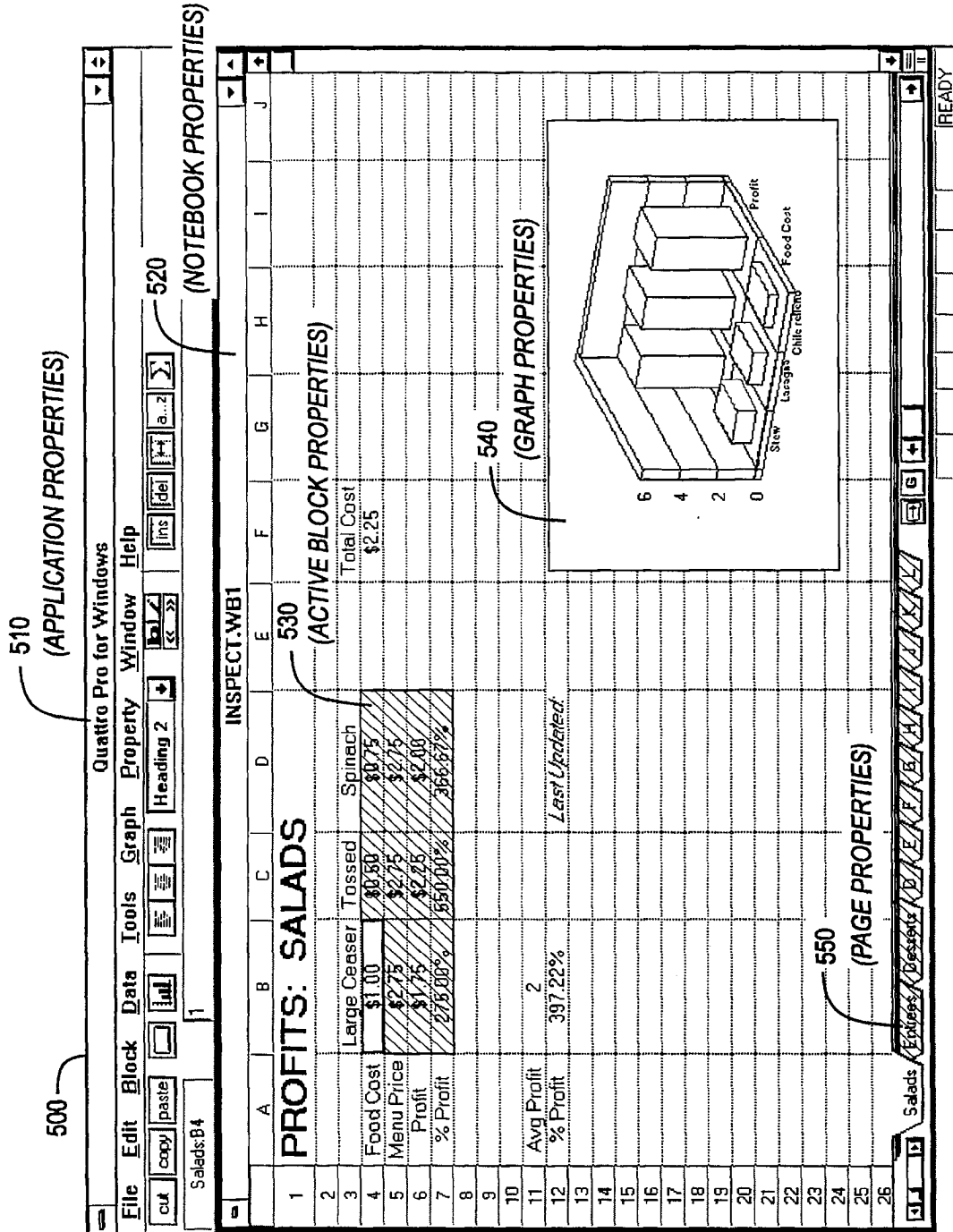


FIG. 5A

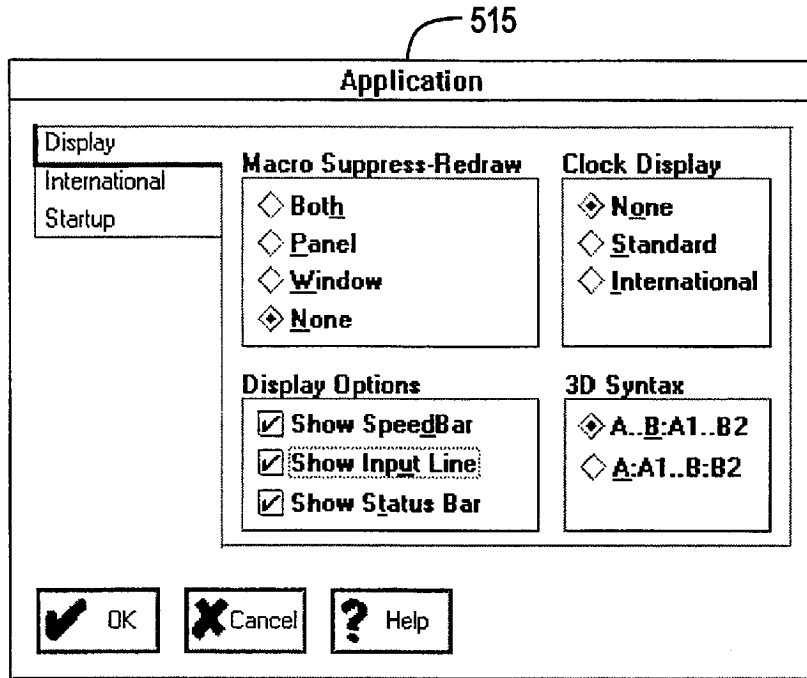


FIG. 5B

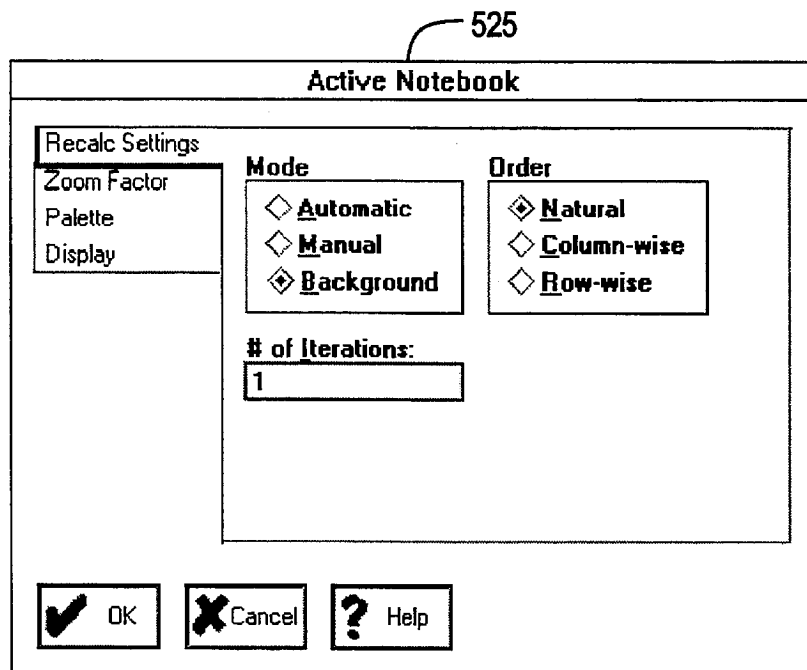


FIG. 5C

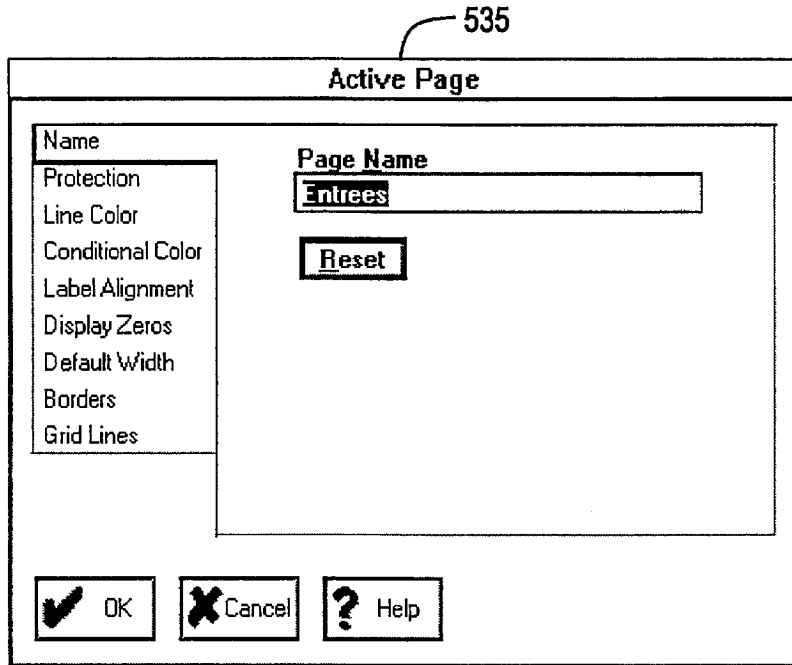


FIG. 5D

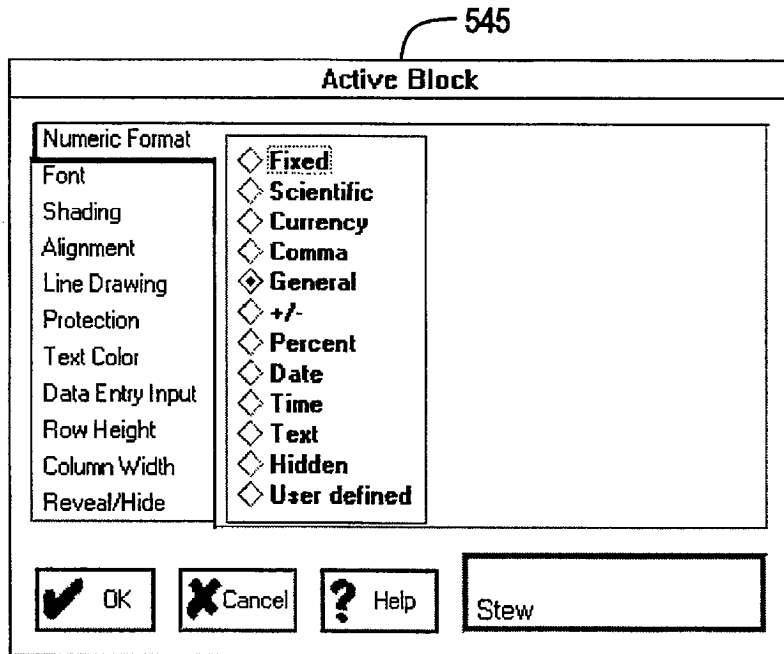


FIG. 5E

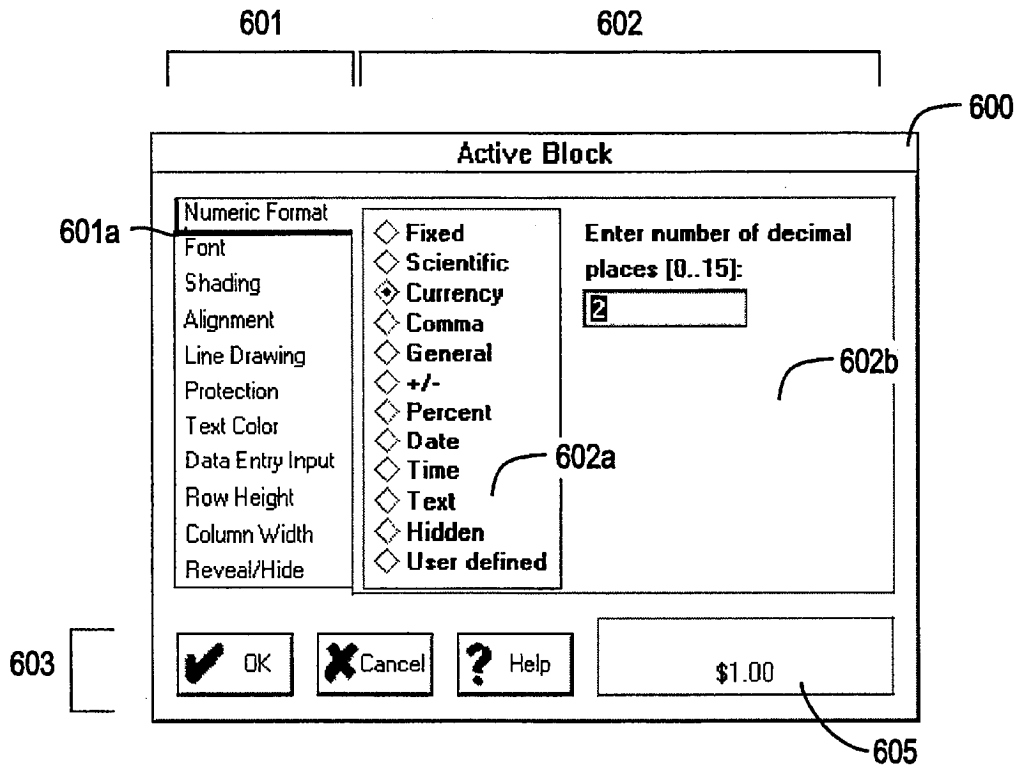


FIG. 6A

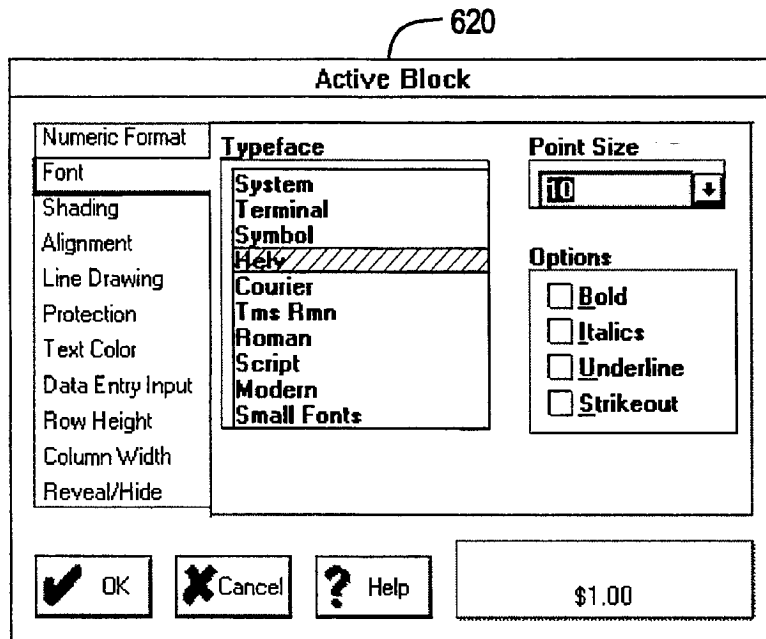


FIG. 6B

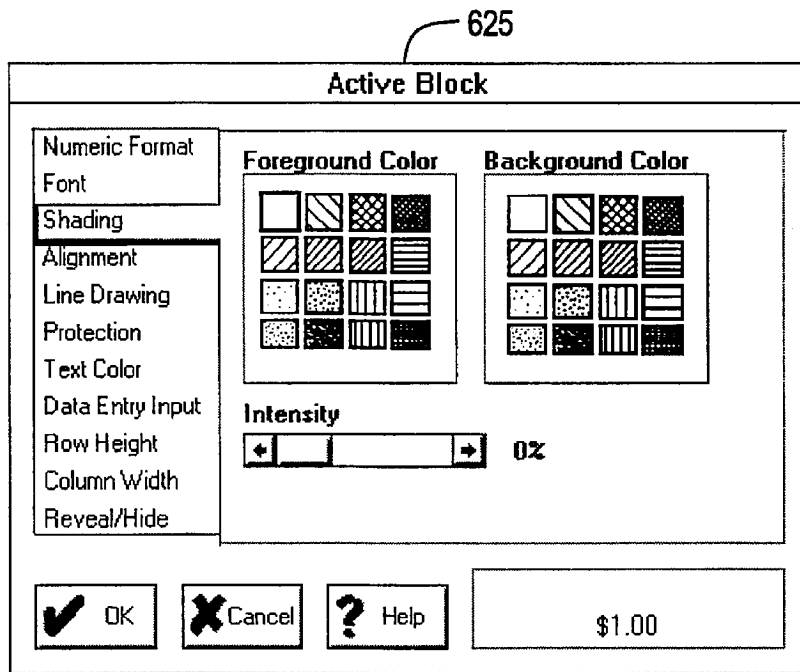


FIG. 6C

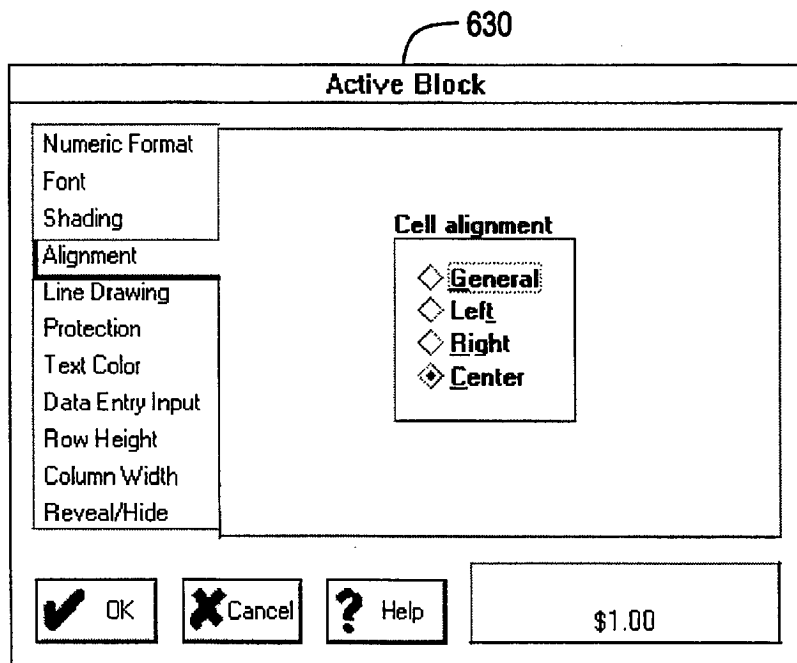


FIG. 6D

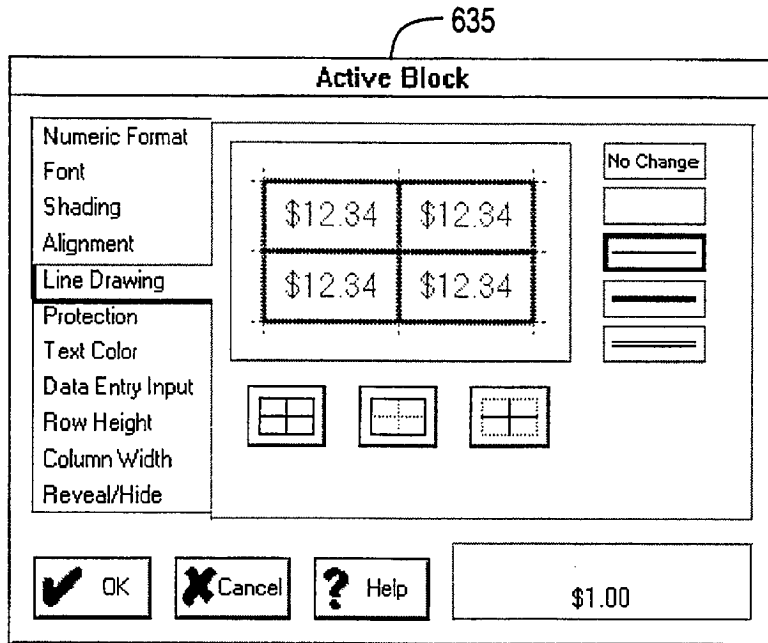


FIG. 6E

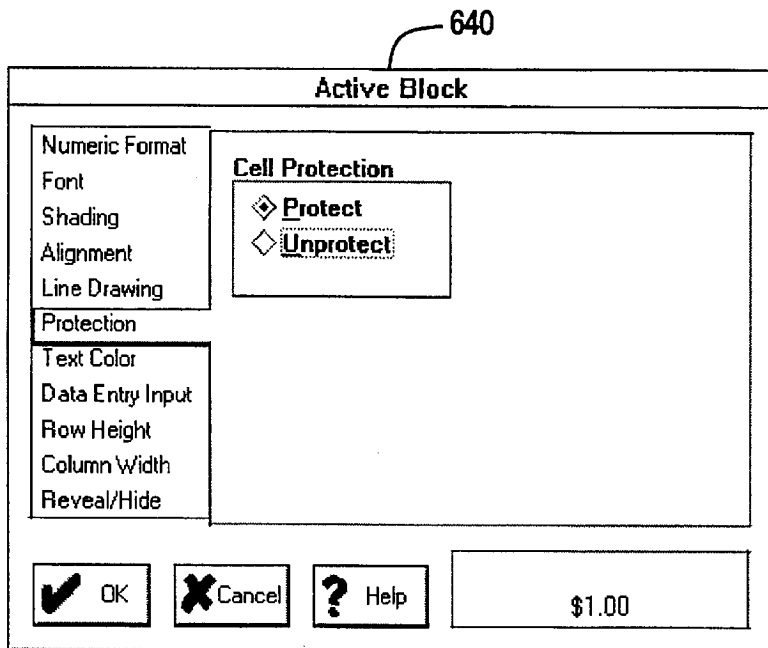


FIG. 6F

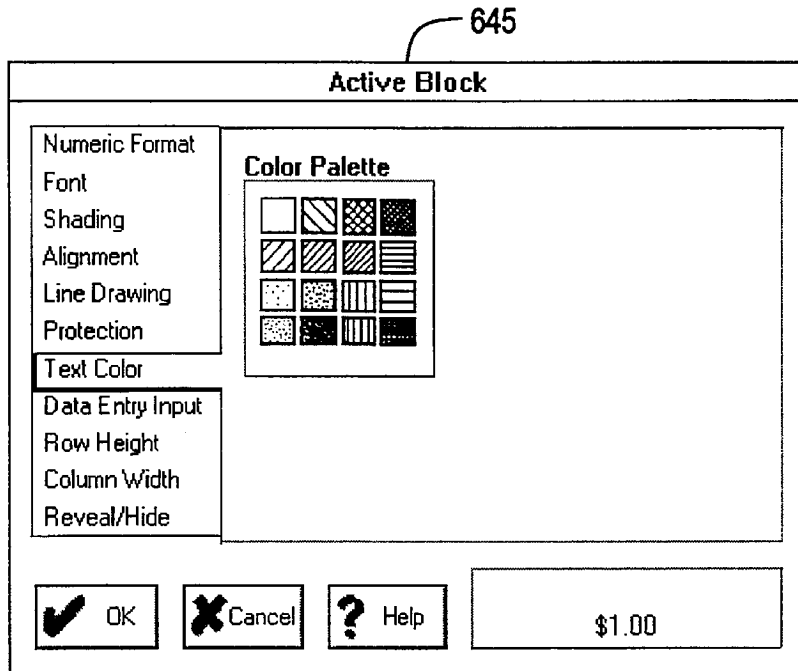


FIG. 6G

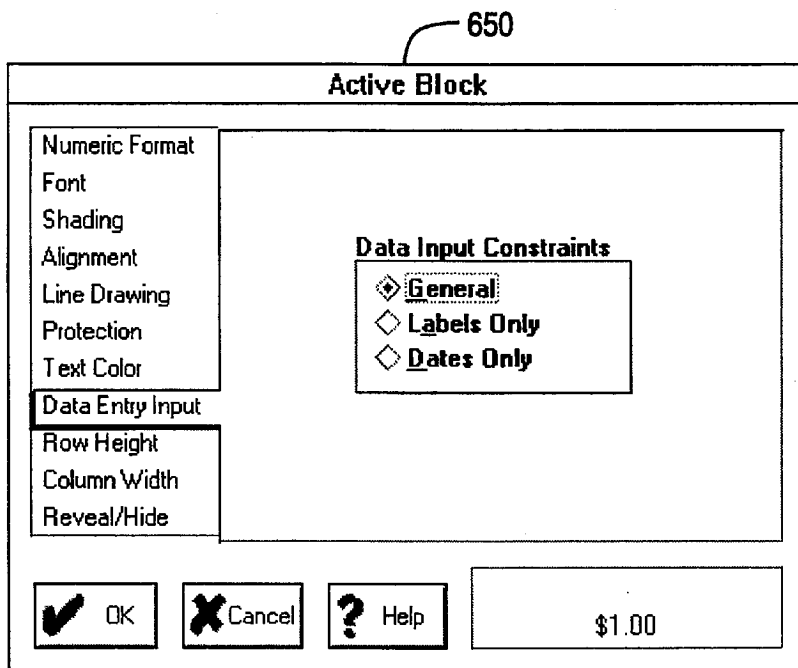


FIG. 6H

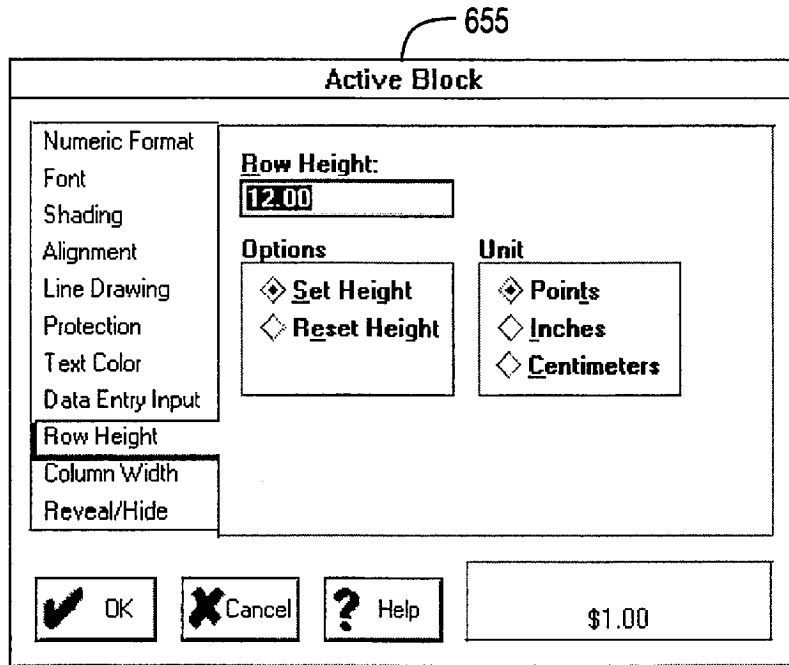


FIG. 6I

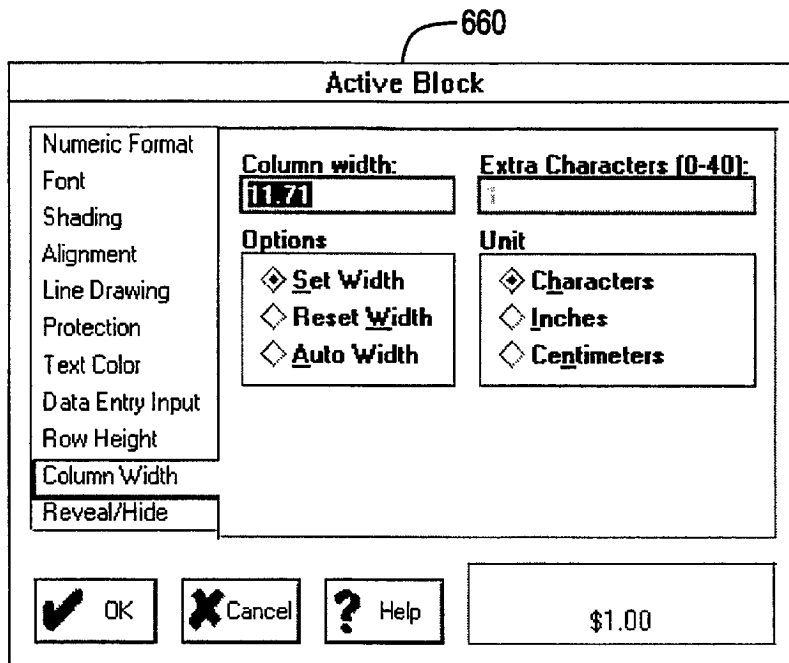


FIG. 6J

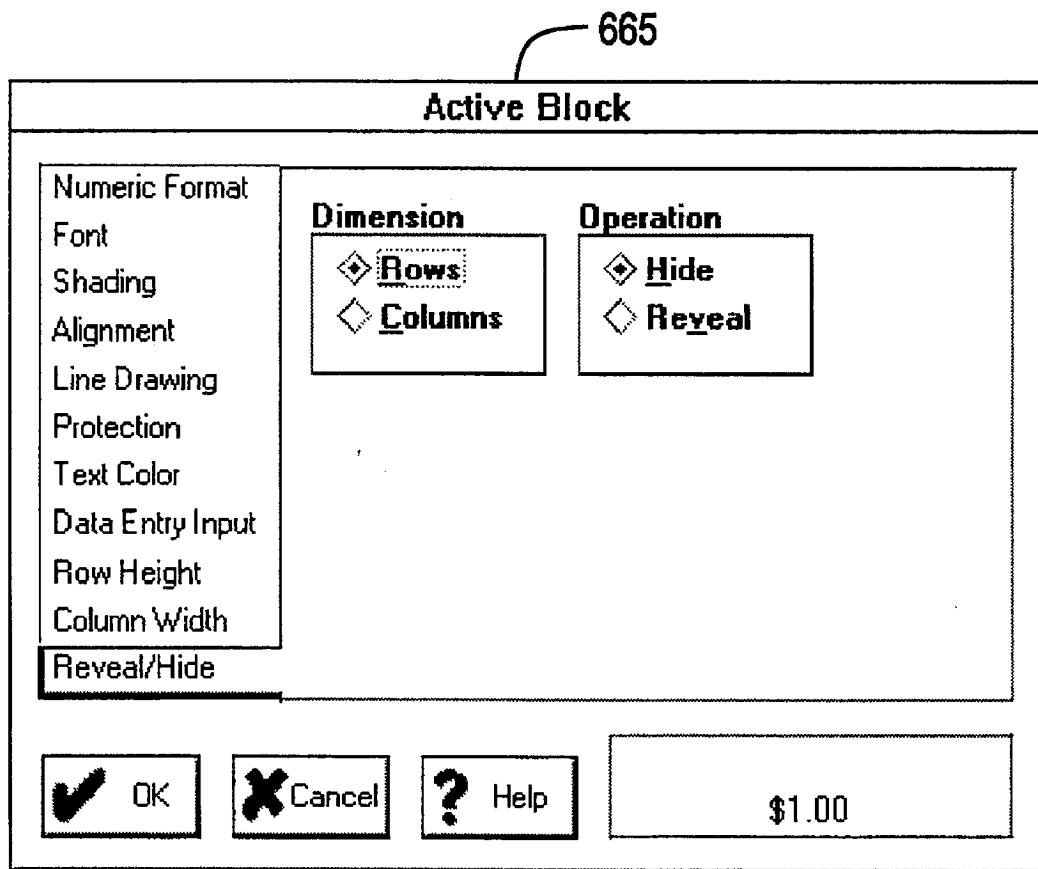


FIG. 6K

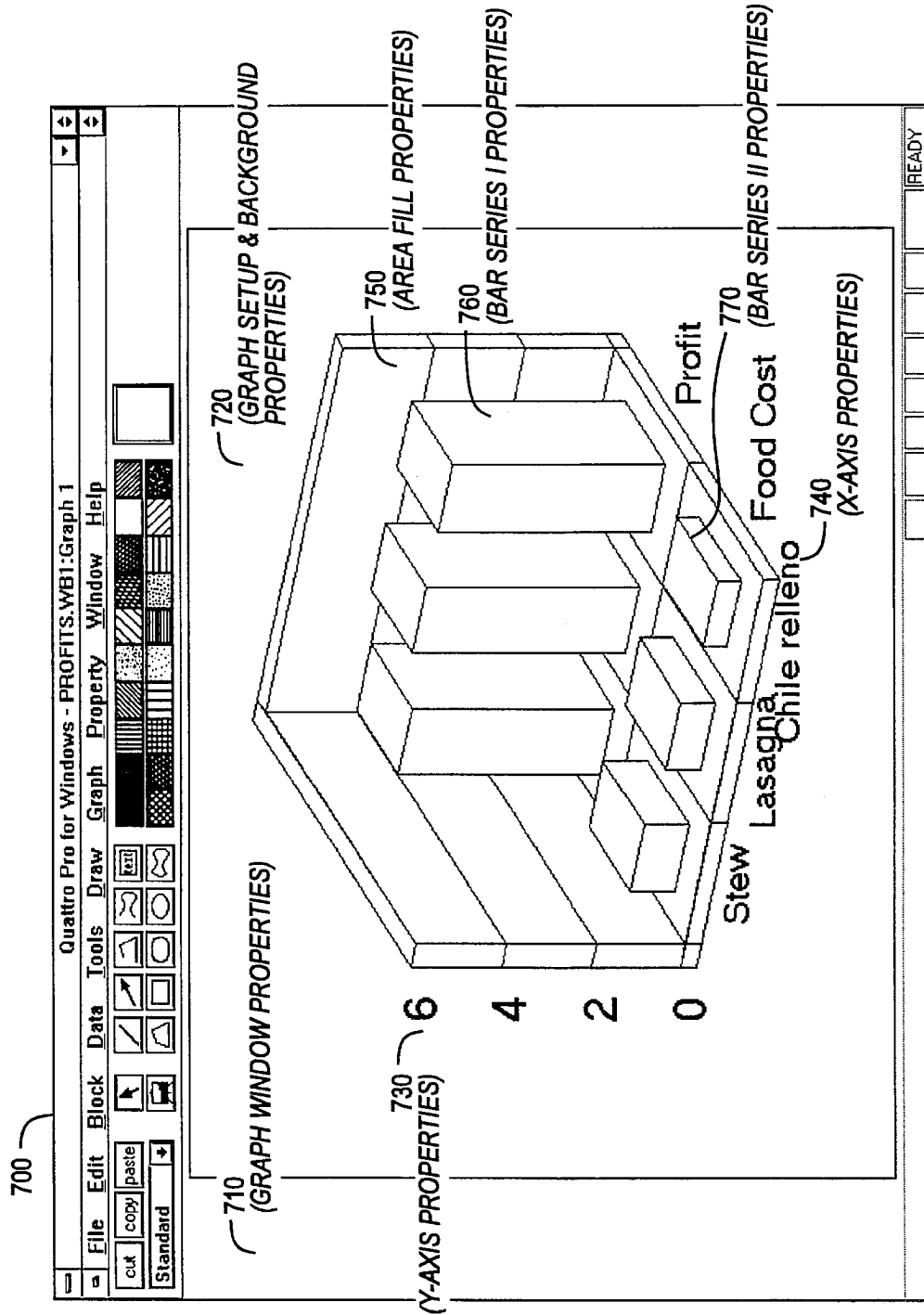


FIG. 7A

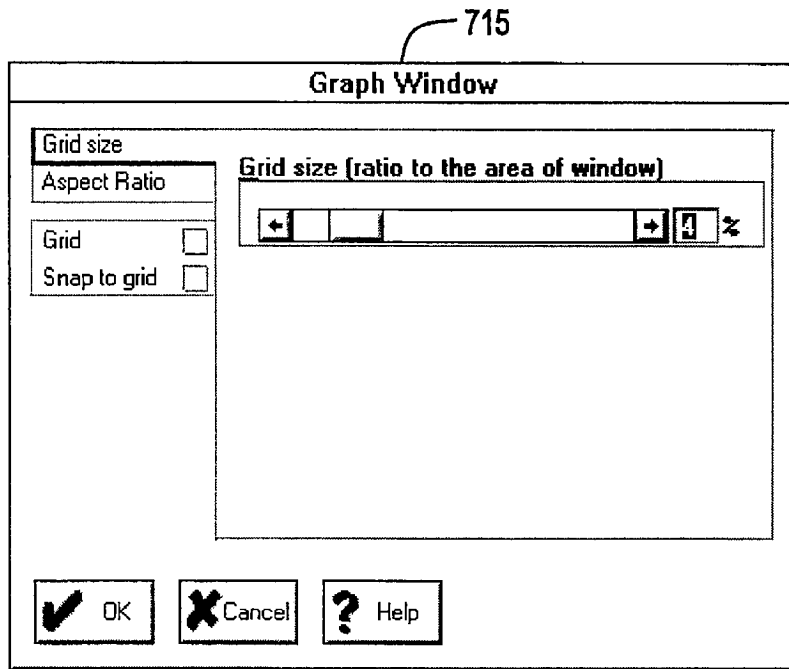


FIG. 7B

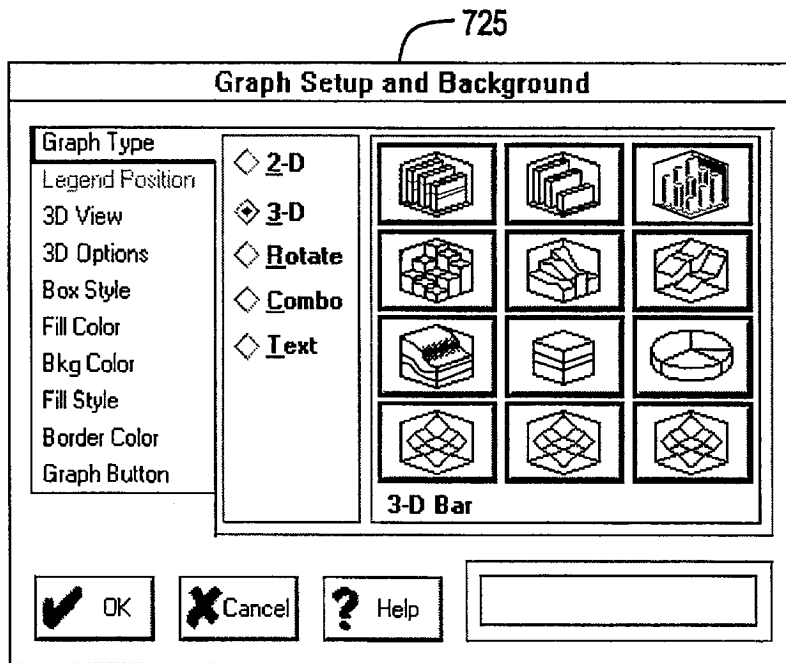


FIG. 7C

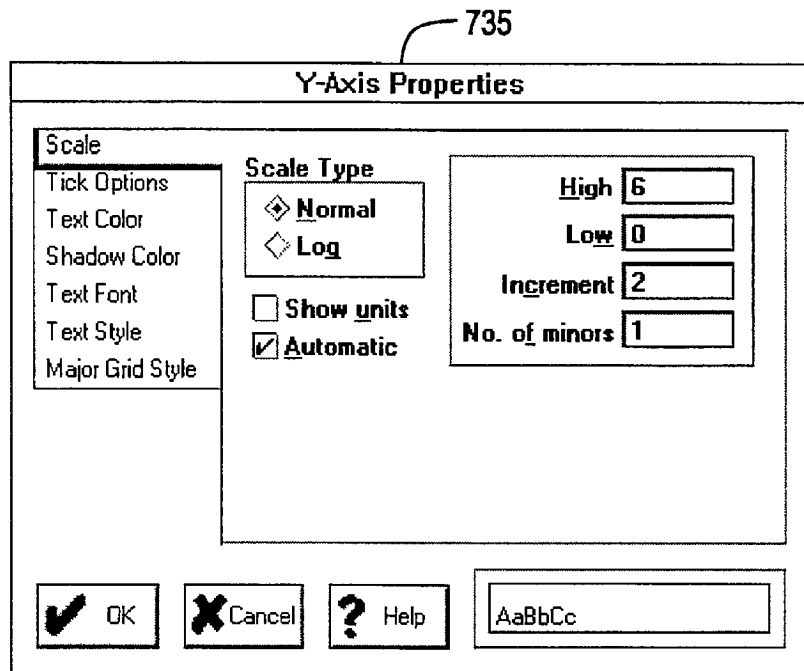


FIG. 7D

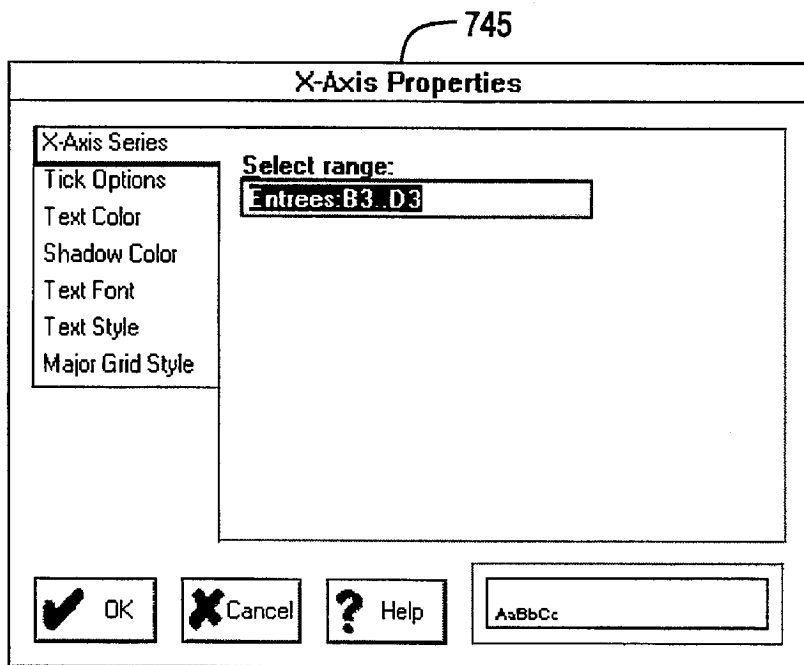


FIG. 7E

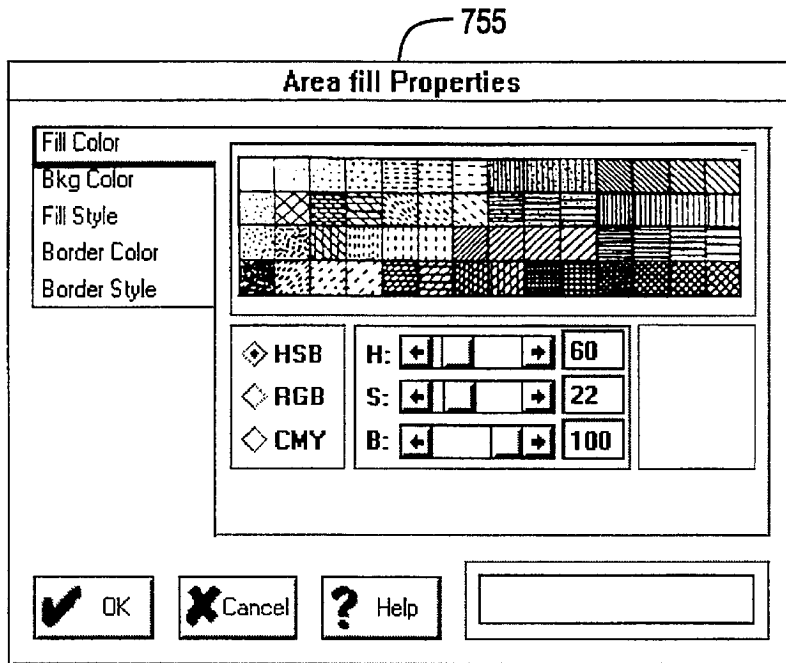


FIG. 7F

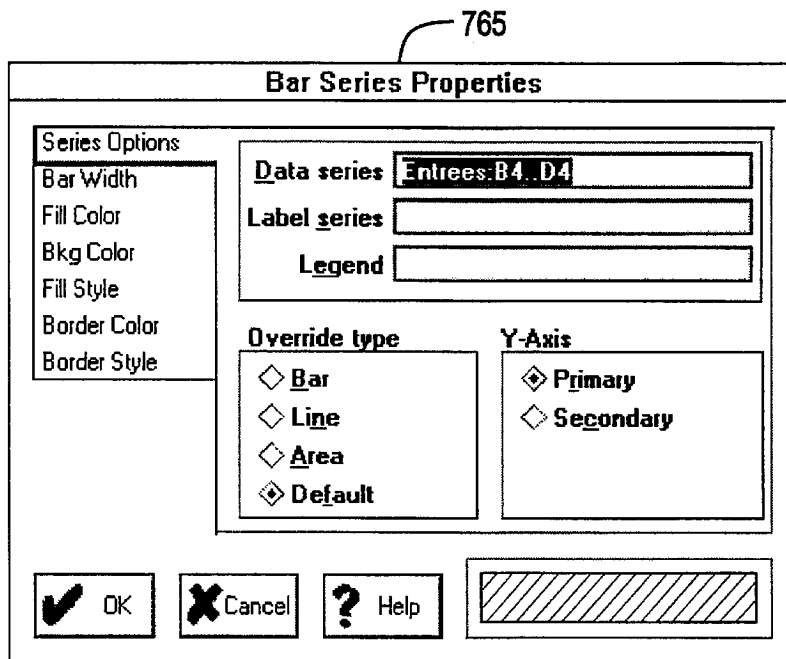


FIG. 7G

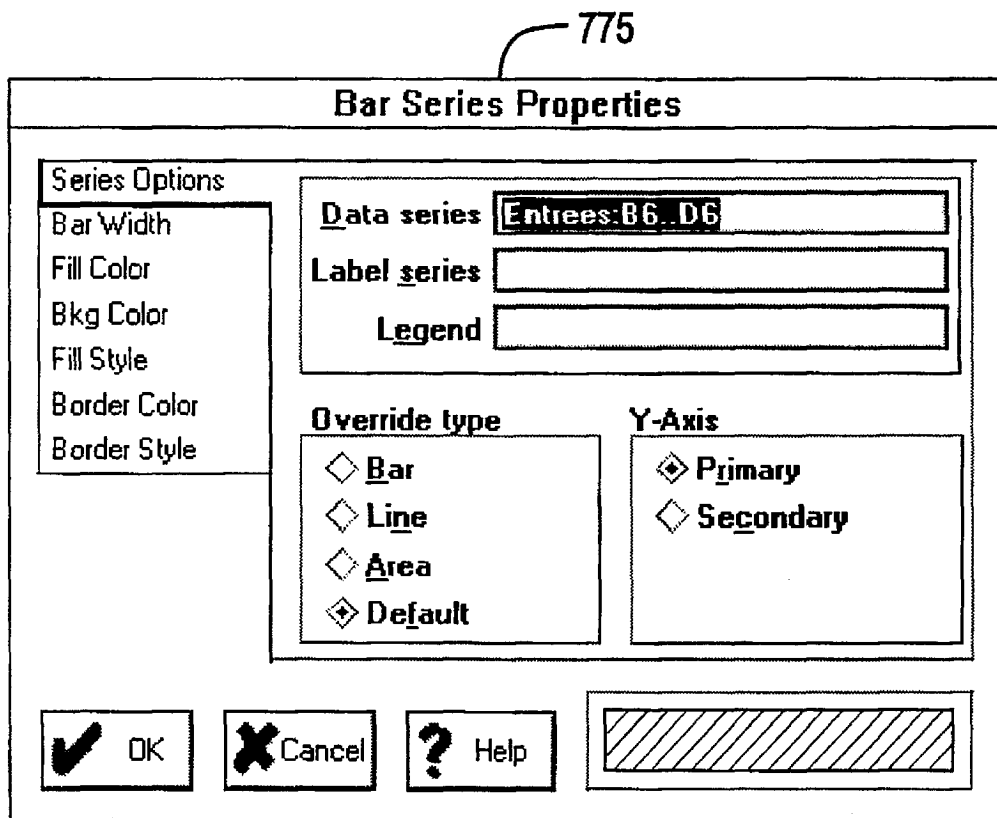


FIG. 7H

800

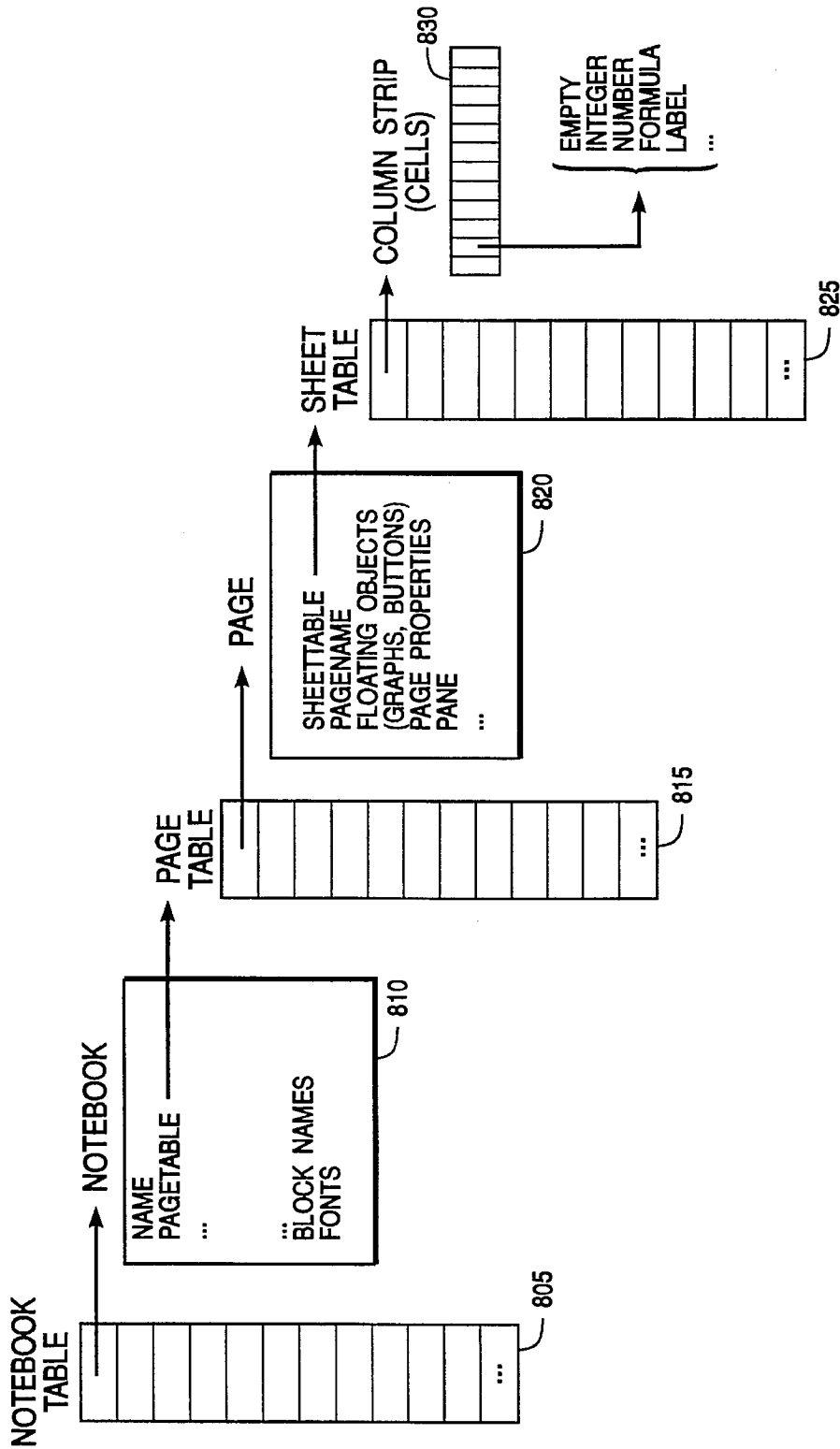
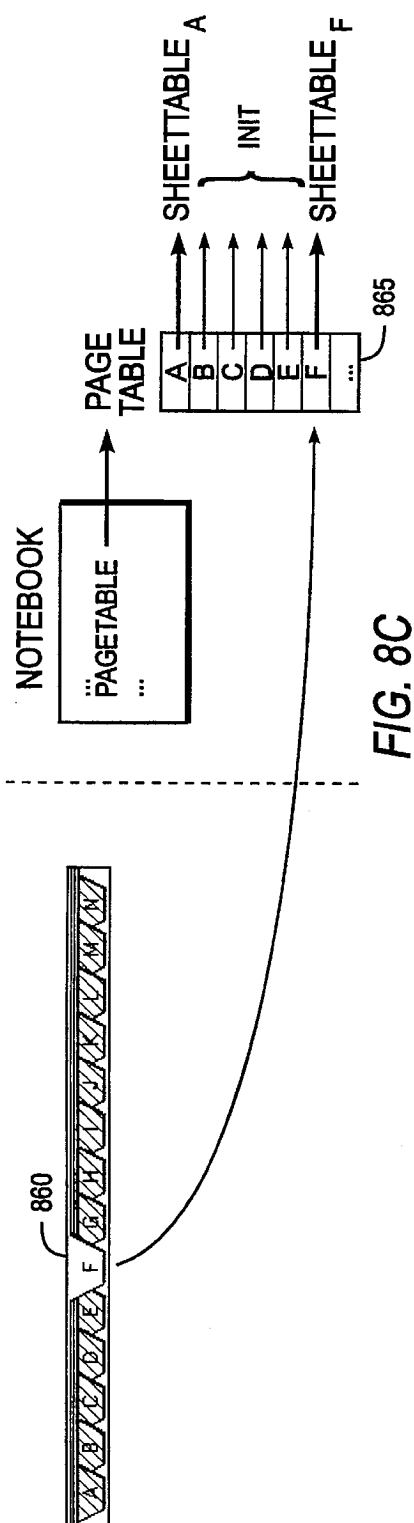
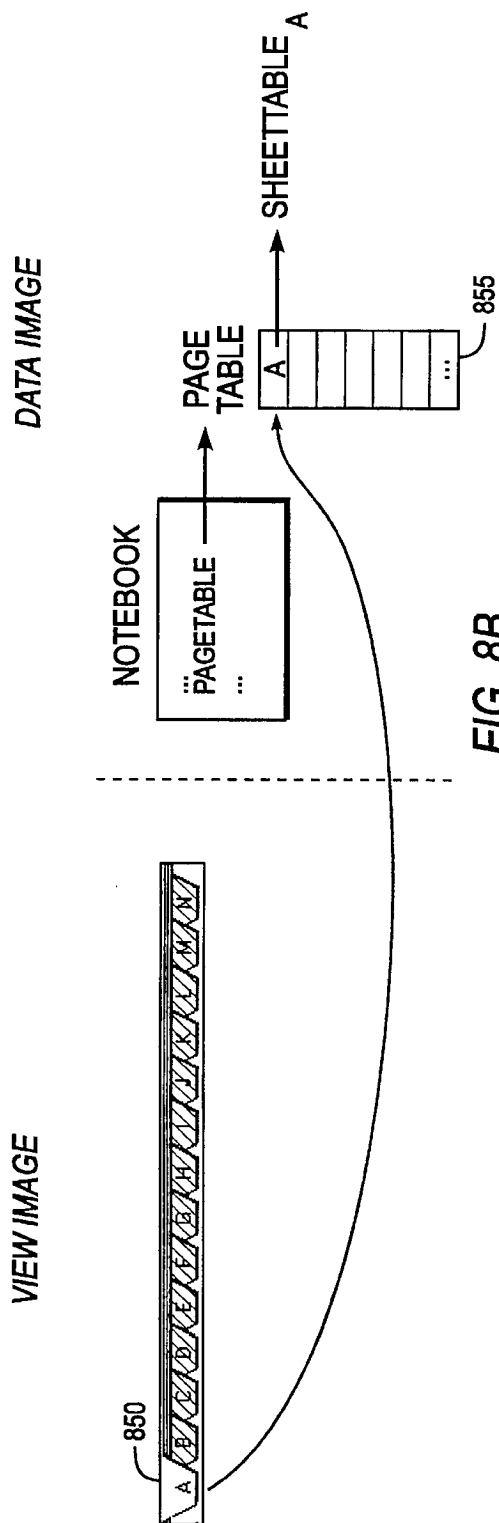


FIG. 8A



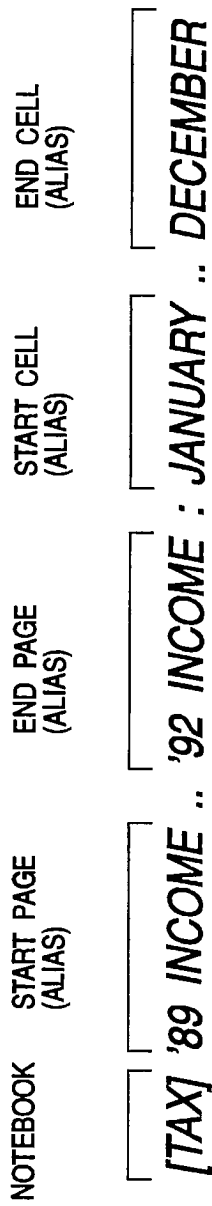


FIG. 8D

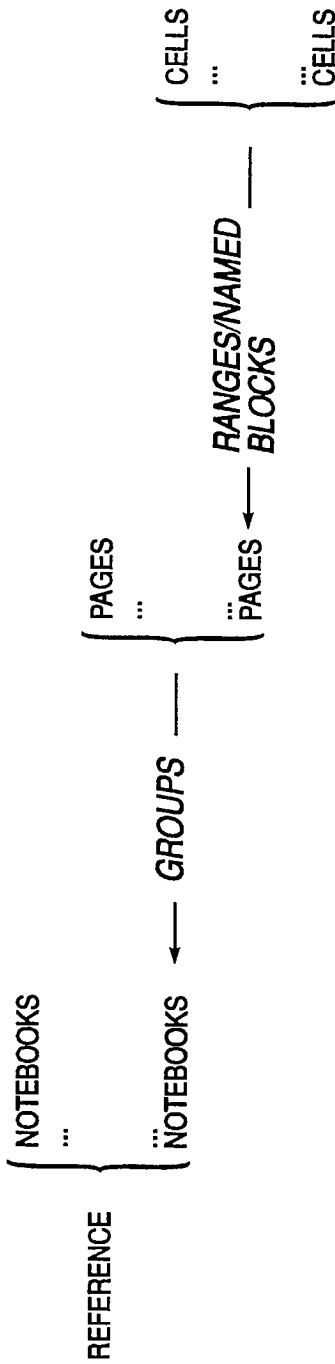


FIG. 8E



FIG. 8F

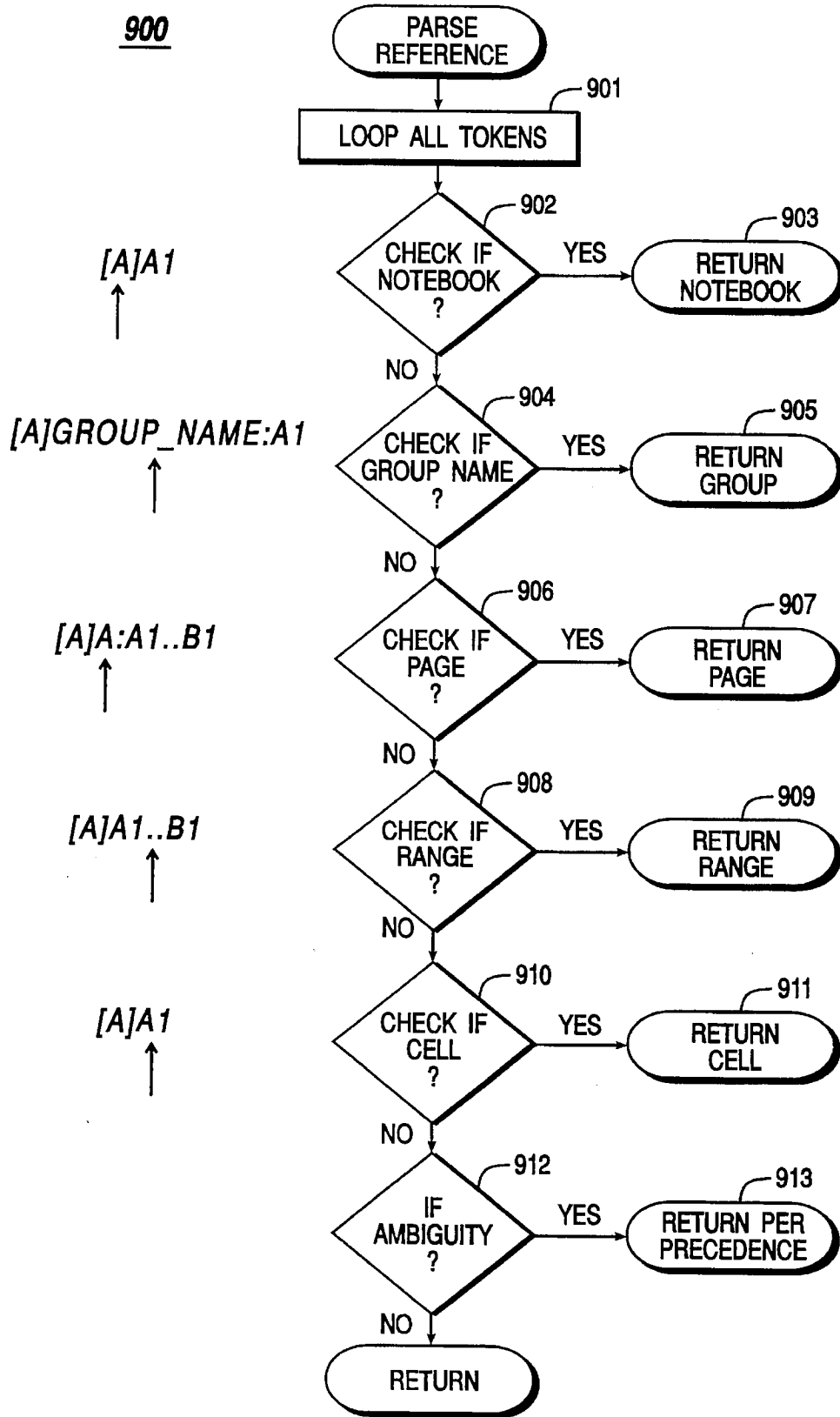


FIG. 9A

920

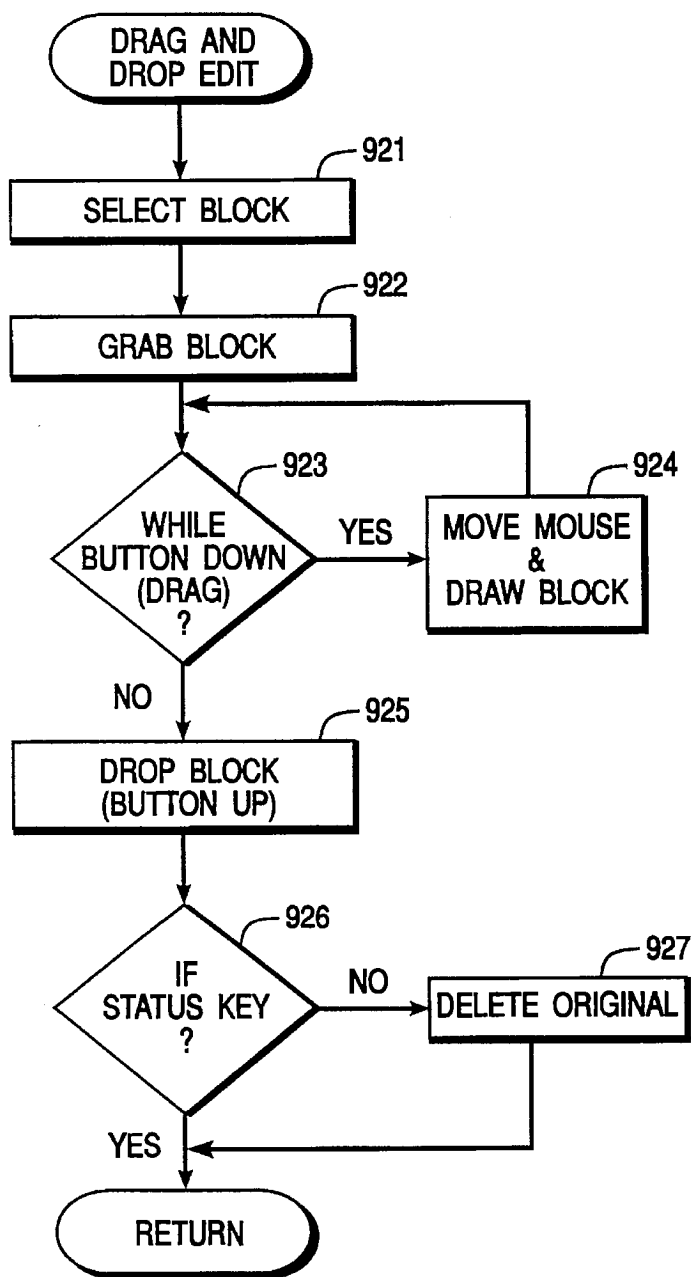


FIG. 9B

940

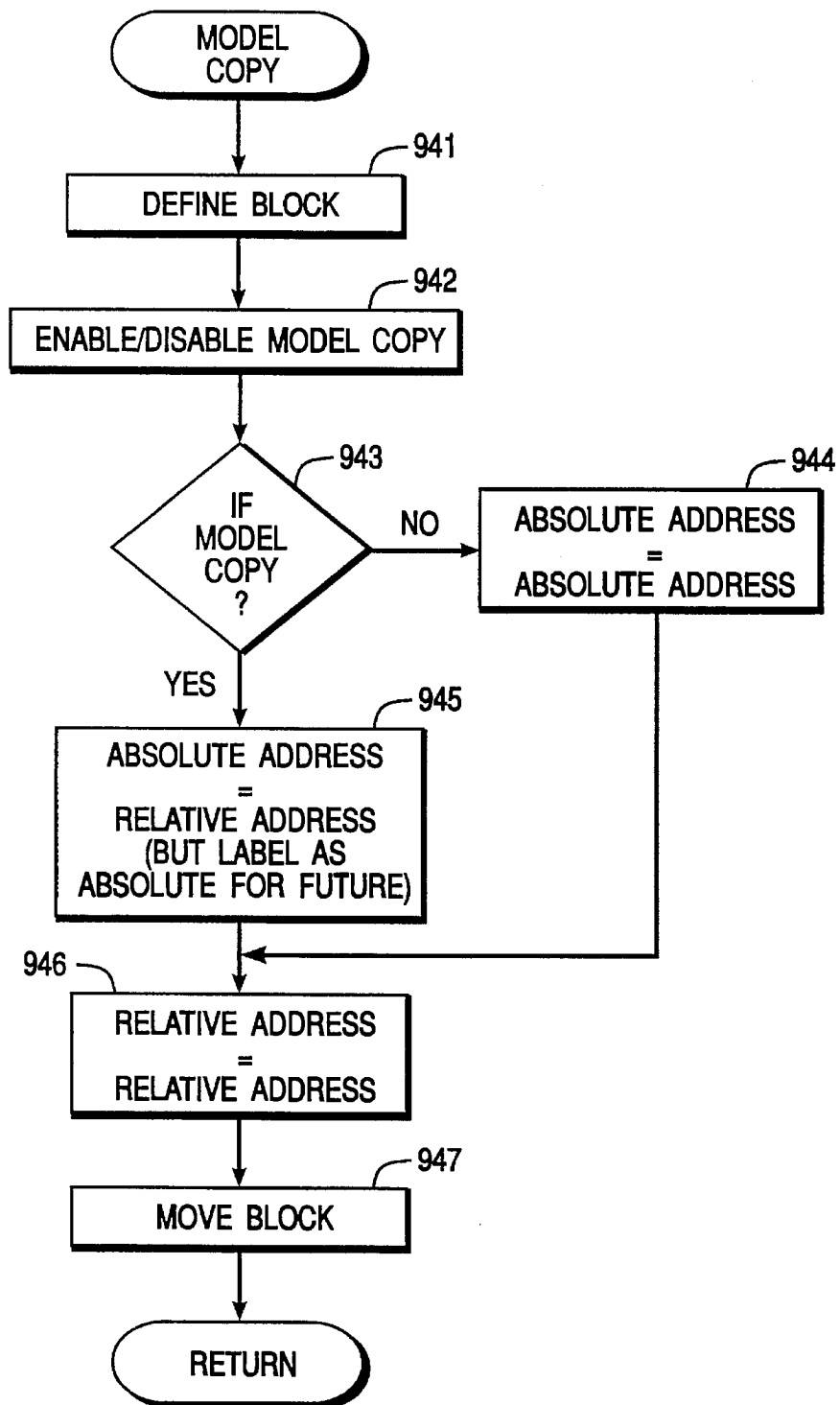


FIG. 9C

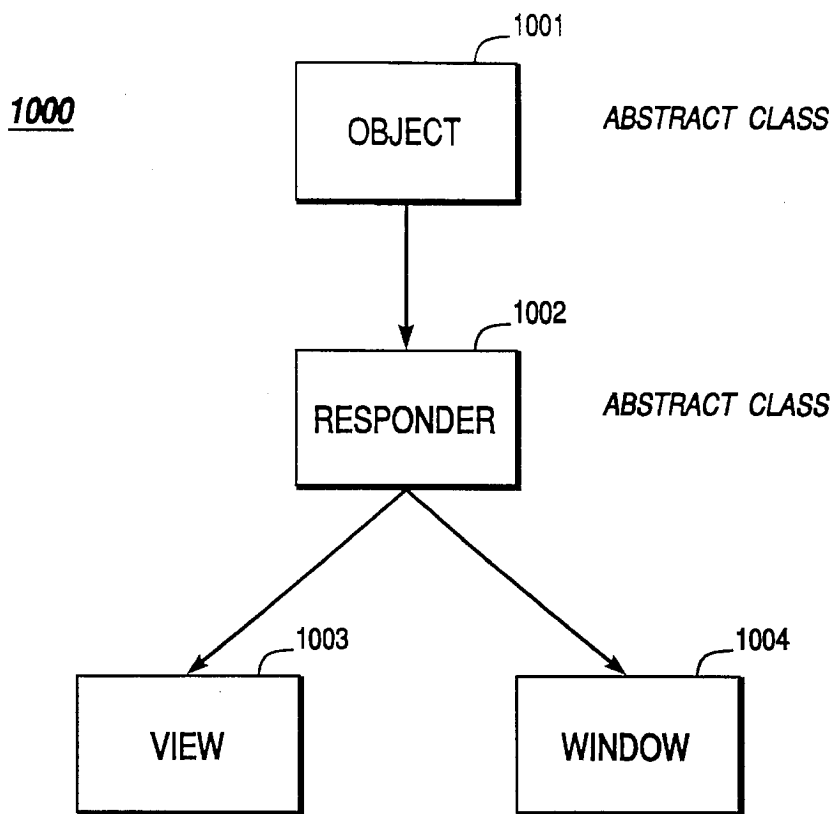


FIG. 10A

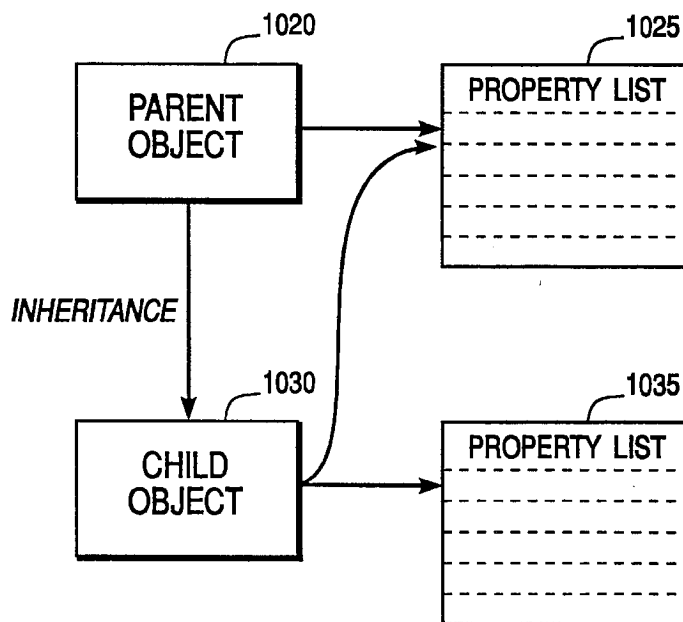


FIG. 10B

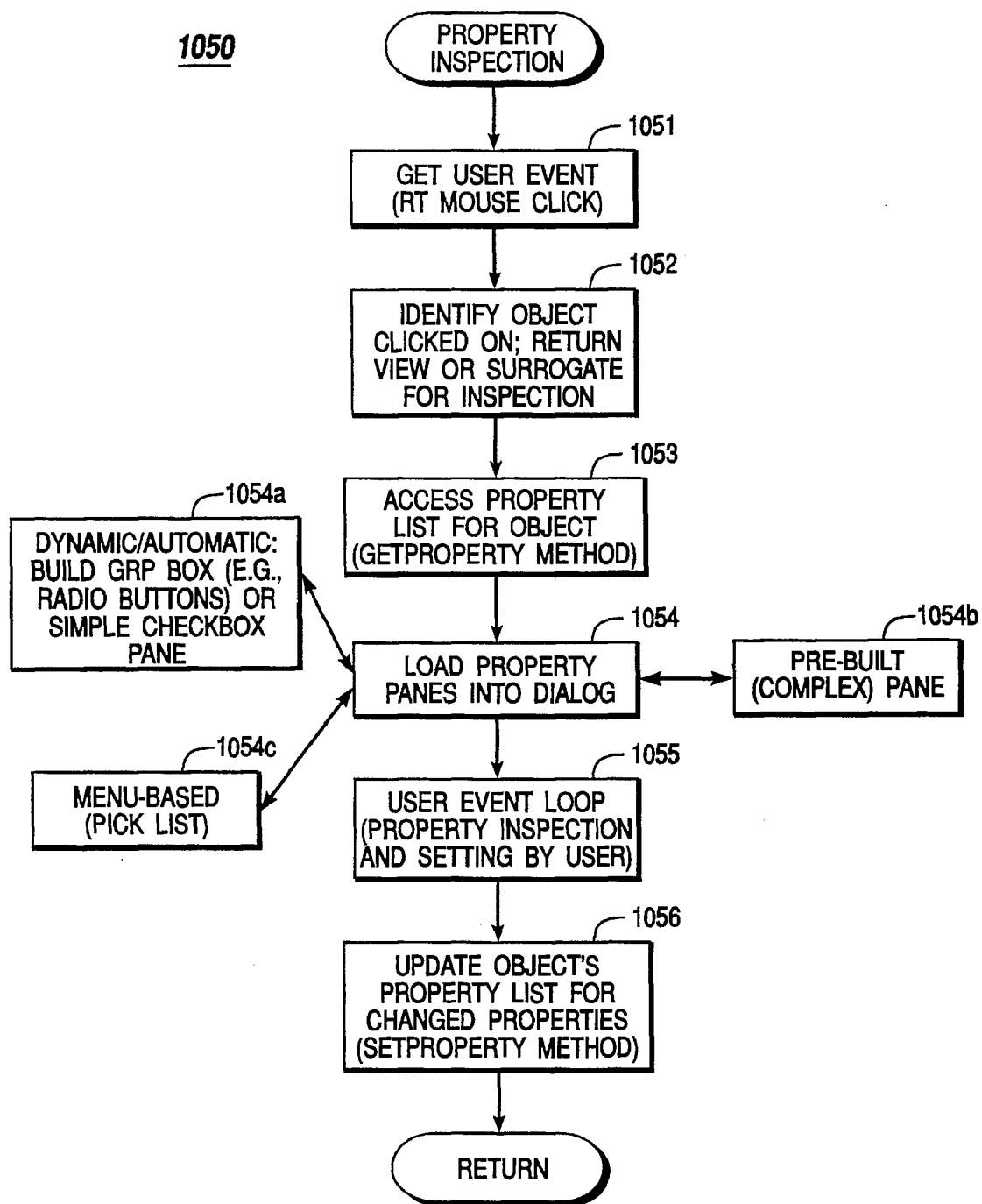


FIG. 10C

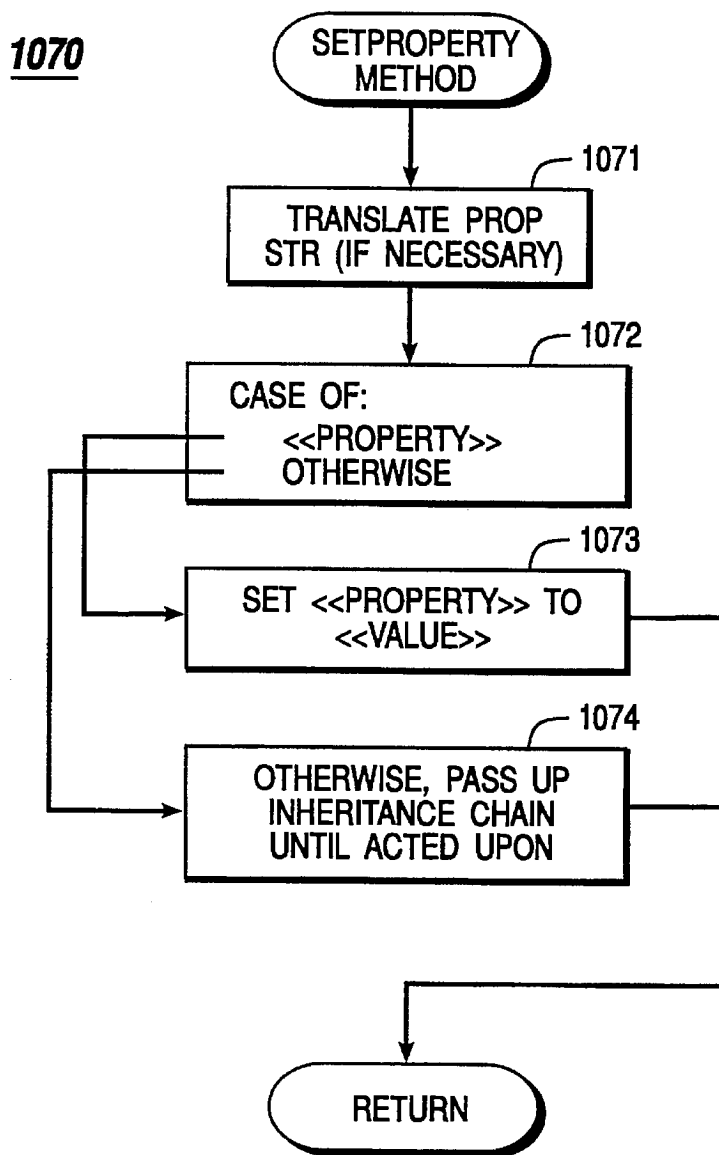


FIG. 10D

5,590,259

1

SYSTEM AND METHODS FOR IMPROVED SPREADSHEET INTERFACE WITH USER-FAMILIAR OBJECTS

The present application is a continuation of U.S. application No. 07/866,658, filed Apr. 8, 1992, now U.S. Pat. No. 5,416,895, issued May 16, 1995.

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

The present invention relates generally to the field of information processing by digital computers and, more particularly, to the processing and presentation of information by program applications, particularly electronic spreadsheets.

Before computers, numerical analyses, particularly financial ones, were usually prepared on an accountant's columnar pad or spreadsheet, with pencil and calculator in hand. By organizing data into columns and rows, spreadsheets afford the rapid assimilation of information by a reader. The task of preparing a spreadsheet on paper, however, is not quite so fast. Instead, the process tends to be very slow, as each entry must be tediously calculated and entered into the spreadsheet. Since all calculations are the responsibility of the preparer, manually prepared spreadsheets are also prone to errors. Hence, preparation of spreadsheets by hand is slow, tedious, and unreliable.

With the advent of microcomputers, a solution was forthcoming in the form of "electronic spreadsheets." Better known simply as "spreadsheets," these software programs provide a computerized replacement for the traditional financial modeling tools: the accountant's columnar pad, pencil, and calculator. In some regards, spreadsheet programs are to those tools what wordprocessors are to typewriters. Spreadsheets offer dramatic improvements in ease of creating, editing, and using financial models.

A typical spreadsheet program configures the memory of a computer to resemble the column/row or grid format of an accountant's columnar pad, thus providing a visible calculator for a user. Because this "pad" exists dynamically in the computer's memory, however, it differs from paper pads in several important ways. Locations in the electronic spreadsheet, for example, must be communicated to the computer in a format which it can understand. A common scheme for accomplishing this is to assign a number to each row in a spreadsheet, and a letter to each column. To reference a location at column A and row 1 (i.e., the upper-left hand corner), for example, the user types in "A1". In this manner, the spreadsheet defines an addressable storage location or "cell" at each intersection of a row with a column.

Data entry into an electronic spreadsheet occurs in much the same manner that information would be entered on an accountant's pad. After a screen cursor is positioned at a desired location, the user can enter alphanumeric information. Besides holding text and numeric information, however, spreadsheet cells can store special instructions or "formulas" specifying calculations to be performed on the

2

numbers stored in spreadsheet cells. In this fashion, cell references can serve as variables in an equation, thereby allowing precise mathematical relationships to be defined between cells. The structure and operation of a spreadsheet program, including advanced functions such as functions and macros, are documented in the technical, trade, and patent literature. For an overview, see e.g., Cobb, S., *Using Quattro Pro 2*, Borland-Osborne/McGraw-Hill, 1990; and LeBlond, G. and Cobb, D., *Using 1-2-3*, Que Corp., 1985. The disclosures of each of the foregoing references are hereby incorporated by reference.

Electronic spreadsheets offer many advantages over their paper counterparts. For one, electronic spreadsheets are much larger (i.e., hold more information) than their paper counterparts; electronic spreadsheets having thousands or even millions of cells are not uncommon. Spreadsheet programs also allow users to perform "what if" scenarios. After a set of mathematical relationships has been entered into a worksheet, the spread of information can be recalculated using different sets of assumptions, with the results of each recalculation appearing almost instantaneously. Performing this operation manually, with paper and pencil, would require recalculating every relationship in the model with each change made.

While electronic spreadsheets offer significant productivity gains in the task of complex data modeling, none has been as intuitive to use as ordinary paper and pencil—objects already familiar to the user. Instead, the user must master many complex and arbitrary operations. To find the proper command for a task at hand, for example, the user must hunt through a complex menuing system, with the desired choice often buried under several menus. Even simple tasks can pose a significant challenge to the user. To change the punctuation format of a number in one prior art spreadsheet, for example, the user must traverse several nodes of a menu tree, carefully selecting among cryptic menu choices along the way. A mistake at any one of the nodes can lead to harsh consequences, including the loss of valuable data.

Finding this approach to be unworkable, many users memorize frequently-needed commands instead. To accomplish the foregoing task, for example, the user would memorize the command: /Worksheet Global Default Other International. As one can only memorize just so many arbitrary commands, however, the user typically masters only a very small subset of available commands and features. And without constant practice, these commands tend to be quickly forgotten. Moreover, many useful and needed commands are sufficiently hidden in layers of menus that they are never discovered by the user. All told, the non-intuitive interfaces of prior art spreadsheets have led to steep learning curves for users. Even after mastering a particular spreadsheet interface, the user typically only knows a fraction of available commands and features, most of which are easily forgotten.

Even with advances in computer and software technology, electronic spreadsheets have not necessarily become easier to use. Instead, technological advances have been largely employed to build more complex functions and modeling features into spreadsheets, often with more complicated menu trees; or worse yet, a staggering array of icons which leave the user even more bewildered. Thus, while prior art spreadsheets have continued to increase in functionality, they have also greatly increased in complexity for the user.

Three dimensionality is one such example. Three-dimensional spreadsheets allow the user to create a worksheet

5,590,259

3

having cells arranged in a 3-D grid. In this manner, the user can manipulate multi-dimensional ranges, i.e., solid blocks of cells. This feature has distinct advantages. For example, the user can build a worksheet consisting of multiple two-dimensional spreads, define 3-D ranges that span these spreads, and copy a range of rows and columns into each of many 2-D spreads at once. This feature eases difficult chores, such as consolidation of multiple spreads.

Despite its advantages, three-dimensionality, as presently implemented, is an advanced feature beyond the grasp of many spreadsheet users. This is not a necessary result of a three-dimensional model per se but, instead, has resulted from poor implementations of the model in prior art spreadsheet programs. One popular implementation of the model in the prior art, for example, requires the user to manipulate each additional spread of a three-dimensional spreadsheet as a separate window in a graphical windowing environment. This approach is far from intuitive, however. In particular, this approach requires the user to master actions which have no counterpart in everyday activities. While three-dimensional spreadsheets provide additional functionality, they serve to illustrate how non-intuitive implementations of new technology greatly add to the complexity of the user interface.

What is needed is application software which maintains a highly intuitive interface. Moreover, the application should implement advanced features, such as three dimensionality, by employing interface objects which are familiar to the user. In this manner, the user will not have to master an elaborate and/or awkward environment but, instead, may rely upon his or her own common fund of knowledge. The present invention fulfills this and other needs.

SUMMARY OF THE INVENTION

Application software, such as electronic spreadsheets, have found wide application in the processing and presentation of information. Despite their computational power, however, these programs require users to master complex interfaces, adopting abstract representation models which are beyond the grasp of ordinary users. As a result, much of the computational power of these applications goes unutilized by end users.

The present invention, in contrast, provides system and methods having a highly intuitive interface for users. More particularly, the present invention provides interface objects which are familiar to the user. In an exemplary embodiment of the present invention, an electronic spreadsheet system includes a notebook interface having a plurality of notebook pages, each of which contains a spread of information cells, or other desired page type (e.g., Graphs page).

Methods are provided for rapidly accessing and processing information on the different pages, including, for example, displaying a plurality of page identifiers for selecting individual pages. Additional methods are provided for editing cells and blocks of cells. In this fashion, the spreadsheet notebook of the present invention provides a convenient means for organizing and presenting information. Moreover, the spreadsheet notebook of the present invention readily accommodates complex data (e.g., consolidation across multiple spreads); yet, at the same time provides the user with a highly intuitive interface, one which employs interface objects which are familiar to the user.

The present invention also includes system and methods for conveniently inspecting and setting the properties of object. A method for accessing an objects property, in

4

accordance with the present invention, includes receiving a request from the user for inspection of an object; accessing properties for the object; and displaying the properties to the user. From the displayed properties, the user may alter the characteristics of the object, as desired.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a block diagram of a computer system in which the present invention may be embodied.

FIG. 1B is a block diagram of a software system of the present invention, which includes operating system, application software, and user interface components.

FIG. 1C is bitmap screen shots illustrating the basic architecture and functionality of a graphical user interface in which the present invention may be embodied.

FIG. 2A is a screen bitmap illustrating a spreadsheet notebook of the present invention.

FIG. 2B is a bitmap of a tool bar component of the spreadsheet of the present invention.

FIG. 2C is a screen bitmap illustrating a notebook window from the spreadsheet notebook of the present invention.

FIGS. 2D-E are bitmaps illustrating page identifiers for rapidly accessing and manipulating individual pages of the notebook of the present invention.

FIGS. 3A-C illustrate navigation (i.e., access of desired information) in the spreadsheet notebook of the present invention.

FIGS. 4A-E are screen bitmaps illustrating the selection of information cells in the spreadsheet notebook of the present invention.

FIG. 4F is a screen bitmap illustrating the operation of grouping one or more desired pages in the spreadsheet notebook of the present invention.

FIG. 4G is a screen bitmap illustrating drag-and-drop editing in the spreadsheet notebook of the present invention.

FIGS. 4H-J are a series of screen bitmaps illustrating the model copy method of the present invention.

FIG. 4K is a set of bitmaps illustrating the operation of moving and copying a notebook page.

FIGS. 4L-M are screen bitmaps illustrating the presentation of information in a notebook of the present invention.

FIG. 5A is a screen bitmap illustrating exemplary objects of the spreadsheet notebook of the present invention.

FIGS. 5B-E are bitmaps illustrating property inspectors of the present invention, for the objects of FIG. 5A.

FIGS. 6A-K are bitmaps illustrating different states (and accompanying panes) for the property inspector of FIG. 5E, each state depending on a particular property of the object being inspected.

FIG. 7A is a screen bitmap illustrating a graph window of the present invention, with different graph objects available for property inspection.

FIGS. 7B-H are bitmaps illustrating exemplary property inspectors of the present invention for the graph objects of FIG. 7A.

FIG. 8A is a block diagram illustrating the structure and operation of a spreadsheet notebook of the present invention.

FIGS. 8A-C illustrate the correspondence between a page table data structure and pages which are displayed on the screen to the user.

FIGS. 8D-F illustrate the referencing of information in the spreadsheet notebook of the present invention.

5,590,259

5

FIG. 9A is a flowchart illustrating a method of the present invention for interpreting information references.

FIG. 9B is a flowchart illustrating a method of the present invention for drag and drop editing.

FIG. 9C is a flowchart illustrating a method of the present invention for model copying.

FIG. 10A is a block diagram illustrating a system class hierarchy of the present invention, which is employed for property inspection.

FIG. 10B is a block diagram illustrating the correspondence between a parent object and a child object, and their respective property lists.

FIG. 10C is a flowchart illustrating a method for inspecting and setting properties of objects, in accordance with the present invention.

FIG. 10D is a flowchart illustrating a set property method of the present invention (substep from the method of FIG. 10C).

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

System Hardware

As shown in FIG. 1A, the present invention may be embodied on a computer system such as the system 100, which comprises a central processor 101, a main memory 102, an input/output controller 103, a keyboard 104, a pointing device 105 (e.g., mouse, track ball, pen device, or the like), a display device 106, and a mass storage 107 (e.g., hard disk). Additional input/output devices, such as a printing device 108, may be included in the system 100 as desired. As illustrated, the various components of the system 100 communicate through a system bus 110 or similar architecture. In a preferred embodiment, the computer system 100 includes an IBM-compatible personal computer, which is available from several vendors (including IBM of Armonk, N.Y.).

Illustrated in FIG. 1B, a computer software system 150 is provided for directing the operation of the computer system 100. Software system 150, which is stored in system memory 102 and on disk memory 107, includes a kernel or operating system 151 and a shell or interface 153. One or more application programs, such as application software 152, may be "loaded" (i.e., transferred from storage 107 into memory 102) for execution by the system 100. The system 100 receives user commands and data through user interface 153; these inputs may then be acted upon by the system 100 in accordance with instructions from operating module 151 and/or application module 152. The interface 153, which is preferably a graphical user interface (GUI), also serves to display results, whereupon the user may supply additional inputs or terminate the session. In a preferred embodiment, operating system 151 is MS-DOS, and interface 153 is Microsoft® Windows; both are available from Microsoft Corporation of Redmond, Wash. Application module 152, on the other hand, includes a spreadsheet notebook of the present invention (as described in further detail hereinbelow).

Interface: User-familiar Objects

A. Introduction

The following description will focus on the presently preferred embodiments of the present invention, which are embodied in spreadsheet applications operative in the

6

Microsoft Windows environment. The present invention, however, is not limited to any particular application or any particular environment. Instead, those skilled in the art will find that the system and methods of the present invention may be advantageously applied to a variety of system and application software, including database management systems, wordprocessors, and the like. Moreover, the present invention may be embodied on a variety of different platforms, including Macintosh, UNIX, NextStep, and the like. Therefore, the description of the exemplary embodiments which follows is for purposes of illustration and not limitation.

Referring now to FIG. 1C, the system 100 includes a windowing interface or workspace 160. Window 160 is a rectangular, graphical user interface (GUI) for display on screen 106; additional windowing elements may be displayed in various sizes and formats (e.g., tiled or cascaded), as desired. At the top of window 160 is a menu bar 170 with a plurality of user-command choices, each of which may invoke additional submenus and software tools for use with application objects. Window 160 includes a client area 180 for displaying and manipulating screen objects, such as graphic object 181 and text object 182. In essence, the client area is a workspace or viewport for the user to interact with data objects which reside within the computer system 100.

Windowing interface 160 includes a screen cursor or pointer 185 for selecting and otherwise invoking screen objects of interest. In response to user movement signals from the pointing device 105, the cursor 185 floats (i.e., freely moves) across the screen 106 to a desired screen location. During or after cursor movement, the user may generate user-event signals (e.g., mouse button "clicks" and "drags") for selecting and manipulating objects, as is known in the art. For example, Window 160 may be closed, resized, or scrolled by "clicking" (selecting) screen components 172, 174/5, and 177/8, respectively.

In a preferred embodiment, screen cursor 185 is controlled with a mouse device. Single-button, double-button, or triple-button mouse devices are available from a variety of vendors, including Apple Computers of Cupertino, Calif., Microsoft Corporation of Redmond, Wash., and Logitech Corporation of Fremont, Calif., respectively. More preferably, screen cursor control device 105 is a two-button mouse device, including both right and left "mouse buttons." Programming techniques and operations for mouse devices are well documented in the programming and hardware literature; see e.g., *Microsoft Mouse Programmer's Reference*, Microsoft Press, 1989, the disclosure of which is hereby incorporated by reference.

B. Spreadsheet Notebooks

In contrast to conventional applications, even those operative in a windowing environment, the present invention includes user-familiar objects, i.e., paradigms of real-world objects which the user already knows how to use. In an exemplary embodiment, system 100 includes a spreadsheet notebook interface, with such user-familiar objects as pages and tabs. In this manner, complexities of the system are hidden under ordinary, everyday object metaphors. By drawing upon skills which the user has already mastered, the present invention provides a highly intuitive interface—one in which advanced features (e.g., three-dimensionality) are easily learned.

1. General interface

Shown in FIG. 2A, a spreadsheet notebook interface of the present invention will now be described. The spreadsheet notebook or workbook of the present invention includes a

5,590,259

7

notebook workspace **200** for receiving, processing, and presenting information, including alphanumeric as well as graphic information. Notebook workspace **200** includes a menu bar **210**, a tool bar **220**, a current cell indicator **230**, an input line **231**, a status line **240**, and a notebook window **250**. The menu bar **210** displays and invokes, in response to user inputs, a main level of user commands. Menu **210** also invokes additional pulldown menus, as is known in windowing applications. Input line **231** accepts user commands and information for the entry and editing of cell contents, which may include data, formulas, macros, and the like. Indicator **230** displays an address for the current cursor (i.e., active cell) position. At the status line **240**, system **100** displays information about the current state of the workbook; for example, a “READY” indicator means that the system is ready for the user to select another task to be performed.

The tool bar **220**, shown in further detail in FIG. 2B, comprises a row or palette of tools which provide a quick way for the user to choose commonly-used menu commands or properties. In an exemplary embodiment, tool bar **220** includes cut, copy, and paste buttons **221**, a power button tool **222**, a graph tool **223**, alignment buttons **224**, a style list **225**, font buttons **226**, insert/delete buttons **227**, a fit button **228**, and action buttons **229**. Buttons **221** cut, copy and paste data and objects to and from Windows clipboard. The same actions are also available as corresponding commands in the Edit menu (available from menu bar **210**). Tool **220** creates “powerbuttons” which allow a user to run spreadsheet macros; in a specific embodiment, powerbuttons appear as floating objects in a layer above spreadsheet cells. In a similar fashion, the graph tool **223** creates floating graphs that appear above spreadsheet cells.

Other tools are available for formatting cells. Alignment buttons **224** place cell entries flush left, centered, or flush right, as desired. The style list **225** specifies the style for the active block and may be selected from a plurality of pre-defined styles (e.g., normal, currency, fixed, percent, and the like). The font buttons **226** effect font changes, including toggling bold and italic fonts, as well as increasing and decreasing font (point) size. The insert and delete buttons **227** permit the user to insert or delete blocks, rows, columns, and pages (described in further detail hereinbelow), as desired. The fit button **228** allows the user to quickly tailor a column’s width to match its widest cell entry. Action buttons **229** provide automated spreadsheet operations, including sorting and summing operations. For example, a Sort button, when invoked, performs a sort on data in a currently active block. A sum button, on the other hand, totals the values in any block of cells by generating an @SUM function, just outside a selected range of blocks.

2. Notebook and pages

Notebook **250**, shown in further detail in FIG. 2C, provides an interface for entering and displaying information of interest. The notebook **250** includes a plurality of spreadsheet pages, such as page **251** (Page A). Notebook **250** also includes a “Graphs page” for readily accessing all the graphs created for the notebook. Each page may include conventional windowing features and operations, such as moving, resizing, and deleting. In a preferred embodiment, the notebook **250** includes **256** spreadsheet pages and one Graphs page, all of which are saved as a single disk file on the mass storage **107**. In accordance with the present invention, workspace **200** may display one or more notebooks, each sized and positioned (e.g., tiled, overlapping, and the like) according to user-specified constraints. When a single notebook is used, notebook **250** will typically occupy a majority of workspace **200**.

8

Each spreadsheet page of a notebook includes a 2-D spread. For example, spreadsheet page **251** includes a grid in row and column format, such as row **252** (row 3) and column **255** (col. F). At each row/column intersection, a box or cell (e.g., cell C4) is provided for entering, processing, and displaying information in a conventional manner. Each cell is addressable, with a selector **253** being provided for indicating a currently active one (i.e., the cell that is currently selected).

As shown in FIGS. 2C–E, individual notebook pages are identified by page identifiers **260**, preferably located along one edge of the notebook **250**. In a preferred embodiment, each page identifier is in the form of a tab member (e.g., members **261a**, **262a**, **263a**) situated along a bottom edge of the notebook. Each tab member may include representative indicia, such as textual or graphic labels, including user-selected titles representing the contents of a corresponding page. In FIG. 2D, the tab members **260** are set to their respective default names. For example, the first three tab members (members **261a**, **262a**, **263a**) are respectively set to A, B, and C. In a preferred embodiment, however, tab members are typically given descriptive names provided by the user. As shown in FIG. 2E, for example, the first three tab members have now been set to “Contents” (tab member **261b**), “Summary” (tab member **262b**), and “Jan” (tab member **263b**). In a similar manner, the remaining tabs are set to subsequent months of the year. In this manner, the user associates the page identifiers with familiar tabs from an ordinary paper notebook. Thus, the user already knows how to select a page or spread of interest: simply select the tab corresponding to the page (as one would do when selecting a page from a paper notebook).

In addition to aiding in the selection of an appropriate page of information, the user-customizable page identifiers aid in the entry of spreadsheet formulas. For example, when entering a formula referring to cells on another page, the user may simply use the descriptive page name in the formula itself (as described hereinbelow), thus making it easier for the user to understand the relationship of the cell(s) or information being referenced.

3. Navigation in a notebook

Referring now to FIGS. 3A–C, movement (i.e., location of desired information cells) within a spreadsheet notebook of the present invention is illustrated. To move to different pages in the notebook, the user simply selects the corresponding tab from tabs **260**. To move to Page B, for example, the user selects (e.g., with keyboard **104** or pointing device **105**) tab **262a**; similarly, Page C is reached by selecting tab **263a**. Continuing the example, the user may return to Page A by selecting tab **261a**. Thus instead of finding information by scrolling different parts of a large spreadsheet, or by invoking multiple windows of a conventional three-dimensional spreadsheet, the present invention allows the user to simply and conveniently “flip through” several pages of the notebook to rapidly locate information of interest.

Notebook **250** also includes supplemental tools for navigation between pages, including a tab scroller **271** and a fast-forward button **272**. The tab scroller (shown in two different states: **271a** and **271b**) permits access to identifiers for pages which are not currently visible on the screen device **106**. If a desired identifier or tab is not currently in view, the user simply activates the tab scroller **271** to reveal additional tabs. The fast-forward button **272**, on the other hand, provides immediate access to the last pages of the notebook, including the Graphs page. As shown in FIG. 3A

5,590,259

9

and B, after invoking the fast-forward button **272a**, the page identifiers for the last pages (e.g., tabs **261c**, **262c**, **263c**, **265**) are accessible. To switch back to the previously active spreadsheet page, the user may select or click the fast-forward button **272c** again. For navigation within a spreadsheet page, horizontal and vertical scrollbars **278**, **259** are provided; general features and operations of basic scroller or sliders are described in Windows user and programming literature, including Microsoft's *Windows Software Development Kit*.

4. Selections and aggregate operations within a notebook

The selection of desired information cells in the notebook of the present invention is now described. For selecting a plurality of information cells, both 2-D blocks (e.g., block **254** of FIG. 2C) and 3-D blocks of cells may be defined. A 2-D block is a rectangular group of one or more cells and is identified by block coordinates, such as the cell addresses of its upper-left and bottom-right corners. Similarly, a 3-D block represents a solid block (e.g., cube) of cells.

A 2-D block is specified by selecting, with mouse **105** or keyboard **104**, opposing corners. In FIG. 2C, for example, the block **254** is defined by corner cells **C5** and **F14**. Additional selection examples are illustrated in FIGS. 4A–E. For example, column B (col. **411**) is selected by clicking column heading **410**; similarly, row **3** (row **421**) is chosen by clicking row heading **420**. Selection may be additive (i.e., additional selections are appended to the current ones), as shown by selection of a row **420** and a column **410** in FIG. 4C. To facilitate the selection of all cell members (e.g., block **431**), a select-all button **430** is also provided. In addition to these “contiguous” blocks, non-contiguous block selection (e.g., selection of blocks **441**, **442**) is provided by use of a status key (e.g., CTRL-, ALT-, or SHIFT-) plus a mouse event (e.g., click and drag operations).

Selection of 3-D cell blocks, i.e., cell ranges spanning more than one page, occurs in a similar fashion. To extend the block **254** (of FIG. 2C) into a 3-D block, the user specifies an additional or third dimension by selecting an appropriate page identifier. If the user selects the D tab while the block **254** is selected (e.g., with a SHIFT or other status key), the 2-D block is extended into a 3-D block spanning from Pages A to D. Additional 3-D operations may be performed by utilizing a method of the present invention for grouping pages, which will now be described.

Pages may be selected or grouped together, thereby providing a means for changing multiple pages simultaneously. In much the same manner as cells from a spread are grouped into 2-D blocks, a range of pages are grouped by specifying beginning and ending members. As shown in FIG. 4F, a range from Page A to Page K may be achieved by selecting tabs A (**261**) and K (**267**) from identifiers **260**, for example, while depressing a key (e.g., status key). A grouping indicator **268** is displayed for indicating members of a group; groupings may also be annotated with user-specified labels. Once grouped, a page of the group may have its operations (e.g., selection, data entry, and the like) percolate to the other members of the group, as desired. A non-contiguous selection of pages may also be selected (even across different pages); for example, a selection of Pages A and D, but not B and C, may be achieved by selecting tabs A and D while depressing a second key (e.g., CTRL- key). Furthermore, groups may overlap (i.e., a page can be in more than one group), as desired. By selectively activating a group mode (e.g., by toggling group button **273**), groupings may be temporarily turned off, in which case events are not percolated to other members of the group.

10

With group mode active, an activity in a page which is a member of a group can also affect similarly situated cells of the other pages of the group. For example, information entered in a cell on one page of the group can also propagate to the same (relative) cell in other pages of the group; data entry may be “drilled” (propagated) to other group pages by concluding data entry, for example, with a “Ctrl-Enter” keystroke (instead of a “Enter” command). For instance, an entry into cell **C4** of Page A may also conveniently and automatically appear in cell **C4** of pages B, C, D, E, F, G, H, I, J, and K, for an A–K grouping. Similarly, block or aggregate operations may propagate across member pages. For example, a block operation (e.g., Block Fill) for cells **A1** to **C4** in Page A in an A–D grouping would also operate (in this case, fill information) in the same cells (i.e., from **A1** to **C4**) for Page B to Page D.

Employing the user-specified page identifiers of the present invention, a simple nomenclature is available for specifying these solid blocks of information. In a preferred embodiment, a solid block is specified by:

```
<<First Page>> . . <<Last Page>> : <<First Cell>> . . <<Last Cell>>
```

For example, a solid block may be defined as A . . . D:A1 . . . C4, in which case the block spans from cells **A1** to **C4**, and spans across Pages A–D. By permitting alias names (i.e., user-supplied alternative labels), the present invention allows the block to be specified as 1989 Sales . . . 1992 Sales: A1 . . . C4; or even 1989 Sales . . . 1992 Sales: First Quarter . . . Third Quarter. Additionally, the present invention readily accommodates notebook information as well, for example, [TAX]1989 Sales . . . 1992 Sales: First Quarter . . . Third Quarter, thus permitting information to be linked across various notebooks. Wildcard Characters (e.g., * and ?) may also be employed for cell, page, and notebook identifiers, as desired. Thus, the spreadsheet notebook of the present invention provides a 3-D interface which readily accommodates real-world information in a format the user understands (instead of forcing the user to adapt his or her information to fit an arbitrary spreadsheet model).

5. Advanced Editing

Whether 2-D or 3-D in nature, blocks of cells may be easily copied and cut. (i.e., moved) using drag-and-drop editing techniques of the present invention. As shown in FIG. 4G for a 2-D block, for example, a method for copying a block of cells includes (1) selecting a source block by dragging a range of cells (e.g., mouse button-down events coupled with mouse movement across the range; close selection with a button-up event), (2) dragging the block (e.g., click within block, followed by repeated mouse button-down events), and (3) dropping the block (e.g., mouse button-up event at desired target location). In a similar fashion, 3-D blocks may be dragged and dropped.

In typical cut and copy operations, relative and absolute cell addressing is employed, as is known in the art (see e.g., *Using* 1-2-3). According to the present invention, however, a “model copy” technique for copying blocks is also provided. Model copying, illustrated in FIGS. 4H–J, is useful when the user is copying a block that contains absolute references to cells within the copied block. In FIG. 4H, a small spread model **496** is shown which contains a formula to figure the monthly payment for a 30-year loan at different interest rates; a reference to the loan amount was entered as absolute so that when the formula is copied, it continues to refer to cell **B1**. The user may want to calculate (at the same time) monthly payments for different loan amounts and, thus, might copy the model, with the loan amount changed (shown in FIG. 4I as spread **497**). However, in this approach

the absolute reference still refers to row 1; to correct this, the user would typically manually edit each formula to refer to B6 instead of B1.

With model copying of the present invention enabled (e.g., by default or through a user dialog), however, the problem is solved. In particular, absolute references adjust to the new location of the referenced cell, as shown by spread 498 in FIG. 4J; however, absolute references remain absolute to make future copies absolute. For instance, should the user make more copies of the formula, the reference to cell B6 is still absolute. In this manner, model copying of the present invention saves the user time-consuming and error-prone editing of formulas.

As shown in FIG. 4K, notebook pages may be copied or moved using the drag-and-drop editing techniques of the present invention (a dialog box is also provided as an alternative). As shown, for example, the "Salads" page is moved by selecting its tab identifier 264a from the identifiers 260; in a preferred method, the identifier is selected by dragging it downward (mouse button-down plus downward movement) with the mouse cursor. Next, the identifier is dragged to a desired new position (mouse button-down plus left and right movement). Once positioned at a desired location (as indicated by in-transit tab identifier 264b), the page is then "dropped" (mouse button-up) into position. Similarly, a "copy" operation may be effected by coupling the above operation with a status key (e.g., CTRL-); in a preferred method of copying, only the page information (not its tab identifier) is copied to a target location.

Additional editing operations may be easily accomplish using the page identifiers of the present invention. For example, an additional page may be inserted by selecting a desired tab and entering "INS" (keyboard or mouse). Similarly, a notebook page may be deleted by coupling the selection of a corresponding page tab to a delete command.

6. Advantages

In contrast to prior art spreadsheet implementations, use of the spreadsheet notebook of the present invention is easily ascertained by the user. The notebook interface of the present invention provides a convenient means for organizing many spreadsheets together into one file. This permits the user to load (into memory 102) all related information with a single command, without having to remember a variety of different file names. Moreover, the notebook interface 250 encourages a user to spread data out among multiple pages, thus better organizing one's data. In FIG. 4L, for example, spreadsheet page 300 illustrates the conventional method for grouping unrelated information onto a single spreadsheet. As shown in column B, the incorporation of unrelated information into a single spread leads to unnecessary whitespace padding of information. In FIG. 4M, in contrast, related information is grouped on separate pages 351, 352 (e.g., with their own columns, Col. B₁ and Col. B₂) of notebook 350. Not only does this eliminate grouping of disparate information, with its disadvantages (e.g., padding data entries), but it also encourages better organization of information.

Inspecting and Setting the Properties of Objects

A. Disadvantages of Prior Techniques

Standard windowing interfaces depend heavily on a clunky system of pull-down menus. While pull-down menus are an improvement over command-line (e.g., MS-DOS) interfaces, they are non-metaphoric or non-analogous to ordinary objects, i.e., ones familiar to the user. Thus, prior

art windowing GUIs are no more intuitive or useful to the user than other menuing systems.

Perhaps the biggest weakness of pull-down menus is the requirement that the user must know beforehand which menu contains the item or function of interest. If the user does not know which pull-down menu, submenu, or dialog box contains the item he or she is seeking, the user will spend an inordinate amount of time checking the contents of each in an effort to hunt down the needed item. And often the search is in vain. Moreover, since functions for a given object (e.g., text object) are often scattered over several disparate menus, the user is discouraged from interacting and experimenting with the object.

One approach to overcoming this problem has been the implementation of "smart icons." This interface technique employs screen buttons painted with icons which are supposed to tell the user what the buttons do. This approach, however, often makes the interface problem even worse because the meaning of the icons are not easily grasped. Thus, the approach is often no more helpful than hiding the desired function deep inside a menuing system. Worse, some button images are so small or so numerous that it is hard to see the icons well enough to discern what they mean.

B. Property Inspectors

1. Introduction

Overcoming this problem, the present invention provides "property inspectors" for inspecting and setting the properties of objects. Property inspectors of the present invention examine an object of interest to the user and tell the user what can be done to it. In this manner, the present invention requires the system and its interface components (e.g., menus or dialog boxes), not the user, to know what functions are being sought or may be of immediate interest to the user. Moreover, property inspectors of the present invention require the program to present functions to the user when and where he or she requests them, rather than making the user hunt for them when they are needed. Thus, the user need only know which data item or object he or she wants to inspect or manipulate.

In the spreadsheet notebook, for example, there are many kinds of objects which one can use. Examples include cells, blocks of cells, pages, notebooks, graphs and their elements (including bars or axes), dialog boxes, and even the application program itself. Each of these objects has properties, which are characteristics of that particular object. For example, blocks of cells have a font property that can be set to bold, so that the text of entries in the block appear in boldface type. A property of a workbook page, on the other hand, is the name that appears on its tab. Each notebook has its own palette property for controlling available colors. Application properties, for instance, include system defaults, such as a default storage directory or file extension. Thus in any system, the user has a variety of objects available, each of which has its own set of properties.

Property inspection of the present invention provides the user with immediate access to an object's properties. If the object of interest is a spreadsheet cell, for example, the property inspector of the present invention produces a menu that includes font, color, border, size, and other display properties, along with formula properties which may be modified. If, on the other hand, the object is a graph, the property inspector will display a menu to change its color, shape, borders, and other features of a spreadsheet graph. Moreover, inspection menus are state or context-sensitive, i.e., constructed on-the-fly to reflect the current state of the object. If an object's properties change so that what a user

5,590,259

13

can do to it also changes, the property inspector of the present invention will create a new and appropriate menu reflecting those changes.

A variety of user events, including keyboard and mouse events, may be employed to invoke property inspection and setting. In a preferred method of the present invention, however, an object is inspected by selecting the object with a screen cursor, i.e., clicking (generating a mouse event) on the object. Since, according to the present invention, property inspection may be available across different modes of operation (i.e., generally available at all times), the generated mouse event or signal is preferably one which does not launch or otherwise invoke routine object actions. Instead, the mouse signal should be one which the user will easily associate with inspection of an object. In a two or three mouse button embodiment of the present invention, therefore, the generated mouse signal is most preferably from the lesser-used or right mouse button (e.g., Windows' WM_RBUTTONDOWN). In this manner, the user associates object actions with left button signals and object inspection with right button signals.

2. Exemplary Embodiments

Referring now to FIGS. 5A–E, the inspecting and setting of object properties, in accordance with the present invention, is illustrated. In FIG. 5A, a notebook application window **500** is shown. Window **500** includes a variety of objects which may be inspected by the property inspector of the present invention. The top-most object is the application itself: application object **510**. By selecting the application object (e.g., by right mouse clicking on the application title bar), various properties for the application may be inspected and set. As shown in FIG. 5B, inspection of the application object **510** invokes application inspector **515**, which lists the various properties of the application object; the user may now inspect and set the properties of the object, as desired.

Referring back to FIG. 5A, the next level which may be inspected is the notebook object **520**. User inspection of this object, e.g., by right mouse clicking on the notebook title bar, invokes the active notebook property inspector **525** of FIG. 5C. In a manner similar to the application property inspector, the notebook property inspector **525** includes all the properties and corresponding options which may be set for the notebook **520**.

Notebook object **520** includes a plurality of page objects, which may themselves be inspected. To inspect the page object **550**, for example, the user brings the page tab into view (if it is not already displayed), then right clicks the tab page. This action invokes the active page property inspector **535**, as shown in FIG. 5D. With access to page properties, the user can control the page name, overall page protection, line color, colors in cells that meet certain conditions, default label alignment, whether zeroes are displayed, default column width, row and column borders, gridline display, and the like. To assign a more descriptive name to a page, for example, its tab is inspected and the option "NAME" is selected from the page property inspector (or, as shown, it is selected by default). After the user enters a new name, the new name appears on the page tab. Other page properties, most of which affect the appearance of data or screen elements, may be set in a similar manner.

Each page object may also contain other objects, including cell, blocks of cells, graph objects, and the like, each of which may be inspected. By right clicking the active block **530**, the active block property inspector **545** is opened as shown in FIG. 5E. With block properties, the user can change alignment of entries, numeric format of values, block

14

protection, line drawing, shading, font, text color, types of data allowed in a cell, row height, column width, and whether columns or rows are revealed or hidden; inspection of a single cell occurs in a similar manner (i.e., active block is one cell). In this manner, the user may select block properties which affect the appearance of cell entries or of rows or columns; additionally, properties which affect the way data can be entered into blocks, such as protection and data entry input, may be made.

Upon invocation of property inspection, a dialog box or property inspector is displayed, preferably at or near the object of interest. A different property inspector appears depending on the type of object which is inspected. At this point, the user may then select the property that he or she wishes to change from a list of object properties. The contents of the right side of the inspector (i.e., a "pane" within the inspector) change to correspond to the property one chooses, as shown. Then, the user chooses settings for the current property. If desired, the user can then choose and set other properties for the current object. In a preferred embodiment, the property inspector includes an example box, which shows the display result of the property the user has just set, before they are applied to the actual object.

Referring now to FIGS. 6A–K, the construction and operation of a property inspector dialog box, in accordance with the present invention, is described. For purposes of illustration, the following description will present active block property inspector **600**; other inspector dialogs may be implemented in a similar fashion. Shown in FIG. 6A, the active block property inspector **600** includes an object property list **601** and property settings pane **602**. For an active block, for example, relevant properties include Numeric Format, Font, Shading, Alignment, Line Drawing, protection, Text Color, Data Entry Input, Row Height, Column Width, and Reveal/Hide. Property settings pane **602** include the valid options or settings which a property may take. For the property of numeric format, for example, valid settings include fixed, scientific, currency, and the like. Property setting pane **602** may further include subproperties. For example, the currency setting **602a** shown may also include a decimal place setting **602b**. In this manner, property inspector dialog **600** presents a notebook dialog: property list **601** serves as notebook tabs (e.g., dialog tab **601a**) and property settings panes **602** serves as different pages of the notebook dialog. Thus, the user may access numerous dialog options with the efficiency of a notebook.

Also shown, property inspector dialog **600** also includes an example window **605** and dialog controls **603**. As new properties are selected and set, their net effect is shown. The example element in example window **605**, for example, shows the effect of selecting a numeric format of currency with two decimal places. Thus, the user may experiment with changes before they are applied to the data object. After desired property changes are entered, control components **603** are invoked to either accept ("OK button") or reject ("CANCEL button") the new property settings; if desired, help may be requested.

As shown in FIGS. 6B–K, other properties of a cell block are accessed by selecting the desired property from the property list **601**. Thus the active blocks property inspector **600** changes to inspector **620** for font properties, inspector **625** for shading properties, inspector **630** for alignment properties, inspector **635** for line drawing properties, inspector **640** for cell protection properties, inspector **645** for text-color properties, inspector **650** for data entry input properties, inspector **655** for row height properties, inspector **660** for column width properties, and inspector **665** for

5,590,259

15

reveal/hide properties. In each instance, a new pane (i.e., one having the attributes of interest) is displayed in the inspector window.

Referring now to FIGS. 7A–H, the inspection (and setting) of properties for graphs is illustrated. Graph window **700** includes a plurality of graph objects, each of which may be customized through use of a corresponding property inspector of the present invention. To display a corresponding property inspector, the user invokes the inspector (e.g., right-clicks) for the part (object) of the graph he or she wishes to change. A right-click on the graph window object **710**, for example, will invoke the graph window inspector **715**; at this point, the user may inspect and set various properties of the graph window object. In a similar manner, other objects of the graph window may be inspected. For example, inspection of graph background **720** invokes inspector **725**, Y-axis object **730** invokes inspector **735**, X-axis **740** invokes inspector **745**, area fill object **750** invokes inspector **755**, bar series object **760** invokes inspector **765**, and bar series object **770** invokes inspector **775**.

Internal Operations

A. Introduction

Internal operations of the system **100** will now be described in detail. In general, operation is driven by methods of the present invention, which are invoked by Windows' message dispatcher in response to system or user events. The general mechanism for dispatching messages in an event-based system, such as Windows, is known in the art; see, e.g., Petzold, C., *Programming Windows*, Second Edition, Microsoft Press, 1990. Additional information can be found in Microsoft's Window Software Development Kit, including: 1) *Guide to Programming*, 2) *Reference, Vols. 1 and 2*, and 3) *Tools*, all available from Microsoft Corp. of Redmond, Wash. Of particular interest to the present invention are class hierarchies and methods of the present invention, which are implemented in the C++ programming language; see e.g., Ellis, M. and Stroustrup, B., *The Annotated C++ Reference Manual*, Addison-Wesley, 1990. Additional information about object-oriented programming and C++ in particular can be found in Borland's C++ 3.0: 1) *User's Guide*, 2) *Programmer's Guide*, and 3) *Library Reference*, all available from Borland International of Scotts Valley, Calif. The disclosures of each of the foregoing are hereby incorporated by reference.

B. Notebooks

Referring now to FIG. 8A, the internal structure and operation of spreadsheet notebooks of the present invention will be described. The spreadsheet notebook system of the present invention includes a system hierarchy **800**, at the top level, having a notebook table **805** with entries or slots for accessing a plurality of notebooks. For example, the first slot addresses or points to notebook **810**. In this manner, the system may track multiple notebooks.

Notebook structure **810**, in turn, includes or accesses various data members for a particular notebook. For example, a "Name" field, stores the name assigned for the notebook. This name is displayed in the titlebar for the notebook and is used as the name for the corresponding disk file; the notebook name is also used to reference the notebook in formulas (e.g., contained in other notebooks). Notebook **810** also includes other data members, such as block names and fonts. Block names are text strings or labels which the user has defined for particular blocks of interest. Fonts, on the other hand, include font information (e.g., type

16

and size) for the notebook. Other desired properties, specific to a notebook, may be included as desired. As exemplary construction of a notebook structure (class), in accordance with the present invention, is set forth in the microfiche Appendix A in U.S. Pat. No. 5,416,895, the parent patent.

Of particular interest in the notebook structure **810** is the Pagetable field, which includes a pointer to a page table for the notebook. Pagetable **815** includes a plurality of slots or entries for accessing individual page structures of the notebook. Each page structure, in turn, contains or accesses information of interest to an individual page. As shown, for example, page **820** (pointed to by the first slot of the pagetable) includes or accesses: a sheet table (or Graphs page), a pagename, floating objects (e.g., graphs and buttons), page properties, pane, and the like.

The Sheetttable field of the page **820** points to a sheet table **825**. It, in turn, includes a plurality of the slots for accessing different column strips. As shown, for example, the first entry in the sheet table **825** accesses a column strip of cells **830**. In turn, column strip **830** addresses individual information cells, each of which may have one of a variety of information types, including empty, integer, number (real), formula, label, and the like. In a preferred embodiment, column strip **830** may address up to 8,000 cells; those skilled in the art, however, will appreciate that column strip **830** may be set to any desired range (depending on the limits of one's target hardware).

Referring now to FIGS. 8B–C, the function of the page table of the present invention is illustrated. In FIG. 8B, two images are presented: a view image and a data image. The view image illustrates what the user sees on the screen **106**; at startup, for example, only a single Page A (page **850**) is active. As shown by the corresponding data image supporting this view image, pagetable **855** includes only a single reference, page A. In turn, page A references sheetttable_A, which supports the information cells required (e.g., they are currently needed for display) for the page. Thus, at startup, only a single page entry exists in the pagetable **855**, even though the screen displays multiple page tabs.

Upon selection of Page F (e.g., by clicking tab F **860**), the data image changes, however. As shown in FIG. 8C, remaining Pages B–F are initialized into the page table **865**. In this example, Page F must now exist as a view object. Thus, Page F references supporting data structures (e.g., sheetttable_F). The intervening Pages (B–E), on the other hand, are initialized, but only to the extent that they can serve as place holders in the pagetable **865**. In this manner, the underlying page table for a notebook instantiates underlying page data structures only as needed (e.g., for viewing or referencing by other pages), but, at the same time, provides on-demand access to pages.

A particular advantage of this design is the ease in which information may be referenced and presented. Even complex data models, such as those spanning multiple dimensions, may be represented in a clear fashion. Moreover, a syntax is provided by the present invention for intuitively referencing information.

Referring now to FIGS. 8D–F, the referencing of information in a spreadsheet notebook of the present invention is now described. Shown in FIG. 8D, a three-dimensional reference (i.e., identifier for a solid block of information cells) includes a notebook, starting page, ending page, starting cell and ending cell. As shown, the notebook (which is the same as the file name) is identified preferably by delimiters, such as brackets ([]). This is followed by page name(s), which may in fact be grouped. As shown, a range

5,590,259

17

of pages has been defined from '89 Income to '92 Income; these are alias names which may correspond to pages A–D, for example.

Next, one or more cells of interest are identified. For example, a range from January to (. . .) December is shown, which serve as aliases for blocks A1 to A12, for example. The block or cell information is separated from group or page information, for example, by a colon (:) identifier.

The hierarchy of this nomenclature is shown in FIG. BE. Specifically, a notebook references pages, which may be members of one or more user-defined groups. In turn, a page references cells, which may alternatively be referenced by alias identifiers. As shown in FIG. 8F, page and cell identifiers may be grouped to form even more succinct references. For example, Pages '89 Income to Income may be renamed as Four-Year Income. Similarly, the cell range from January to December may be renamed Entire Year. In this manner, information ranges in the spreadsheet notebook of the present invention are easily named and easily visualized by the user.

Depending on the context of the system, certain identifiers may be assumed and, thus, eliminated from a reference. Unless otherwise specified, for example, the notebook identifier is assumed to be the currently active notebook. Similarly, the page identifier may be assumed to be the currently active page, unless the user specifies otherwise. Thus, a valid reference need only include as much information (identifiers) as is necessary to access the desired information.

Referring now to FIG. 9A, a method for interpreting or parsing an information reference, in accordance with the present invention, is shown. Upon receiving a reference (e.g., text string) to information in a spreadsheet notebook, the method proceeds as follows. In step 901, a loop is established to tokenize (i.e., parse) the string into individual identifiers, which may then be processed. Each token is examined as follows. In step 902, the token is checked to determine whether it is a notebook identifier. In a preferred method, a notebook identifier is delimited, for example, by bracket symbols; alternatively, a notebook identifier may be realized simply by the position of the identifier relative to other tokens in the string, or by its relationship to disk (notebook) files on mass storage 107. If the notebook is successfully identified, the notebook identifier is returned at step 903, and the method loops back to step 901 for any additional tokens.

If it is not a notebook (no at step 902), however, then in step 904 the token is examined to determine whether it is a group name. Group names may be determined by accessing group names listed for the notebook (e.g., by accessing a group name string pointer from notebook structure 810), and/or by the token's context (e.g., preceding a ":" delimiter). If a group name is identified, it is returned at step 905, and the method loops for any remaining tokens. Otherwise (no at step 904), the token is examined to determine whether it is a page. In a manner similar to that for checking group names, a page may be identified by examining a notebook's page table, with corresponding page name accessed (dereferenced). The page is returned at step 907, with the method looping back to step 901 for remaining tokens.

If a page is not found (no at step 906), however, then at step 908 the token is examined to determine whether it defines a range. A range may include a named block of cells (e.g., "entire year") or, simply, two cell addresses separated by an appropriate identifier (e.g., A1 . . . B1). If a range is found, then it is returned in step 909, with the method looping for any remaining tokens. Otherwise (no at step

18

908), the identifier is examined to determine whether it is a cell. A token may be identified as a cell if it is of the appropriate column/row format (e.g., A1). If a cell is found, it is returned at step 911, with the method looping for any remaining tokens.

As shown (conceptually) at step 912, if any ambiguities remain, they may be resolved according to an order of precedence; for example, notebook >groupname> page and the like. At the conclusion of the method, a reference, if it is in proper form, will be successfully identified and may now be processed according to three-dimensional techniques.

Referring now to FIGS. 9B–C, other methods of the present invention are illustrated. In FIG. 9B, for example, a drag-and-drop edit method 920 of the present invention is shown. Its steps are as follows. In step 901, a block of cells is defined (e.g., by dragging a mouse cursor from one corner of the block to another). After a block has been selected, it is next grabbed with the mouse cursor (i.e., position the mouse cursor within the block and depress a mouse button). In steps 923 and 924, a loop is established to "drag" the block to a new location. In particular, while the mouse button is depressed, any movement of the mouse will cause the block to follow (animated on screen, e.g., by XOR technique). At step 925, the block is "dropped" into place by releasing the mouse button (button up signal). Dropping the block causes information from the source location to be copied into the target location. By coupling the method with a status key (e.g., SHIFT- or CTRL-), the technique supports either "cut" (erase original) or "copy" (leave original) modes of operation. Thus, for example, if a status key is not active as step 926, then in step 927 the original (at the source location) is deleted. Otherwise (yes at step 926), the original remains at the source location as well.

Referring now to FIG. 9C, a model copy method 940 of the present invention is illustrated. In step 941, a block is defined or selected (e.g., dragging a selection). In step 942, model copy is enabled or disabled (as desired); alternatively, model copy may be enabled by default. In step 943 if model copy has been enabled, then in step 945 absolute address references are copied as if they were relative address references, as previously described (with reference to FIGS. 4H–J). However, the address labels will remain absolute, so that they will be treated as absolute for future copying operations. Otherwise (no at step 943), absolute addresses are treated conventionally (i.e., referencing absolute addresses) in step 944. As shown in step 946, relative addresses are not affected, i.e., they continue to be treated relatively. In step 947, the copy operation is performed, employing the addresses as just determined, after which the method concludes.

Additional methods of the present invention are set forth in microfiche Appendix A in parent application No. 07/866,658, now U.S. Pat. No. 5,416,895.

C. Property Inspection

Referring now to FIGS. 10A–C, construction and operation of property inspection in accordance with the principles of the present invention will be described: As shown in FIG. 10A, user interface (UI) objects are constructed from a UI object class hierarchy 1000. As shown, class hierarchy 1000 includes as its base class an object class 1001. From object class 1001, a responder class 1002 is derived. As the child of class 1001, responder class 1002 inherits all properties of its parent and adds event handling capability. Both object class 1001 and responder class 1002 are abstract classes, i.e., objects are not instantiated from them directly. From

5,590,259

19

responder class **1002**, two child classes are derived: view class **1003** and window class **1004**. From view class **1003**, all screen objects (e.g., text, graphs, scrollbars, and the like) are created. From window class **1004**, container objects are instantiated; in particular, window class **1004** provides container objects which hold the various view objects.

Associated with each class (and thus objects derived from each class) is a ClassInfo structure. ClassInfo, which is itself implemented as a separate class, contains information about a corresponding class and, thus, objects of that class. For example, it contains information about object type, name, number of properties, and the like. Of particular relevance to property inspection are two data members: a pointer to the parent and a property record pointer which points to an array of property records for an object of interest.

Referring now to FIG. **10B**, the relationship between parent and child (and hence the importance of the pointer to the parent) is illustrated. In the system of the present invention, an object-oriented mechanism is employed whereby new objects may be created from existing objects (classes). As shown, for example, a child object **1030** may be derived from a parent object **1020**. Also shown, each object has its own property record array or list. For example, the parent object **1020** has a property list **1025** describing its properties. Child object **1030**, in turn, also has its own property list **1035**; child object **1030** still inherits from parent object **1020**, however. By means of the parent pointer, the child object **1030** also has direct access to its parent **1020** and, thus, the property list **1025** of the parent. In this manner, when an object is inspected, the system of the present invention may view back across the inheritance hierarchy to fetch all (ancestor) properties for the object, as needed.

The property record, on the other hand, describes an individual property for the object. The property record, which is implemented as a separate class, includes a name ID for the property and a pointer to the property object itself (which may be shared). For example, property record objects may be instantiated from the following exemplary C++ class definition:

```
class _EXPORT_ PropRecord
{
public:
    Property * pProp; // pointer to SHARED (!) property object
    WORD nameID; // property name string ID
                    // (PROPSTR.HPP, PROPSTR.CPP)
    WORD flags; // optional information about the property
               // ( HIDDEN, USE MENU, LINK ONLY, etc.)

    inline Property * getProp ();
    inline LPSTR getName ();
    inline PROPTYPE getType ();
    inline WORD getNamesCnt ();
    inline WORD getNameID ();
    inline WORD getFlags ();
};
```

The property object, in turn, includes a type ID for identifying the property and also includes a pointer to a dialog for the property. An property object may be instantiated from the following exemplary C++ class:

```
class _EXPORT_ Property
{
public:
    static DWORD dwPropErr;
    static char far conversionBuffer [MAXPROPSTR];
```

20

-continued

```
PROPTYPE typeID;
WORD namesCnt;
LPSTR pDialog;
Property ( WORD id );
virtual BOOL stringToValue ( LPSTR, PROPTYPE
pt = 0 );
virtual BOOL valueToString ( LPSTR, PROPTYPE
pt = 0 );
/*
Convert the passed binary value to string using the conversionBuffer
*/
LPSTR convertToString ( LPSTR pS, PROPTYPE pt );
/*
Convert the string in conversionBuffer to binary value pointed by pS
*/
BOOL convertToBinary ( LPSTR pS, PROPTYPE pt );
};
```

Exemplary property types may include:

```
prop_ Undefined = 0,
prop_ Text,
prop_ Bool,
prop_ Window,
prop_ Color,
prop_ Bitmap,
prop_ List,
prop_ Word,
prop_ Int,
prop_ Key,
prop_ Double,
...
```

The pointer to the property dialog (LPSTR pDialog), on the other hand, indicates the appropriate dialog to be displayed to the user for the current property; the dialog displays the appropriate information (e.g., pane) for the property which the user may then use for inspecting and setting attributes of that property.

Referring now to FIG. **10C**, a method **1050** for property inspection, in accordance with the present invention, is illustrated; additional disclosure is provided for the inspection of an exemplary object: a spreadsheet cell. In step **1051**,

the user indicates that he or she desires property inspection for an object. In a preferred method of the invention, the user will right-mouse click on the object of interest (as set forth hereinabove). To inspect a spreadsheet cell, for example, the user would position the mouse cursor over the cell and click the right mouse button. In this instance, the notebook which contains that cell will receive (i.e., will be the recipient of) a right-mouse button down message. The event will be processed by routing the message to a RightMouseDown method.

5,590,259

21

In step 1052, the object which the user clicked on is identified by the RightMouseDown method. Specifically, the method invokes a RouteMouseEvent method which returns the current view object (i.e., the object which the user has clicked on). This is accomplished as follows. Every view object contains a StartInspect method which returns an actual object for inspection. The object which appears on the screen to the user (i.e., view object) is not necessarily the object which will be inspected. For example, it would be computationally too expensive to have each cell of a spreadsheet be its own object. Instead, the present invention embraces the notion of surrogate or imaginary objects which may be inspected (using the properties of the screen or view object selected). In most instances, the StartInspect method will return an object for inspection which is the same as the view object. In those instances where it is not feasible to inspect an object which the user perceives on a screen, however, the method returns a surrogate object whose properties assume the attributes of the screen object.

An additional example illustrates this point. When inspecting a block of cells, for example, StartInspect returns a current block object which is a surrogate object, i.e., it does not have a visible expression on the screen. Instead, it is a substitute object which assumes the characteristics necessary to allow inspection of the screen object of interest. In this manner, it functions as a surrogate for the object which the user observes on the screen. If desired, objects may also be blocked from inspection at this step (e.g., by setting flags); in which case, the method terminates. At the conclusion of step 1052, the object of interest, either actual or surrogate, is ready for inspection.

Next, the user interface for inspection is built by a DoUI method of the present invention. The method proceeds as follows. The first property record for the object is accessed in step 1053. Preferably, DoUI is a virtually defined method, with each object (class) designed to include method steps for accessing the property record according to that object's own particularities. The remaining property records for the object are also located (e.g., by traversing a list of records).

At step 1054, the dialog panes for each property are loaded (e.g., into memory 102, if not already present) for use by a compound dialog box. As previously described, the dialog associated with each property is accessible from the property record (by a pointer to the dialog). At this point, an empty property inspection window is displayed. Into the property inspection window, a corresponding pane for each property is loaded (substep 1054b), thus creating a display hierarchy. Using a getProperty method, each property is initially set to the attribute(s) contained in the retrieved property record; the getProperty method operates essentially the reverse of a setProperty method, which is described in detail hereinbelow. An associated screen button or label is provided for rapidly accessing a desired pane in the inspector window, in much the same fashion as one accesses pages in a notebook with tabs. In this manner, a property inspector is built from a dynamic list of properties (as opposed to a static list of properties which is determined beforehand), each of which may be rapidly accessed by the user.

While the property list loaded into an inspector window for each object is dynamically constructed, the panes supporting each property may be pre-built. For example, the pane supporting a color palette, since it is complex in design, will be built in advance.

However, the method of the present invention may dynamically build individual panes as well. Candidates for dynamic pane building include simple properties, especially those which may have only a Boolean or yes/no value.

22

Referring back to FIG. 7B, for example, inspector 715 includes a snap-to-grid property. Instead of loading a pre-constructed pane for this property, the method of the present invention dynamically constructs an appropriate pane (in this case, a simple check box), as shown in substep 1054a. An automatic list, on the other hand, is typically a simple group box (i.e., one having a plurality of radio buttons), which is preferably constructed dynamically, as shown by the inspector window 650 of FIG. 6H. In either case, the method may build appropriate inspector dialog components on-the-fly by simply examining a particular property and discerning its possible attributes. In a similar manner, "pick" lists of properties may be constructed and displayed, as shown in substep 1054c. A property pick list serves as an index to other property lists or other functions, as desired. By dynamically building inspectors for simpler properties, the method conserves system resources.

Construction of the inspector window is completed by inserting dialog controls (e.g., "OK", "CANCEL", and "HELP") for operating the dialog. In addition, an example window is displayed for indicating various changes to the properties of an object. This is accomplished by sending a message to the inspected object setting forth the current properties in the dialog; the inspected object returns an example or sample object which may then be displayed in the window. After changing a property, the dialog tab or button (e.g., tab 601a of FIG. 6A) corresponding to that property is updated (e.g., different font or screen color) for indicating that a change has been entered.

After constructing the property inspector dialog or window, at step 1055 the method enters a user event loop where the user inspects and sets various properties of the object by accessing the property (through the screen button or tab) and setting a new attribute. At the conclusion of the user event (e.g., by selecting "OK" or "CANCEL"), the user event is terminated.

At step 1056, the property list for the object being inspected is updated for any changes which occurred to the properties during the user event (of step 1055). This is accomplished by calling a setProperty method, which conceptually loops through each pane which has changed and updates the property list, accordingly. By way of illustration, the setProperty method may be defined as:

```
virtual BOOL setProperty( LPCTSTR lpPropStr, LPCTSTR lpValueStr,
                        PROTOTYPE pt = 0 );
```

The method receives the name of a property, either passed as a text string (lpPropStr) or as a handle (pt), and a value for that property, typically passed as a text string (lpValueStr).

Referring now to FIG. 10D, the general steps of a setProperty method 1070 are illustrated. In step 1071, the property text string is translated; alternatively, the property is referenced directly by passing a handle (i.e., integer or word value). At steps 1072-1073, the property is updated for the property value passed (e.g., by a CASE statement construct). If the property is not acted upon, it may be passed up the object's inheritance chain for action by an ancestor object, at step 1074. In this fashion, the values collected from the various property panes are used by the method to set the properties within the object.

While the foregoing outlines the general steps or framework for setProperty, the method is in fact defined as a virtual method. Thus, each class of objects may assume responsibility for setting its own properties. For an Annotate object (i.e., screen objects found in Graph windows), for example, an exemplary setProperty method may be constructed as follows:

5,590,259

23

24

```

BOOL Annotate::setProperty ( LPSTR lpPropStr, LPSTR lpValueStr, PROPTYPE
pt )
{
    WORD w;
    if ((w = info.propIndex (lpPropStr, pt)) < annoEndProps) {
        if (info.ppProps [w].getType() > prop_Compound)
            switch (w) {
                case annoFrame:
                    setFrame (GlobalFrameProp.x,
GlobalFrameProp.y,
GlobalFrameProp.w,
GlobalFrameProp.h);
                    break;
            }
        if ((info.ppProps [w].getProp( ))->stringToValue (lpValueStr,
pt)) {
            BOOL f = TRUE;
            switch (w) {
                case annoFrame:
                    setFrame (GlobalFrameProp.x,
GlobalFrameProp.y,
GlobalFrameProp.w,
GlobalFrameProp.h);
                    break;
                case annoHidden:
                    if (GlobalYesNoProp.index)
                        hide (TRUE);
                    else
                        show (TRUE);
                    break;
                case annoShow:
                    if (GlobalYesNoProp.index)
                        show (TRUE);
                    else
                        hide (TRUE);
                    break;
                case annoId:
                    if (pCW && pCW->infoPtr()->objType ==
ot_Dialog)
                        f = ((DialogView
*)pCW->getContentView( ))->setIdx (this, GlobalWordProp.w);
                    else
                        idx = GlobalWordProp.w;
                    break;
                case annoAttach:
                    flags.nochild = !GlobalYesNoProp.index;
                    break;
                case annoPosition:
                    *(WORD *)&flags = (*(WORD *)&flags
& ~(vfPOSITION | vfLEFTREL |
vfTOPREL | vfRIGHTREL | vfBOTTOMREL
| vfTOPTYPE | vfRIGHTTYPE)) |
GlobalPositionProp.flags;
                    flags.bottomType = flags.topType;
                    flags.lcftType = flags.rightType;
                    break;
            }
            if (f)
                callConnection (lpPropStr, lpValueStr, pt);
            return f;
        }
        else
            return FALSE;
    }
    else
        return Tracker::setProperty (lpPropStr, lpValueStr, pt);
}

```

As shown, the override method processes properties specific for Annotate (e.g., change frame, ID, position, and the like). In the event that a property is not identified, the property is passed up the inheritance chain for appropriate processing (by a parent's setProperty method). In this manner, an individual object (e.g., an Annotate object) in the system is designed to know how to get (getProperty) and set (setProperty) its own properties.

After the update of step 1056, the method 1050 concludes. At this point, the system is ready for another inspection.

Alternatively, the user may undertake other activities, as desired.

In parent application No. 07/866,658, now U.S. Pat. No. 5,416,895, is a microfiche Appendix A containing C++ source code listings, which provide a description of the invention suitable for use in a general purpose digital computer system, such as an IBM-compatible personal computer. A suitable compiler for compiling and linking C++ code is available from Borland International.

5,590,259

25

While the invention is described in some detail with specific reference to a single preferred embodiment and certain alternatives, there is no intent to limit the invention to that particular embodiment or those specific alternatives. For example, the property inspection of the present invention has been illustrated with mouse devices. Those skilled in the art, however, will appreciate that other input devices, including trackballs, joysticks, light pens, and the like, may be employed. Moreover, the present invention may be advantageously implemented in a variety of UI platforms other than Windows, including Macintosh, X-Windows, Motif, NextStep, OS/2, and the like. Thus, the true scope of the present invention is not limited to any one of the foregoing exemplary embodiments but is instead defined by the following claims.

What is claimed is:

1. In a computer system having a three-dimensional spreadsheet, said three-dimensional spreadsheet comprising a plurality of information cells arranged in a three-dimensional matrix, said information cells storing user-supplied data and formulas operative on said user-supplied data, each said information cell being uniquely identified by a three-dimensional cell address comprising three-dimensional coordinates, wherein at least one cell stores a formula having a cell address for referencing information stored by another cell, an improved method for creating a formula with cell references to information cells in said three-dimensional spreadsheet, the method comprising:

(a) dividing said three-dimensional matrix of information cells into a plurality of two-dimensional matrices, each of said two-dimensional matrices being represented on a display device as a page array of information cells, wherein cell references between cells on one page array are formed by specifying horizontal and vertical coordinates, and cell references between cells on different page arrays are formed by specifying horizontal, vertical, and page coordinates;

(b) displaying with each said page array of information cells a tab identifier, so that each tab identifier of a particular page array displays the page coordinate for that particular page array;

(c) receiving as input from a user a descriptive name for naming a page array;

(d) in response to step (c), assigning said descriptive name to the page coordinate for the named page array; and

(e) creating a formula having at least one cell reference to a cell on said named page array from a cell on a different page array, so that said at least one cell reference includes said descriptive name as the page coordinate for said named page array.

2. The method of claim 1, wherein said displaying step includes:

displaying each tab identifier along a horizontal side of said plurality of two-dimensional matrices.

3. The method of claim 1, wherein said displaying step includes:

displaying each tab identifier along a bottom side of said plurality of two-dimensional matrices.

4. The method of claim 1, wherein said descriptive name includes at least one user-settable identifying character.

5. The method of claim 1, wherein said descriptive name initially is set by said system to a default character.

6. The method of claim 5, wherein a particular formula stored on said second spreadsheet page includes a cell reference to a cell on said first spreadsheet page, and wherein said method further comprises the step, performed

26

by said system in response to changing said at least one user-settable identifying character for the spreadsheet page identifier of said first spreadsheet page to a user-supplied page name label, of changing the page identifier for the cell reference of said particular formula to said user-supplied page name label.

7. The method of claim 1, wherein said formula stores said cell reference to the cell on said different spreadsheet page with a general format of:

<page name label of first spreadsheet page> : <cell column><cell row>.

8. The method of claim 1, further comprising:

receiving user input for selecting one of said plurality of said two-dimensional matrices; and

indicating to the user a currently-selected two-dimensional matrix by displaying the notebook tab for the two-dimensional matrix with a different appearance than notebook tabs for others of said plurality of spreadsheet pages.

9. The method of claim 1, further comprising the step, performed by said system in response to user input, of grouping together desired ones of said two-dimensional matrices, so that a spreadsheet operation in any one two-dimensional matrix of the group propagates to other two-dimensional matrices of the group.

10. The method of claim 9, wherein said grouping step comprises:

receiving user input for selecting a two-dimensional matrix;

receiving user input for selecting a second two-dimensional matrix, the user input defining a selection of two-dimensional matrices spanning from said first selected two-dimensional matrix to said second selected two-dimensional matrix;

receiving user input for instructing the system to group together two-dimensional matrices; and

grouping together all two-dimensional matrices spanning from said first selected two-dimensional matrix to said second selected two-dimensional matrix.

11. The method of claim 10, wherein said receiving user input for selecting a first two-dimensional matrix comprises receiving user input for selecting the notebook tab for said first selected two-dimensional matrix.

12. In an electronic spreadsheet system for storing and manipulating information, a computer-implemented method of representing a three-dimensional spreadsheet on a screen display, the method comprising:

displaying on said screen display a first spreadsheet page from a plurality of spreadsheet pages, each of said spreadsheet pages comprising an array of information cells arranged in row and column format, at least some of said information cells storing user-supplied information and formulas operative on said user-supplied information, each of said information cells being uniquely identified by a spreadsheet page identifier, a column identifier, and a row identifier;

while displaying said first spreadsheet page, displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page, each said spreadsheet page identifier being displayed as an image of a notebook tab on said screen display and indicating a single respective spreadsheet page, wherein at least one spreadsheet page identifier of said displayed row of spreadsheet page identifiers comprises at least one user-settable identifying character;

receiving user input for requesting display of a second spreadsheet page in response to selection with an input

5,590,259

27

device of a spreadsheet page identifier for said second spreadsheet page;

in response to said receiving user input step, displaying said second spreadsheet page on said screen display in a manner so as to obscure said first spreadsheet page from display while continuing to display at least a portion of said row of spreadsheet page identifiers; and receiving user input for entering a formula in a cell on said second spreadsheet page, said formula including a cell reference to a particular cell on another of said spreadsheet pages having a particular spreadsheet page identifier comprising at least one user-supplied identifying character, said cell reference comprising said at least one user-supplied identifying character for said particular spreadsheet page identifier together with said column identifier and said row identifier for said particular cell.

13. The method of claim 12, wherein said displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page step includes:

displaying said row of spreadsheet page identifiers along a horizontal side of said first spreadsheet page.

14. The method of claim 12, wherein said displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page step includes:

displaying said row of spreadsheet page identifiers along a bottom side of said first spreadsheet page.

15. The method of claim 12, wherein said at least one user-settable identifying character for said first spreadsheet page is initially set by said system to a default name.

16. The method of claim 12, further comprising the step, performed by said computer system in response to receiving user input, of changing said at least one user-settable identifying character for the spreadsheet page identifier of said first spreadsheet page to a user-supplied page name label describing contents of said first spreadsheet page.

17. The method of claim 16, wherein a particular formula stored on said second spreadsheet page includes a cell reference to a cell on said first spreadsheet page, and wherein said method further comprises the step, performed by said system in response to changing said at least one user-settable identifying character for the spreadsheet page identifier of said first spreadsheet page to a user-supplied page name label, of changing the page identifier for the cell reference of said particular formula to said user-supplied page name label.

18. The method of claim 17, wherein said particular formula stores said cell reference to the cell on said first spreadsheet page with a general format of:

<page name label of first spreadsheet page> : <cell column> <cell row>.

19. The method of claim 12, further comprising:

receiving user input for selecting one of said plurality of spreadsheet pages; and

indicating to the user a currently-selected spreadsheet page by displaying the notebook tab for said selected spreadsheet page with a different appearance than notebook tabs for others of said plurality of spreadsheet pages.

20. The method of claim 19, wherein said indicating step includes displaying said notebook tab of said selected spreadsheet page as an extension of said selected spreadsheet page and displaying notebook tabs for others of said plurality of spreadsheet pages as if extending out from under said selected spreadsheet page.

21. The method of claim 19, wherein said indicating step includes displaying said notebook tab of said selected

28

spreadsheet page with a first screen background color and displaying said notebook tabs of others of said plurality of spreadsheet pages with a second screen background color.

22. The method of claim 21, wherein said first screen background color is white, and wherein said second screen background color is gray.

23. The method of claim 12, wherein said displaying a row of spreadsheet page identifiers step includes displaying, at any given time, only spreadsheet page identifiers for some of said plurality of spreadsheet pages.

24. The method of claim 23, further comprising:

receiving user input for displaying a particular spreadsheet page identifier which is not currently displayed and, thereupon, displaying said particular spreadsheet page identifier.

25. The method of claim 12, further comprising the step, performed by said system in response to user input, of modifying values displayed in cells in said second spreadsheet page when displaying said second spreadsheet page.

26. The method of claim 12, further comprising the step, performed by said system in response to user input, of grouping together desired ones of said spreadsheet pages, so that a spreadsheet operation in any one spreadsheet page of the group propagates to other spreadsheet pages of the group.

27. The method of claim 26, wherein said grouping step comprises:

receiving user input for selecting a first spreadsheet page; receiving user input for selecting a second spreadsheet page, the user input defining a selection of spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page;

receiving user input for instructing the system to group together spreadsheet pages; and

grouping together all spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page.

28. The method of claim 27, wherein said receiving user input for selecting a first spreadsheet page comprises receiving user input for selecting the notebook tab for said first selected spreadsheet page.

29. An electronic spreadsheet system for storing and manipulating information comprising:

a computer having a processor and a memory, said memory storing said information in a three-dimensional spreadsheet;

a screen display for presenting said three-dimensional spreadsheet as a plurality of spreadsheet pages, each of said spreadsheet pages comprising an array of information cells arranged in row and column format, at least some of said information cells storing user-supplied information, each of said information cells being uniquely identified by a spreadsheet page identifier, a column identifier and a row identifier;

a storage device for storing said plurality of spreadsheet pages;

means for displaying a first spreadsheet page on said screen device and simultaneously displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page, each said spreadsheet page identifier being displayed as an image of a notebook tab on said screen display and indicating a single respective spreadsheet page, wherein at least one spreadsheet page identifier of said displayed row of spreadsheet page identifiers comprises at least one user-settable identifying character;

5,590,259

29

a first input means for receiving user input for requesting display of a second spreadsheet page in response to selection with said first input means of a spreadsheet page identifier for said second spreadsheet page;

means, responsive to said first input means, for displaying said second spreadsheet page on said screen display in a manner so as to obscure said first spreadsheet page from display while continuing to display at least a portion of said row of spreadsheet page identifiers; and

a second input means for receiving user input for entering a formula in a cell on said second spreadsheet page, said formula including a cell reference to a particular cell on another of said spreadsheet pages having a particular spreadsheet page identifier comprising at least one user-supplied identifying character, said cell reference comprising said at least one user-supplied identifying character for said particular spreadsheet page identifier together with said column identifier and said row identifier for said particular cell.

30. The system of claim 29, wherein said means for displaying a first spreadsheet page on said screen device and simultaneously displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page includes:

means for displaying said row of spreadsheet page identifiers along a horizontal side of said first spreadsheet page.

31. The system of claim 29, wherein said means for displaying a first spreadsheet page on said screen device and simultaneously displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page includes:

means for displaying said row of spreadsheet page identifiers along a bottom side of said first spreadsheet page.

32. The system of claim 29, wherein said at least one user-settable identifying character for said first spreadsheet page is initially set by said system to a default name.

33. The system of claim 29, further comprising:

means, responsive to user input, for changing said at least one user-settable identifying character for the spreadsheet page identifier of said first spreadsheet page to a user-supplied page name label describing contents of said first spreadsheet page.

34. The system of claim 33, wherein a particular formula stored on said second spreadsheet page includes a cell reference to a cell on said first spreadsheet page, and wherein said system further comprises:

means, responsive to changes of said at least one user-settable identifying character for the spreadsheet page identifier of said first spreadsheet page to a user-supplied page name label, for changing the page identifier for the cell reference of said particular formula to said user-supplied page name label.

35. The system of claim 34, wherein said particular formula stores in said memory said cell reference to the cell on said first spreadsheet page with a general format of:

<page name label of first spreadsheet page> : <cell column> <cell row>.

36. The system of claim 29, further comprising:

means for receiving user input for selecting one of said plurality of spreadsheet pages; and

means for indicating to the user a currently-selected spreadsheet page, said indicating means including means for displaying the notebook tab for said selected spreadsheet page with a different appearance than notebook tabs for others of said plurality of spreadsheet pages.

37. The system of claim 36, wherein said indicating means includes:

30

means for displaying said notebook tab of said selected spreadsheet page as an extension of said selected spreadsheet page; and

means for displaying notebook tabs for others of said plurality of spreadsheet pages as if extending out from under said selected spreadsheet page.

38. The system of claim 36, wherein said indicating means includes:

means for displaying said notebook tab of said selected spreadsheet page with a first screen background color; and

means for displaying said notebook tabs of others of said plurality of spreadsheet pages with a second screen background color.

39. The system of claim 38, wherein said first screen background color is white, and wherein said second screen background color is gray.

40. The system of claim 29, wherein said means for displaying a row of spreadsheet page identifiers includes:

means for displaying, at any given time, only spreadsheet page identifiers for some of said plurality of spreadsheet pages.

41. The system of claim 40, further comprising:

means for receiving user input for displaying on said screen display a particular spreadsheet page identifier which is not currently displayed; and

means, responsive to said means for receiving user input for displaying a particular spreadsheet page identifier which is not currently displayed, for displaying said particular spreadsheet page identifier.

42. The system of claim 29, further comprising:

means, responsive to user input, for modifying values displayed in cells in said second spreadsheet page while displaying on said screen display said second spreadsheet page.

43. The method of claim 29, further comprising:

means, responsive to user input, for grouping together desired ones of said spreadsheet pages, so that a spreadsheet operation in any one spreadsheet page of the group propagates to other spreadsheet pages of the group.

44. The system of claim 43, wherein said grouping means comprises:

means for receiving user input for selecting a first spreadsheet page;

means for receiving user input for selecting a second spreadsheet page, the user input defining a selection of spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page;

means for receiving user input for instructing the system to group together spreadsheet pages; and

means for grouping together all spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page.

45. The system of claim 44, wherein said means for receiving user input for selecting a first spreadsheet page comprises:

means for receiving user input for selecting the notebook tab for said first selected spreadsheet page.

46. In an electronic spreadsheet system for storing and manipulating information, a computer-implemented method of representing a three-dimensional spreadsheet on a screen display, the method comprising:

displaying on said screen display a first spreadsheet page from a plurality of spreadsheet pages, each of said

5,590,259

31

spreadsheet pages comprising an array of information cells arranged in row and column format, at least some of said information cells storing user-supplied information and formulas operative on said user-supplied information, each of said information cells being uniquely identified by a spreadsheet page identifier, a column identifier, and a row identifier;

while displaying said first spreadsheet page, displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page, each said spreadsheet page identifier being displayed as an image of a notebook tab on said screen display and indicating a single respective spreadsheet page, wherein at least one spreadsheet page identifier of said displayed row of spreadsheet page identifiers comprises at least one user-settable identifying character;

in response to first user input for selecting a spreadsheet page identifier for a second spreadsheet page, displaying said second spreadsheet page on said screen display in a manner so as to obscure said first spreadsheet page from display while continuing to display at least a portion of said row of spreadsheet page identifiers;

in response to second user input for entering a formula in a cell on said second spreadsheet page, said formula including a cell reference to a particular cell on another of said spreadsheet pages having a particular spreadsheet page identifier comprising at least one user-supplied identifying character, creating a cell reference comprising said at least one user-supplied identifying character for said particular spreadsheet page identifier together with said column identifier and said row identifier for said particular cell;

in response to third user input for requesting a change in said at least one user-supplied identifying character for said particular spreadsheet page identifier, (a) changing said at least one user-settable identifying character for the spreadsheet page identifier of said particular spreadsheet page, and (b) changing the page identifier for the cell reference of said particular formula to said at least one user-supplied identifying character, so changed;

in response to fourth user input for grouping together desired ones of said spreadsheet pages, grouping together said desired ones of said spreadsheet pages so that a subsequent spreadsheet operation in any one spreadsheet page of the group propagates to other spreadsheet pages of the group.

47. The method of claim 46, wherein said displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page step includes:

displaying said row of spreadsheet page identifiers along a horizontal side of said first spreadsheet page.

32

48. The method of claim 46, wherein said displaying a row of spreadsheet page identifiers along one side of said first spreadsheet page step includes:

displaying said row of spreadsheet page identifiers along a bottom side of said first spreadsheet page.

49. The method of claim 46, wherein said at least one user-settable identifying character for said first spreadsheet page is initially set by said system to a default name.

50. The method of claim 46, wherein said particular formula stores said cell reference to the cell on said particular spreadsheet page with a general format of:

<particular spreadsheet page identifier> : <cell column> <cell row.

51. The method of claim 46, further comprising:

receiving user input for selecting one of said plurality of spreadsheet pages; and

indicating to the user a currently-selected spreadsheet page by displaying the notebook tab for said selected spreadsheet page with a difference appearance than notebook tabs for others of said plurality of spreadsheet pages.

52. The method of claim 46, wherein said displaying a row of spreadsheet page identifiers step includes displaying, at any given time, only spreadsheet page identifiers for some of said plurality of spreadsheet pages.

53. The method of claim 52, further comprising:

receiving user input for displaying a particular spreadsheet page identifier which is not currently displayed and, thereupon, displaying said particular spreadsheet page identifier.

54. The method of claim 46, further comprising the step, performed by said system in response to user input, of modifying values displayed in cells in said second spreadsheet page when displaying said second spreadsheet page.

55. The method of claim 46, wherein said grouping step comprises:

receiving user input for selecting a first spreadsheet page;

receiving user input for selecting a second spreadsheet page, the user input defining a selection of spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page;

receiving user input for instructing the system to group together spreadsheet pages; and

grouping together all spreadsheet pages spanning from said first selected spreadsheet page to said second selected spreadsheet page.

56. The method of claim 55, wherein said receiving user input for selecting a first spreadsheet page comprises receiving user input for selecting the notebook tab for said first selected spreadsheet page.

* * * * *

EXHIBIT 7



United States Patent [19]

[11] Patent Number: **6,154,844**

Touboul et al.

[45] Date of Patent: **Nov. 28, 2000**

[54] **SYSTEM AND METHOD FOR ATTACHING A DOWNLOADABLE SECURITY PROFILE TO A DOWNLOADABLE**

Primary Examiner—Robert W. Beausoliel, Jr.
 Assistant Examiner—Christopher A. Revak
 Attorney, Agent, or Firm—Squire, Sanders & Dempsey, L.L.P.

[75] Inventors: **Shlomo Touboul**, Kefar-Haim;
Nachshon Gal, Tel-Aviv, both of Israel

[57] **ABSTRACT**

[73] Assignee: **Finjan Software, Ltd.**, San Jose, Calif.

A system comprises an inspector and a protection engine. The inspector includes a content inspection engine that uses a set of rules to generate a Downloadable security profile corresponding to a Downloadable, e.g., Java™ applets, ActiveX™ controls, JavaScript™ scripts, or Visual Basic scripts. The content inspection engine links the Downloadable security profile to the Downloadable. The set of rules may include a list of suspicious operations, or a list of suspicious code patterns. The first content inspection engine may link to the Downloadable a certificate that identifies the content inspection engine which created the Downloadable security profile. Additional content inspection engines may generate and link additional Downloadable security profiles to the Downloadable. Each additional Downloadable security profile may also include a certificate that identifies its creating content inspection engine. Each content inspection engine preferably creates a Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds. The protection includes a Downloadable interceptor for receiving a Downloadable, a file reader coupled to the interceptor for determining whether the Downloadable includes a Downloadable security profile, an engine coupled to the file reader for determining whether to trust the Downloadable security profile, and a security policy analysis engine coupled to the verification engine for comparing the Downloadable security profile against a security policy if the engine determines that the Downloadable security profile is trustworthy. A Downloadable ID verification engine retrieves the Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds, generates the Downloadable ID for the Downloadable and compares the generated Downloadable to the linked Downloadable. The protection engine further includes a certificate authenticator for authenticating the certificate that identifies a content inspection engine which created the Downloadable security profile as from a trusted source. The certificate authenticator can also authenticate a certificate that identifies a developer that created the Downloadable.

[21] Appl. No.: **08/995,648**

[22] Filed: **Dec. 22, 1997**

Related U.S. Application Data

[60] Provisional application No. 60/030,639, Nov. 8, 1996.

[51] Int. Cl.⁷ **H04L 9/36**

[52] U.S. Cl. **713/201; 714/38; 713/164**

[58] Field of Search 713/201, 200,
 713/202, 164, 165, 166, 167, 176; 714/38,
 704, 207, 33; 709/229; 380/4, 25, 24; 705/51,
 54, 55

References Cited

U.S. PATENT DOCUMENTS

5,077,677 12/1991 Murphy et al. 395/10
 5,359,659 10/1994 Rosenthal 380/4
 5,361,359 11/1994 Tajalli et al. 395/700

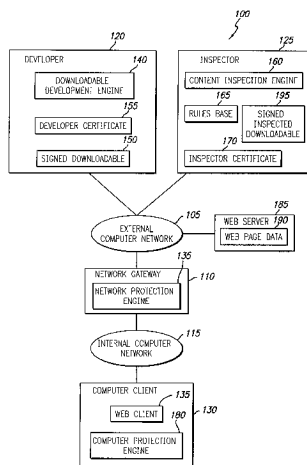
(List continued on next page.)

OTHER PUBLICATIONS

X.N. Zhang, "Secure Code Distribution," Computer, pp. 76–79, Jun. 1997.
 IBM AntiVirus User's Guide Version 2.4, International Business Machines Corporation, Nov. 15, 1995, pp. 6–7.
 Jim K. Omura, "Novel Applications of Cryptography in Digital Communications", IEEE Communications Magazine, May, 1990; pp. 21–27.
 Norvin Leach et al, "IE 3.0 Applets Will Earn Certification", PC Week, v13, n29, 1998, 2 pages.
 Microsoft Authenticode Technology, "Ensuring Accountability and Authenticity for Software Components on the Internet", Microsoft Corporation, Oct. 1996, including contents, Introduction and pp. 1–10.

(List continued on next page.)

44 Claims, 7 Drawing Sheets



6,154,844

Page 2

U.S. PATENT DOCUMENTS

5,485,409	1/1996	Gupta et al.	395/186
5,485,575	1/1996	Chess et al.	395/183.14
5,572,643	11/1996	Judson	395/793
5,623,600	4/1997	Ji et al.	395/187.01
5,638,446	6/1997	Rubin	380/25
5,692,047	11/1997	McManis	380/4
5,692,124	11/1997	Holden et al.	395/187.01
5,720,033	2/1998	Deo	395/186
5,724,425	3/1998	Chang et al.	380/25
5,740,248	4/1998	Fieres et al.	380/25
5,761,421	6/1998	van Hoff et al.	395/200.53
5,765,205	6/1998	Breslau et al.	711/203
5,784,459	7/1998	Devarakonda et al.	380/4
5,796,952	8/1998	Davis et al.	395/200.54
5,805,829	9/1998	Cohen et al.	395/200.32
5,832,208	11/1998	Chen et al.	395/187.01
5,850,559	12/1998	Angelo et al.	395/750.03
5,859,966	1/1999	Hayman et al.	713/200
5,864,683	1/1999	Boebert et al.	395/200.79
5,892,904	4/1999	Atkinson et al.	713/201
5,956,481	9/1999	Walsh et al.	713/200
5,974,549	10/1999	Golan	713/200
5,983,348	11/1999	Ji	713/200

OTHER PUBLICATIONS

Web Page, Article “Frequently Asked Questions About Authenticode”, Microsoft Corporation, last updated Feb. 17, 1997, URL: <http://www.microsoft.com/workshop/security/authcode/signfaq.asp#9>, pp. 1–13.

Web page: [“Finjan Announces a Personal Java™ Firewall for Web Browsers—the SurfinShield™ 1.6”, Press Release of Finjan Releases SurfinShield, Oct. 21, 1996, 2 pages.](http://iel.ihs.com:80/cgi-bin/iel13.cgi?se...2ehts%26ViewTemplate%3ddocview%5fb%2ehts,Okamoto, E. et al., “ID-Based Authentication System For Computer Virus Detection”, IEEE/IEE Electronic Library online, Electronics Letters, vol. 26, Issue 15, ISSN 0013-5194, Jul. 19, 1990, Abstract and pp. 1169-1170.</p>
</div>
<div data-bbox=)

“Finjan Software Releases SurfinBoard, Industry’s First JAVA Security Product For the World Wide Web”, Article published on the Internet by Finjan Software, Ltd., Jul. 29, 1996, 1 page.

“Powerful PC Security for the New World of Java™ and Downloadables, Surfin Shield™” Article published on the Internet by Finjan Software Ltd., 1996, 2 pages.

“Company Profile Finjan—Safe Surfing, The Java Security solutions Provider” Article published on the Internet by Finjan Software Ltd., Oct. 31, 1996, 3 pages.

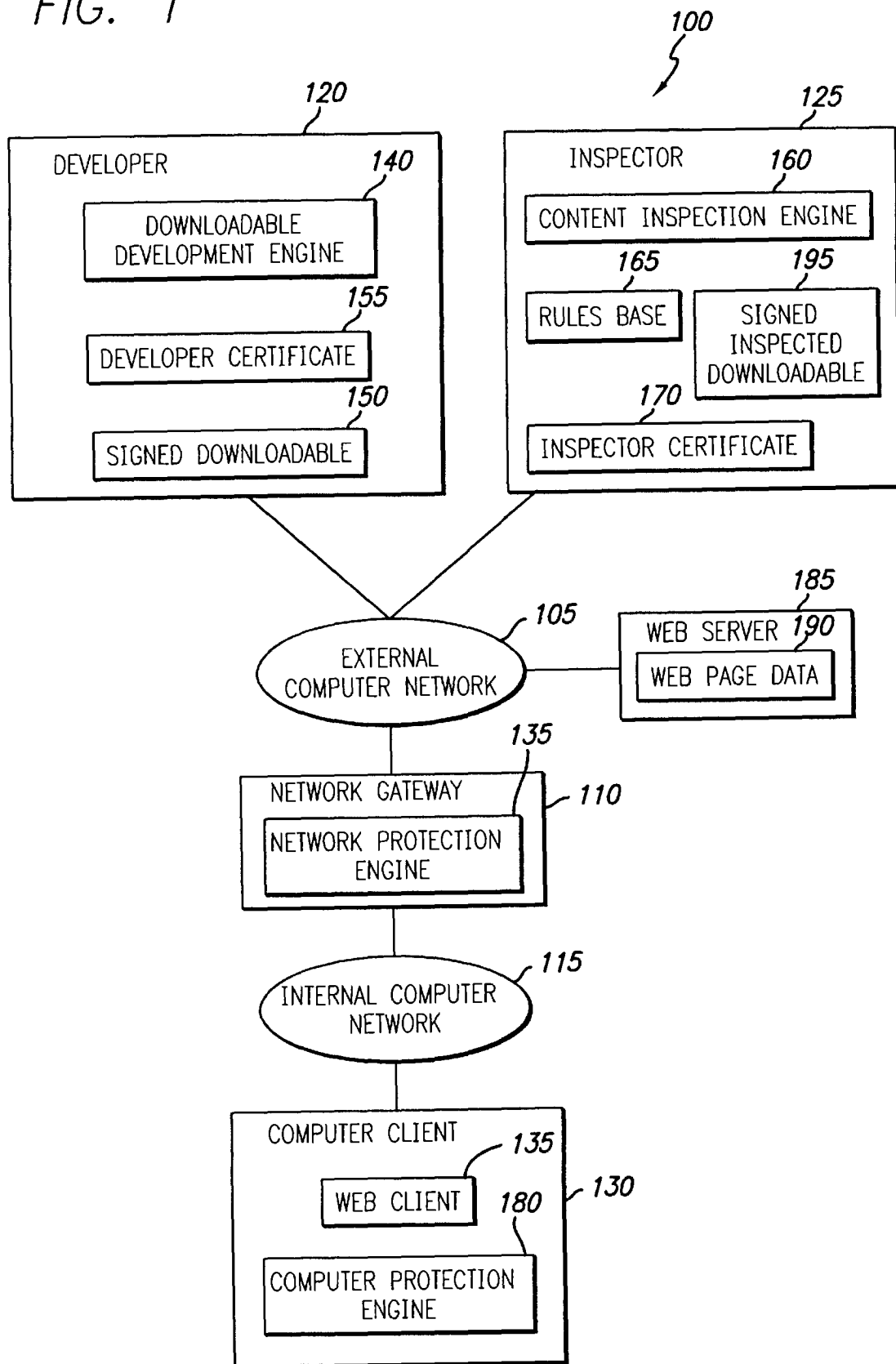
“Finjan Announces Major Power Boost and New Features for SurfinShield™ 2.0” Las Vegas Convention Center/Pavillion 5 P5551, Nov. 18, 1996, 3 pages.

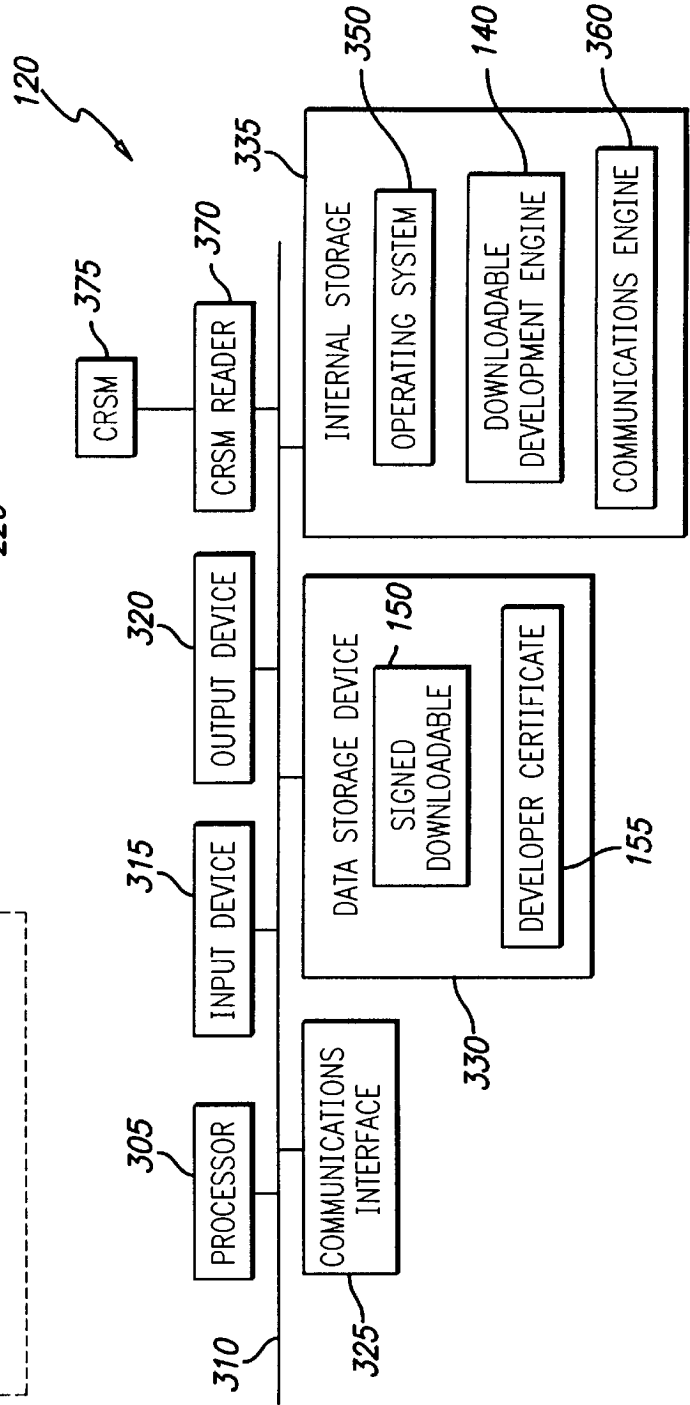
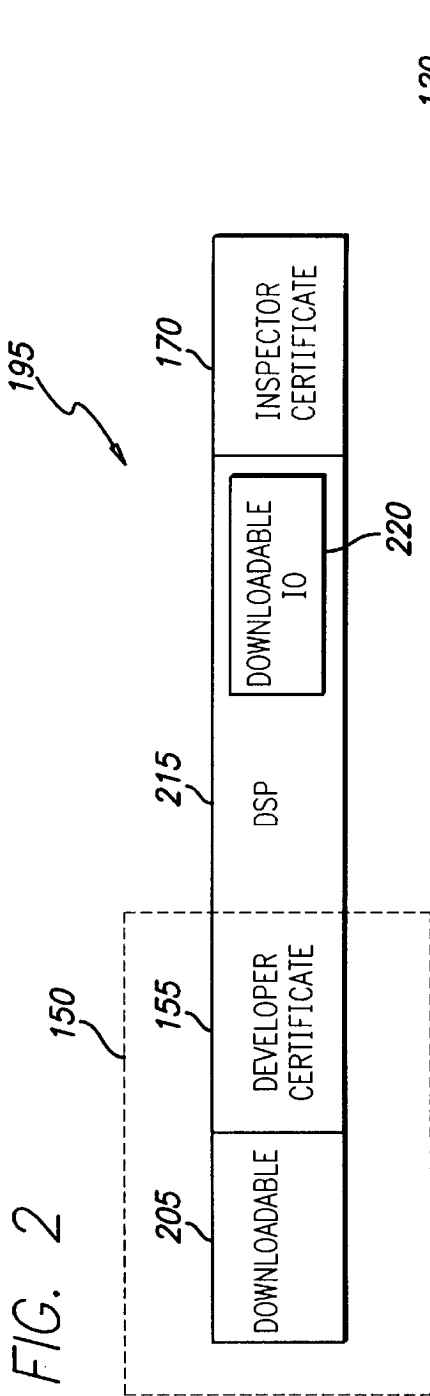
“Java Security: Issues & Solutions” Article published on the Internet by Finjan Software Ltd., 1996, 8 pages.

“Products” Article published on the Internet, 7 pages.

Mark LaDue, “Online Business Consultant” Article published on the Internet, Home Page, Inc. 1996, 4 pages.

FIG. 1





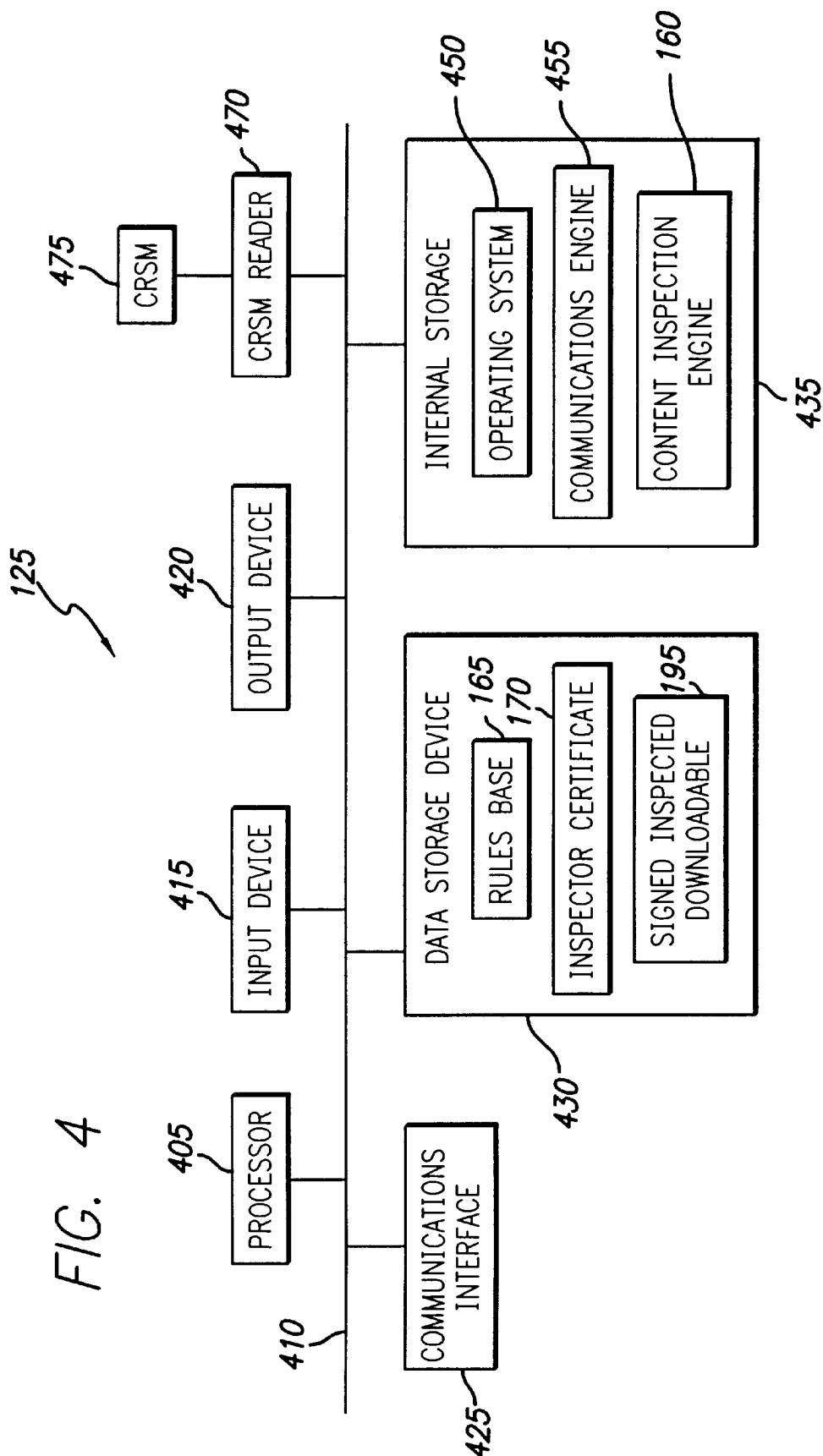


FIG. 4

FIG. 5

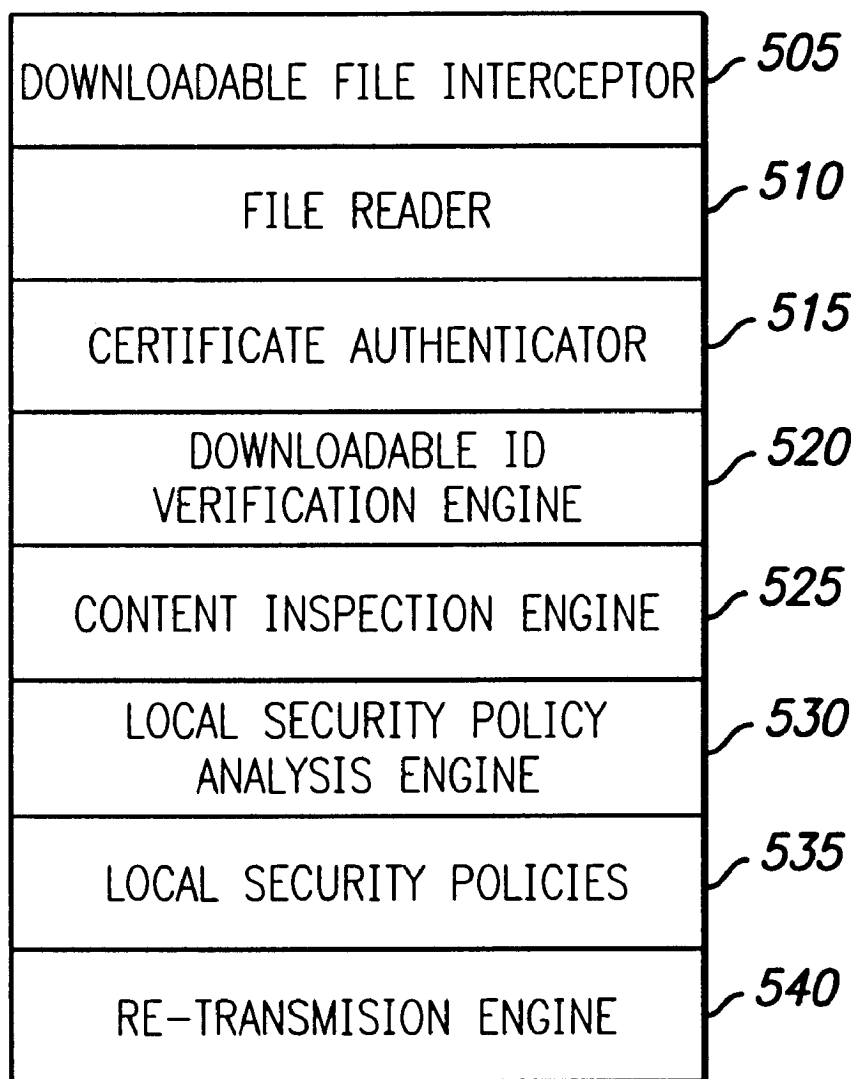
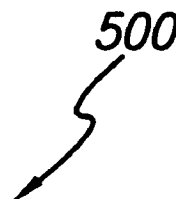


FIG. 6

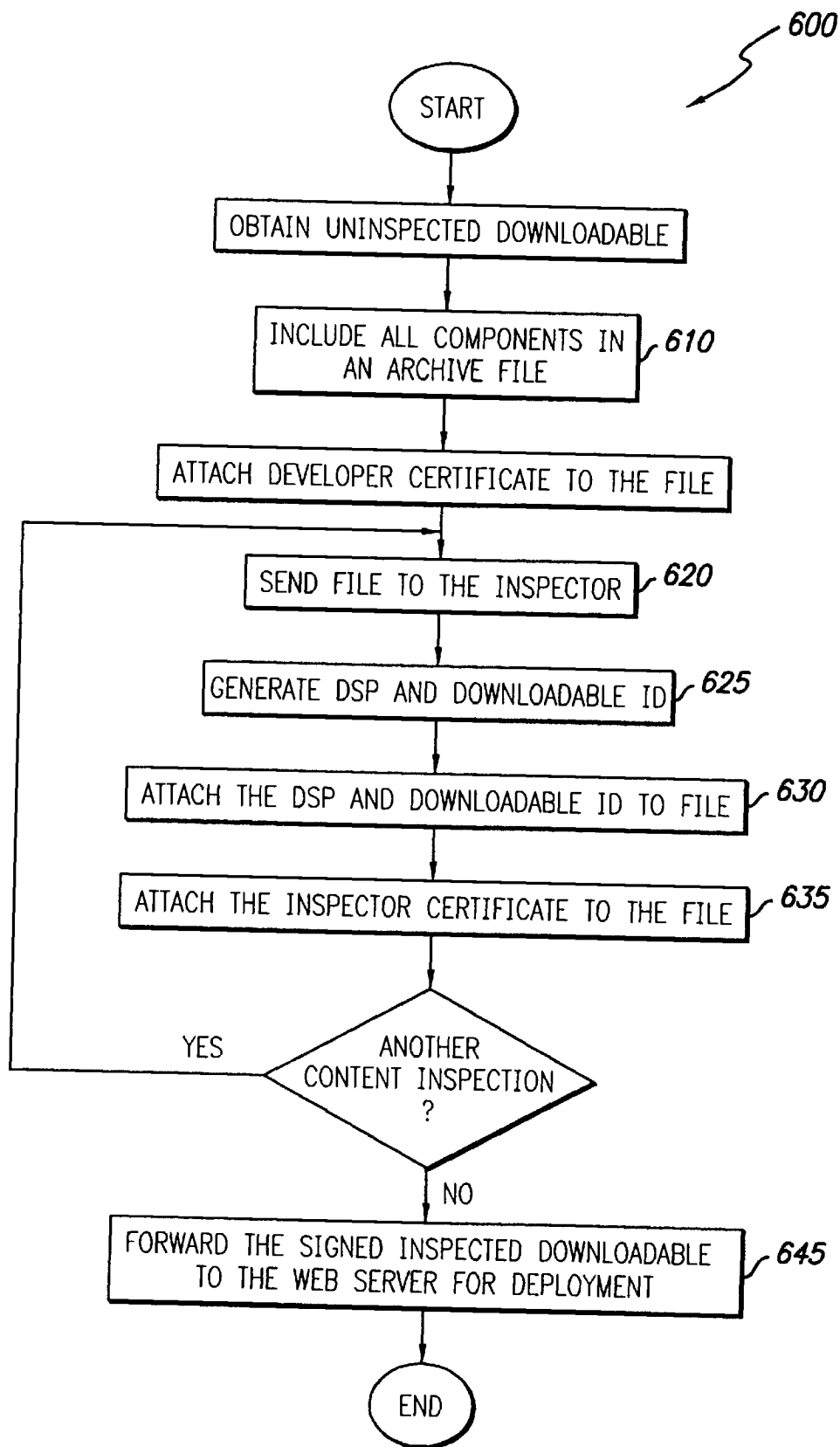
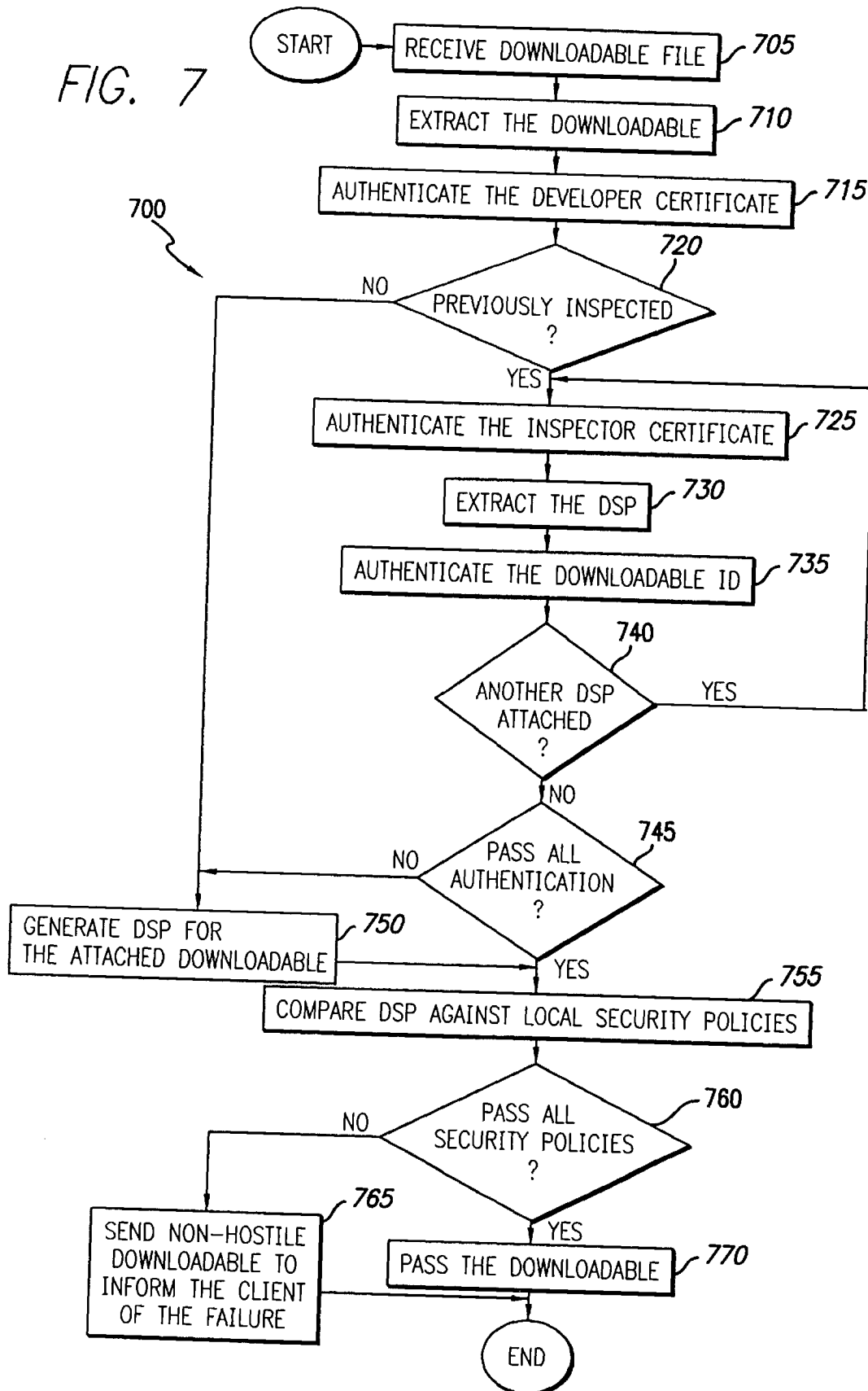


FIG. 7



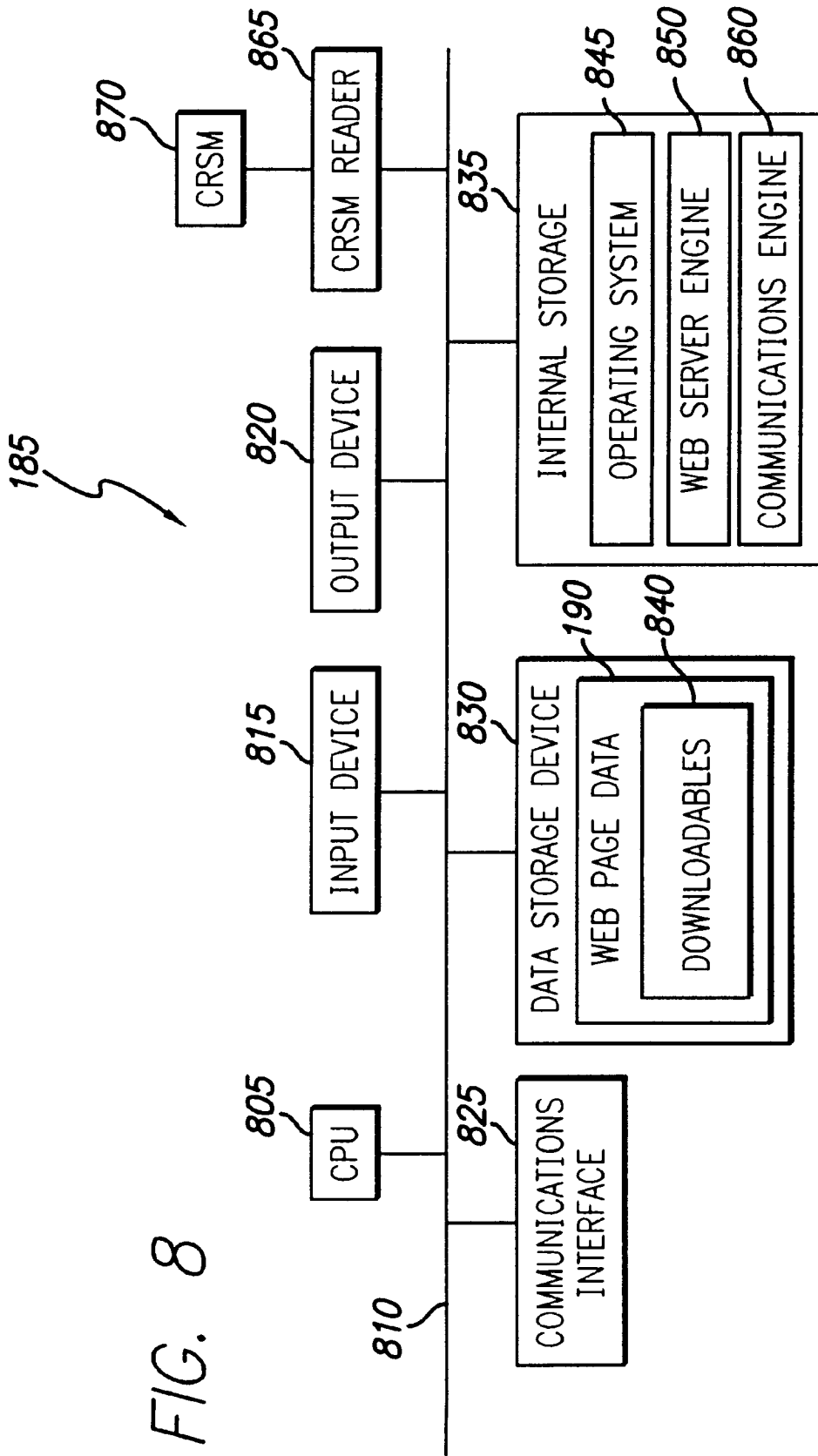


FIG. 8

6,154,844

1

SYSTEM AND METHOD FOR ATTACHING A DOWNLOADABLE SECURITY PROFILE TO A DOWNLOADABLE

PRIORITY REFERENCE TO RELATED APPLICATIONS

This application claims benefit of and hereby incorporates by reference provisional application Ser. No. 60/030,639, entitled "System and Method for Protecting a Computer from Hostile Downloadables," filed on Nov. 8, 1996, by inventor Shlomo Touboul; patent application Ser. No. 08/964,388, entitled "System and Method for Protecting a Computer and a Network from Hostile Downloadables," filed on Nov. 6, 1997, by inventor Shlomo Touboul; and patent application Ser. No. 08/790,097, entitled "System and Method for Protecting a Client from Hostile Downloadables," filed on Jan. 29, 1997, also by inventor Shlomo Touboul.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for attaching a Downloadable security profile to a Downloadable to facilitate the protection of computers and networks from a hostile Downloadable.

2. Description of the Background Art

The Internet is currently a collection of over 100,000 individual computer networks owned by governments, universities, nonprofit groups and companies, and is expanding at an accelerating rate. Because the Internet is public, the Internet has become a major source of many system damaging and system fatal application programs, commonly referred to as "viruses."

Accordingly, programmers continue to design computer and computer network security systems for blocking these viruses from attacking both individual and network computers. On the most part, these security systems have been relatively successful. However, these security systems are not configured to recognize computer viruses which have been attached to or configured as Downloadable application programs, commonly referred to as "Downloadables." A Downloadable is an executable application program, which is downloaded from a source computer and run on the destination computer. A Downloadable is typically requested by an ongoing process such as by an Internet browser or web client. Examples of Downloadables include Java™ applets designed for use in the Java™ distributing environment developed by Sun Microsystems, Inc., JavaScript™ scripts also developed by Sun Microsystems, Inc., ActiveX™ controls designed for use in the ActiveX™ distributing environment developed by the Microsoft Corporation, and Visual Basic also developed by the Microsoft Corporation. Downloadables may also include plugins, which add to the functionality of an already existing application program. Therefore, a system and method are needed to protect a network from hostile Downloadables.

SUMMARY OF THE INVENTION

The present invention provides systems for protecting a network from suspicious Downloadables, e.g., Java™ applets, ActiveX™ controls, JavaScript™ scripts, or Visual Basic scripts. The network system includes an inspector for linking Downloadable security profiles to a Downloadable, and a protection engine for examining the Downloadable

2

and Downloadable security profiles to determine whether or not to trust the Downloadable security profiles.

The inspector includes a content inspection engine that uses a set of rules to generate a Downloadable security profile corresponding to a Downloadable. The content inspection engine links the Downloadable security profile to the Downloadable. The set of rules may include a list of suspicious operations, or a list of suspicious code patterns. The first content inspection engine may link to the Downloadable a certificate that identifies the content inspection engine which created the Downloadable security profile. The system may include additional content inspection engines for generating and linking additional Downloadable security profiles to the Downloadable. Each additional Downloadable security profile may also include a certificate that identifies its creating content inspection engine. Each content inspection engine may create a Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds.

The protection engine includes a Downloadable interceptor for receiving a Downloadable, a file reader coupled to the interceptor for determining whether the Downloadable includes a Downloadable security profile, an engine coupled to the file reader for determining whether to trust the Downloadable security profile, and a security policy analysis engine coupled to the verification engine for comparing the Downloadable security profile against a security policy if the engine determines that the Downloadable security profile is trustworthy. The engine preferably determines whether the first Downloadable security profile corresponds to the Downloadable. The system preferably includes a Downloadable ID verification engine for retrieving a Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds. To confirm the correspondence between the Downloadable security profile and the Downloadable, the Downloadable ID verification engine generates the Downloadable ID for the Downloadable and compares the generated Downloadable to the linked Downloadable. The system may also include a content inspection engine for generating a Downloadable security profile for the Downloadable if the first Downloadable security profile is not trustworthy. The system further includes a certificate authenticator for authenticating a certificate that identifies a content inspection engine which created the Downloadable security profile as from a trusted source. The certificate authenticator can also authenticate a certificate that identifies a developer that created the Downloadable.

The present invention provides a method in a first embodiment comprising the steps of receiving a Downloadable, generating a first Downloadable security profile for the received Downloadable, and linking the first Downloadable security profile to the Downloadable. The present invention further provides a method in a second embodiment comprising the steps of receiving a Downloadable with a linked first Downloadable security profile, determining whether to trust the first Downloadable security profile, and comparing the first Downloadable security profile against the security policy if the first Downloadable security profile is trustworthy.

It will be appreciated that the system and method of the present invention may provide computer protection from known hostile Downloadables. The system and method of the present invention may identify Downloadables that perform operations deemed suspicious. The system and method of the present invention may examine the Downloadable code to determine whether the code contains any

6,154,844

3

suspicious operations, and thus may allow or block the Downloadable accordingly. It will be appreciated that, because the system and method of the present invention link a verifiable Downloadable security profile to a Downloadable, the system and method may avoid decomposing the Downloadable into the Downloadable security profile on the fly.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a network system in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of an example inspected Downloadable of FIG. 1;

FIG. 3 is a block diagram illustrating details of a developer of FIG. 1;

FIG. 4 is a block diagram illustrating details of an inspector of FIG. 1;

FIG. 5 is a block diagram illustrating details of a generic protection engine of FIG. 1;

FIG. 6 is a flowchart illustrating a method for attaching a Downloadable security profile to a Downloadable in accordance with the present invention;

FIG. 7 is a flowchart illustrating a method for examining a Downloadable in accordance with the present invention; and

FIG. 8 is a block diagram illustrating details of the web server of FIG. 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a computer network system 100 in accordance with the present invention. The computer network system 100 includes an external computer network 105, such as the Wide Area Network (WAN) commonly referred to as the Internet, coupled via a network gateway 110 to an internal computer network 115, such as a Local Area Network (LAN) commonly referred to as an intranet. The network system 100 further includes a developer 120 coupled to the external computer network 105, an inspector 125 also coupled to the external computer network 105, a web server 185 also coupled to the external computer network 105, and a computer client 130 coupled to the internal computer network 115. One skilled in the art will recognize that connections to external or internal network systems are merely exemplary, and alternative embodiments may have other connections. Further, although the developer 120, inspector 125 and web server 185 are being described as distinct sites, one skilled in the art will recognize that these elements may be a part of an integral site, may each include components of multiple sites, or may include combinations of single and multiple sites.

The developer 120 includes a Downloadable development engine 140 for generating a signed (yet uninspected) Downloadables 150. The developer 120 may obtain an uninspected Downloadable or may initially use the Downloadable development engine 140 to generate an uninspected Downloadable. The developer 120 can then use the Downloadable development engine 140 to transmit the signed Downloadable to the inspector 125 for hostility inspection. The developer 120 includes a developer certificate 155, which the Downloadable development engine 140 attaches to each uninspected Downloadable so that the inspector 125, the network gateway 110 and the computer client 130 can authenticate the developer 120.

The inspector 125 includes a content inspection engine 160 for examining a received Downloadable, e.g., the signed

4

Downloadable 150 received from the developer 120, for generating a Downloadable Security Profile (DSP) based on a rules base 165 for the Downloadable, and for attaching the DSP to the Downloadable. A DSP preferably includes a list of all potentially hostile or suspicious computer operations that may be attempted by the Downloadable, and may also include the respective arguments of these operations. Generating a DSP includes searching the Downloadable code for any pattern, which is undesirable or suggests that the code was written by a hacker. The content inspection engine 160 preferably performs a full-content inspection. It will be appreciated that generating a DSP may also include comparing a Downloadable against Downloadables which Original Equipment Manufacturers (OEMs) know to be hostile, Downloadables which OEMs know to be non-hostile, and Downloadables previously examined by the content inspection engine 160. Accordingly, the rules base may include a list of operations and code patterns deemed suspicious, known hostile Downloadables, known viruses, etc.

An Example List of Operations Deemed Suspicious

File operations: READ a file, WRITE a file, DELETE a file, RENAME a file;

Network operations: LISTEN on a socket, CONNECT to a socket, SEND data, RECEIVE data, VIEW INTRANET;

Registry operations: READ a registry item, WRITE a registry item;

Operating system operations: EXIT WINDOWS, EXIT BROWSER, START PROCESS/THREAD, KILL PROCESS/THREAD, CHANGE PROCESS/THREAD PRIORITY, DYNAMICALLY LOAD A CLASS/LIBRARY, etc.; and

Resource usage thresholds: memory, CPU, graphics, etc.

Further, the content inspection engine 160 generates and attaches a Downloadable ID to the Downloadable. The Downloadable ID is typically stored as part of the DSP, since multiple DSPs may be attached to a Downloadable and each may have a different Downloadable ID. Preferably, to generate a Downloadable ID, the content inspection engine 160 computes a digital hash of the complete Downloadable code. The content inspection engine 160 preferably prefetches all components embodied in or identified by the code for Downloadable ID generation. For example, the content inspection engine 160 may prefetch all classes embodied in or identified by the Java™ applet bytecode, and then may perform a predetermined digital hash on the Downloadable code (and the retrieved components) to generate the Downloadable ID. Similarly, the content inspection engine 160 may retrieve all components listed in the .INF file for an ActiveX™ control to compute a Downloadable ID. Accordingly, the Downloadable ID for the Downloadable will be the same each time the content inspection engine 160 (or a protection engine as illustrated in FIG. 5) receives the same Downloadable and applies the same digital hash function. The downloadable components need not be stored with the Downloadable, but can be retrieved before each use or Downloadable ID generation.

Generating a DSP and generating a Downloadable ID are described in great detail with reference to the patent application Ser. No. 08/964,388, entitled "System and Method for Protecting a Computer and a Network from Hostile Downloadables," filed on Nov. 6, 1997, by inventor Shlomo Touboul, which has been incorporated by reference above.

After performing content inspection, the inspector 125 attaches an inspector certificate 170 to the Downloadable. The inspector certificate 170 verifies the authenticity of the

6,154,844

5

DSP attached to the Downloadable. Details of an example signed inspected Downloadable **150** are illustrated and described with reference to FIG. 2. The inspector **125** then transmits the signed inspected Downloadable **195** to the web server **185** for addition to web page data **190** and web page deployment. Accordingly, the computer client **130** includes a web client **175** for accessing the web page data **190** provided by the web server **185**. As is known in the art, upon recognition of a Downloadable call, the web client **175** requests the web server **185** to forward the corresponding Downloadable. The web server **185** then transmits the Downloadable via the network gateway **110** to the computer client **130**.

The network gateway **110** includes network protection engine **135**, and the computer client **130** includes a computer protection engine **180**. Both the network protection engine **135** and the computer protection engine **180** examine all incoming Downloadables and stop all Downloadables deemed suspicious. It will be appreciated that a Downloadable is deemed suspicious if it performs or may perform any undesirable operation, or if it threatens or may threaten the integrity of any computer component. It is to be understood that the term "suspicious" includes hostile, potentially hostile, undesirable, potentially undesirable, etc. Thus, if the incoming Downloadable includes a signed inspected Downloadable **195**, then the network protection engine **135** and the computer protection engine **180** can review the attached certificates to verify the authenticity of the DSP. If the incoming Downloadable does not include a signed inspected Downloadable **195**, then each of the network protection engine **135** and the computer protection engine **180** must generate the DSP, and compare the DSP against local security policies (**535**, FIG. 5).

Components and operation of the network protection engine **135** and the computer protection engine **180** are described in greater detail with reference to FIG. 5. It will be appreciated that the network gateway **110** may include the components described in the patent-application Ser. No. 08/964,388, entitled "System and Method for Protecting a Computer and a Network from Hostile Downloadables," filed on Nov. 6, 1997, by inventor Shlomo Touboul, which has been incorporated by reference above. It will be further appreciated that the computer protection engine **180** may include the components described in the patent application Ser. No. 08/790,097, entitled "System and Method for Protecting a Client from Hostile Downloadables," filed on Jan. 29, 1997, also by inventor Shlomo Touboul.

It will be appreciated that the network system **100** may include multiple inspectors **125**, wherein each inspector **125** may provide a different content inspection. For example, one inspector **125** may examine for suspicious operations, another inspector **125** may examine for known viruses that may be attached to the Downloadable **150**, etc. Each inspector **125** would attach a corresponding DSP and a certificate verifying the authenticity of the attached DSP. Alternatively, a single inspector **125** may include multiple content inspection engines **160**, wherein each engine provides a different content inspection.

FIG. 2 is a block diagram illustrating details of a signed inspected Downloadable **195**, which includes a Downloadable **205**, a developer certificate **155**, a DSP **215** which includes a Downloadable ID **220**, and an inspector certificate **170**. The Downloadable **205** includes the downloadable and executable code that a web client **175** receives and executes. The Downloadable **205** may be encrypted using the developer's private key. The attached developer certificate **155** may include the developer's public key, the devel-

6

oper's name, an expiration date of the key, the name of the certifying authority that issued the certificate, and a serial number. The signed Downloadable **150** comprises the Downloadable **205** and the developer certificate **155**. The DSP **215** and Downloadable ID **220** may be encrypted by the inspector's private key. The Downloadable ID **220** is illustrated as part of the DSP **215** for simplicity, since each signed inspected Downloadable **195** may include multiple DSPs **215** (and each DSP **215** may include a separate and distinct Downloadable ID **220**). The inspector certificate **170** may include the inspector's public key, an expiration date of the key, the name of the certifying authority that issued the certificate, and a Ser. No.

Although the signed inspected Downloadable **195** illustrates the DSP **215** (and Downloadable ID **220**) as an attachment, one skilled in the art will recognize that the DSP **215** can be linked to the Downloadable **205** using other techniques. For example, the DSP **215** can be stored in the network system **100**, and alternatively a pointer to the DSP **215** can be attached to the signed inspected Downloadable **195**. The term "linking" herein will be used to indicate an association between the Downloadable **205** and the DSP **215** (including using a pointer from the Downloadable **195** to the DSP **215**, attaching the DSP **215** to the Downloadable **205**, etc.)

FIG. 3 is a block diagram illustrating details of the developer **120**, which includes a processor **305**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a signal bus **310**. The developer **120** further includes an input device **315** such as a keyboard and mouse, an output device **320** such as a Cathode Ray Tube (CRT) display, a data storage device **330** such as a magnetic disk, and an internal storage **335** such as Random-Access Memory (RAM), each coupled to the signal bus **310**. A communications interface **325** couples the signal bus **325** to the external computer network **105**, as shown in FIG. 1.

An operating system **350** controls processing by processor **305**, and is typically stored in the data storage device **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**. The Downloadable development engine **140** generates signed Downloadables **150** as described above, and also may be stored in the data storage device **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**. The data storage device **330** stores the signed Downloadables **150** and the developer certificate **155**. A communications engine **360** controls communications via the communications interface **325** with the external computer network **105**, and also may be stored in the data storage device **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**.

One skilled in the art will understand that the developer **120** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system in alternative ways. For example, a computer-readable storage medium (CRSM) reader **370** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a computer-readable storage medium (CRSM) **375** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the developer **120** may receive programs and data via the CRSM reader **370**.

FIG. 4 is a block diagram illustrating details of the inspector **125**, which includes a processor **405**, such as an

6,154,844

7

Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a signal bus **410**. The inspector **125** further includes an input device **415** such as a keyboard and mouse, an output device **420** such as a CRT display, a data storage device **430** such as a magnetic disk, and an internal storage **435** such as RAM, each coupled to the signal bus **410**. A communications interface **425** couples the signal bus **425** to the external computer network **105**, as shown in FIG. 1.

An operating system **450** controls processing by processor **405**, and is typically stored in the data storage device **430** and loaded into internal storage **435** (as illustrated) for execution by processor **405**. The content inspection engine **160** performs a content inspection of Downloadables from the developer **120** and attaches the results of the content inspection. The content inspection engine **160** also may be stored in the data storage device **330** and loaded into internal storage **335** (as illustrated) for execution by processor **405**. The data storage device **330** stores the rules base **165**, the inspector certificate **170** and the signed inspected Downloadable **195**. A communications engine **455** controls communications via the communications interface **425** with the external computer network **105**, and also may be stored in the data storage device **430** and loaded into internal storage **435** (as illustrated) for execution by processor **405**.

One skilled in the art will understand that the inspector **125** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system in alternative ways. For example, a computer-readable storage medium (CRSM) reader **470** such as a magnetic disk drive, hard disk drive, magneto-optical reader, processor, etc. may be coupled to the signal bus **410** for reading a computer-readable storage medium (CRSM) **475** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the inspector **125** may receive programs and data via the CRSM reader **470**.

FIG. 5 is a block diagram illustrating details of a generic protection engine **500**, which exemplifies each of the network protection engine **135** and the computer protection engine **180**. The generic protection engine **500** includes a Downloadable file interceptor **505** for intercepting incoming Downloadables (i.e., Downloadable files) for inspection, and a file reader **510** for opening the received Downloadable file and initiating appropriate components.

If the enclosed Downloadable includes a signed inspected Downloadable **195**, the file reader **510** initiates execution of a certificate authenticator **515**. The certificate authenticator **515** verifies the authenticity of the developer certificate **155** and the authenticity of the inspector certificate **170**. One skilled in the art will appreciate that certificate verification may include using authenticated or known public keys to decrypt the certificates or the enclosed files. A Downloadable ID verification engine **520** regenerates the Downloadable ID for the enclosed Downloadable **150**, and compares the regenerated Downloadable ID against the enclosed Downloadable ID **220**. If the received Downloadable fails any of the above tests (or if the received Downloadable is not a signed inspected Downloadable **195**), then a local content inspection engine **525** generates a DSP for the enclosed Downloadable **205**. Otherwise, if the received Downloadable file passes all of the above tests, then the content inspection engine **525** trusts the attached DSP **215** and thus need not generate a DSP. The content inspection

8

engine **525** is similar to the content inspection engine **160** of the inspector **125**. One skilled in the art will recognize that the generic protection engine **500** may include multiple content inspection engines **525** for performing distinct content examinations.

A local security policy analysis engine **530** compares the attached or generated DSP against local security policies **535**. The local security policies **535** may include a list of specific Downloadables to block, a list of specific Downloadables to allow, generic rules to apply regardless of the intended recipient and recipient's status, specific rules to apply based on the intended recipient or the intended recipient's status, trusted certificates, etc. If the received Downloadable passes all the local security policies **535**, a retransmission engine **540** passes the Downloadable onward to the intended recipient for execution.

It will be appreciated that components of the generic protection engine **500** may include components described in the patent applications incorporated by reference above. For example, the local security policies **535** in this application may include the security policies **305** of the patent application Ser. No. 08/964,388, entitled "System and Method for Protecting a Computer or a Network from Hostile Downloadables," filed on Nov. 6, 1997, by inventor Shlomo Touboul; the content inspection engine **525** and the content inspection engine **160** each may include the code scanner **325** of the same patent application; the certificate authenticator **515** may include the certificate scanner **340** and the certificate comparator **345** of the same patent application, the local security policy analysis engine **530** may include the access control lists comparator **330** and the logical engine **333** of the same patent; the Downloadable ID verification engine **520** may include the ID generator **315** of the same patent application; and the file reader **510** may include the first comparator **320** of the same patent application.

FIG. 6 is a flowchart illustrating a method **600** for inspecting a Downloadable **205** in accordance with the present invention. Method **600** begins with the developer **120** in step **605** obtaining or using the Downloadable development engine **140** to create a Downloadable **205**. The Downloadable development engine **140** in step **610** includes all components, e.g., all Java™ classes for a Java™ applet, into a Downloadable archive file, and in step **615** attaches the developer certificate **155** to the archive file, thereby creating a signed Downloadable **150**.

The Downloadable development engine **140** in step **620** transmits the signed Downloadable **150** to the inspector **125**. The content inspection engine **160** in step **625** generates a DSP **215** and a Downloadable ID **220** for the Downloadable **150**. As stated above, generating a DSP includes examining the Downloadable **205** (and the Downloadable components) for all suspicious operations that will or may be performed by the Downloadable, all suspicious code patterns, all known viruses, etc. Generating a DSP may include comparing all operations that will or may be performed against a list of suspicious operations or against a list of rules, e.g., a rules base **165**. Accordingly, if an operation in the Downloadable **205** matches one of the suspicious operations or violates one of the rules, then the operation is listed in the DSP **215**. Generating a Downloadable ID **220** includes computing a digital hash of the Downloadable **205** (and the Downloadable components), so that the Downloadable ID is identical for each instance of the same Downloadable **205**.

The content inspection engine **160** in step **630** attaches the DSP **215** and the Downloadable ID **220** to the Downloadable archive file, i.e., to the signed Downloadable **150**. The

6,154,844

9

content inspection engine 160 in step 635 attaches the inspector certificate 170 to the file, thereby providing authentication of the attached DSP 215 (including the Downloadable ID 220).

The inspector 125 in step 640 determines whether another content inspection is to be effected. If so, then method 600 returns to step 600 to send the file to the next inspector 620. One skilled in the art will recognize that the same inspector 125 may be used to perform another content inspection, attachment of another DSP 215 and Downloadable ID 220 and attachment of another inspector certificate 170. If the inspector 125 determines in step 640 that no more content inspection is to be effected, then the inspector 125 forwards the signed inspected Downloadable 195 to the web server 185 for addition to web page data 190 and web engine deployment. Accordingly, the web client 175 can access the web page data 190, and thus can retrieve the signed inspected Downloadable 195. Method 600 then ends.

FIG. 7 is a flowchart illustrating a method 700 for examining a Downloadable (whether or not signed and inspected). Method 700 begins with the Downloadable file interceptor 505 in step 705 receiving a Downloadable file. The file reader 510 in step 710 extracts the Downloadable 150, and in step 715 instructs the certificate authenticator 515 to authenticate the developer certificate 155 as from a trusted developer 120. Developer certificate authentication may include retrieving the public key for the developer 120 from a trusted source, and using the public key to decrypt the Downloadable 205.

The file reader 510 in step 720 determines whether an inspector 125 has previously inspected the received Downloadable. If the received Downloadable has not been inspected, then method 700 jumps to step 750. Otherwise, the file reader 510 initiates the certificate authenticator 515 in step 725 to authenticate the inspector certificate 170 as from a trusted inspector 125. Inspector certificate authentication may include retrieving the public key for the inspector 125 from a trusted source, and using the public key to decrypt the attached DSP 215 (including the Downloadable ID 220). The file reader 510 in step 730 extracts the DSP 215 (including the Downloadable ID 220), and in step 735 instructs the Downloadable ID verification engine 520 to authenticate the Downloadable ID 220. That is, the Downloadable ID verification engine 520 performs the same digital hash on the Downloadable 205 (including all components) to regenerate the Downloadable ID. If the components are not included in the file, then the Downloadable ID verification engine 520 may prefetch the components. Thus, if the regenerated Downloadable ID is the same as the attached Downloadable ID 220, then the Downloadable ID verification engine 520 verifies that the attached DSP 215 corresponds to the attached Downloadable 205.

The file reader 510 in step 740 determines whether another DSP 215 is attached to the file. If so, then method 700 returns to step 725. Otherwise, the content inspection engine 525 in step 745 determines whether the Downloadable 205 passed or failed any of the authentication tests above. If the Downloadable 205 failed any of the steps (or if as stated above the received Downloadable did not include a signed inspected Downloadable 195), then method 700 jumps to step 750.

The content inspection engine 525 in step 750 generates a DSP (or DSPs) for the received Downloadable as described above with reference to FIGS. 1-6. Otherwise, if the Downloadable 205 passed all the steps, then the content inspection engine 525 indicates that the DSP 215 (or DSPs

10

215) attached to the received Downloadable may be trusted. The local security policy analysis engine 530 in step 755 compares the DSP 215 (or DSPs 215), whether generated on the fly or extracted from the file, against local security policies 535. As stated above, the local security policies 535 may include a rules base 165 that identifies suspicious operations, suspicious code patterns, known viruses, etc. One skilled in the art will appreciate that the security policies 535 may depend on the type of DSP 215. For example, the local security policy analysis engine 530 may compare a first attached DSP 215 against the list of suspicious operations and suspicious code patterns, and may compare a second attached DSP 215 against known viruses.

The content inspection engine 160 in step 760 determines whether each DSP 215 passes all corresponding security policies 535. If each DSP 215 passes, then the local security policy analysis engine 530 in step 770 instructs the retransmission engine 540 to pass the Downloadable. If a DSP 215 fails, then the local security policy analysis engine 530 in step 765 stops the Downloadable and sends a substitute non-hostile Downloadable to the computer client 130 to inform the computer client 130 of the failure. Method 700 then ends.

FIG. 8 is a block diagram illustrating details of the web server 185, which includes a processor 805, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a signal bus 810. The web server 185 further includes an input device 815 such as a keyboard and mouse, an output device 820 such as a Cathode Ray Tube (CRT) display, a data storage device 830 such as a magnetic disk, and an internal storage 835 such as Random-Access Memory (RAM), each coupled to the signal bus 810. A communications interface 825 couples the signal bus 825 to the external computer network 105, as shown in FIG. 1.

An operating system 845 controls processing by processor 805, and is typically stored in the data storage device 830 and loaded into internal storage 835 (as illustrated) for execution by processor 805. A web server engine 850 controls web engine access to the web page data 190, and also may be stored in the data storage device 830 and loaded into internal storage 835 (as illustrated) for execution by processor 805. The data storage device 330 stores the web page data 190, which may include Downloadables 840 (whether or not signed and inspected). A communications engine 860 controls communications via the communications interface 825 with the external computer network 105, and also may be stored in the data storage device 830 and loaded into internal storage 835 (as illustrated) for execution by processor 805.

One skilled in the art will understand that the web server 185 may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system in alternative ways. For example, a computer-readable storage medium (CRSM) reader 865 such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus 310 for reading a computer-readable storage medium (CRSM) 870 such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the web server 185 may receive programs and data via the CRSM reader 865.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other

6,154,844

11

variations of the above-described embodiments and methods are provided by the present invention. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. Connections may be wired, wireless, modem, etc. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A method comprising:

receiving by an inspector a Downloadable;

generating by the inspector a first Downloadable security profile that identifies suspicious code in the received Downloadable; and

linking by the inspector the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.

2. The method of claim 1, wherein the first Downloadable security profile is linked to the Downloadable via attachment.

3. The method of claim 1, wherein the first Downloadable security profile is linked to the Downloadable via a pointer.

4. The method of claim 1, further comprising the step of linking to the first Downloadable security profile a Downloadable ID that identifies the Downloadable.

5. The method of claim 1, wherein the Downloadable includes a Java™ applet.

6. The method of claim 1, wherein the Downloadable includes an ActiveX™ control.

7. The method of claim 1, wherein the Downloadable includes a JavaScript™ script.

8. The method of claim 1, wherein the Downloadable includes a Visual Basic script.

9. The method of claim 1, further comprising the step of linking to the Downloadable a certificate that identifies the developer which created the Downloadable.

10. The method of claim 1, further comprising the step of linking to the Downloadable a certificate that identifies a first content inspection engine which generated the first Downloadable security profile.

11. The method of claim 1, wherein the first Downloadable security profile includes a list of operations deemed suspicious by the inspector.

12. The method of claim 1, further comprising the steps of generating a second Downloadable security profile, and linking the second Downloadable security profile to the received Downloadable.

13. The method of claim 12, further comprising the steps of linking a first certificate that identifies a first content inspection engine which generated the first Downloadable security profile, and linking a second certificate that identifies a second content inspection engine which generated the second Downloadable security profile.

14. The method of claim 13, wherein each of the first Downloadable security profile and the second Downloadable security profile includes a Downloadable ID identifying the received Downloadable.

15. An inspector system comprising:

memory storing a first rule set; and

a first content inspection engine for using the first rule set to generate a first Downloadable security profile that identifies suspicious code in a Downloadable, and for linking the first Downloadable security profile to the

12

Downloadable before a web server makes the Downloadable available to web clients.

16. The system of claim 15, wherein the first rule set includes a list of suspicious operations.

17. The system of claim 15, wherein the first rule set include a list of suspicious code patterns.

18. The system of claim 15, wherein the first content inspection engine links to the Downloadable a certificate that identifies the first content inspection engine which created the first Downloadable security profile.

19. The system of claim 15, further comprising a second content inspection engine for generating a second Downloadable security profile, and for linking the second Downloadable security profile to the Downloadable.

20. The system of claim 15, wherein the first content inspection engine links to the Downloadable a first certificate that identifies the first content inspection engine which created the first Downloadable security profile, and wherein the second content inspection engine links to the Downloadable a second certificate that identifies the second content inspection engine which created the second Downloadable security profile.

21. The system of claim 15, wherein the first content inspection engine creates a first Downloadable ID that identifies the Downloadable to which the first Downloadable security profile corresponds, and links the Downloadable ID to the Downloadable security profile.

22. A method performed by a network gateway comprising:

receiving a Downloadable with a linked Downloadable security profile that identifies suspicious code in the Downloadable, the Downloadable security profile being linked to the Downloadable before the web server make the Downloadable available to the web client; and

comparing the Downloadable security profile against a security policy.

23. A method performed by a network gateway comprising:

receiving a Downloadable with a linked first Downloadable security profile that identifies suspicious code in the Downloadable, the Downloadable security profile being linked to the Downloadable before the web server make the Downloadable available to the web client:

determining whether to trust the first Downloadable security profile; and

comparing the first Downloadable security profile against the security policy if the first Downloadable security profile is trustworthy.

24. The method of claim 23, wherein the step of determining includes determining whether the first Downloadable security profile corresponds to the Downloadable.

25. The method of claim 24, wherein a Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds is linked to the Downloadable security profile, and wherein the step of determining includes retrieving the linked Downloadable ID.

26. The method of claim 25, further comprising the steps of generating the Downloadable ID for the Downloadable, and comparing the generated Downloadable to the linked Downloadable.

27. The method of claim 23, further comprising the step of generating a Downloadable security profile for the Downloadable if the first Downloadable security profile is not trustworthy.

13

28. The method of claim 23, wherein a certificate that identifies a content inspection engine which created the Downloadable security profile is linked to the Downloadable security profile, and wherein the step of determining includes retrieving the certificate.

29. The method of claim 28, wherein the step of determining further includes authenticating the certificate as from a trusted source.

30. The method of claim 23, wherein a certificate that identifies a developer that created the Downloadable is linked to the Downloadable, and wherein the step of determining includes retrieving the certificate.

31. The method of claim 30, wherein the step of determining further includes authenticating the certificate as from a trusted developer.

32. A network gateway system comprising:

a Downloadable interceptor for receiving a Downloadable;

a file reader coupled to the interceptor for determining whether the Downloadable includes a linked Downloadable security profile that identifies suspicious code in the Downloadable, wherein if the Downloadable includes a linked Downloadable security profile, the Downloadable was linked before the web server makes the Downloadable available to the web client;

an engine coupled to the file reader for determining whether to trust the Downloadable security profile; and

a security policy analysis engine coupled to a verification engine for comparing the Downloadable security profile against a security policy if the engine determines that the Downloadable security profile is trustworthy.

33. The system of claim 32, wherein the engine determines whether the first Downloadable security profile corresponds to the Downloadable.

34. The system of claim 33, wherein a Downloadable ID that identifies the Downloadable to which the Downloadable security profile corresponds is linked to the Downloadable security profile, and wherein the engine includes a Downloadable ID verification engine for retrieving the linked Downloadable ID.

35. The system of claim 34, wherein the Downloadable ID verification engine generates the Downloadable ID for the Downloadable and compares the generated Downloadable to the linked Downloadable.

36. The system of claim 32, further comprising a content inspection engine coupled to the engine for generating a Downloadable security profile for the Downloadable if the first Downloadable security profile is not trustworthy.

37. The system of claim 32, wherein a certificate that identifies a content inspection engine which created the Downloadable security profile is linked to the Downloadable security profile, and wherein the engine retrieves the certificate.

38. The system of claim 37, wherein the engine includes a certificate authenticator for authenticating the certificate as from a trusted source.

14

39. The system of claim 32, wherein a certificate that identifies a developer that created the Downloadable is linked to the Downloadable, and wherein the engine retrieves the certificate.

40. The system of claim 39, wherein the engine includes a certificate authenticator for authenticating the certificate as from a trusted developer.

41. A computer-readable storage medium storing program code for causing a data processing system on an inspector to perform the steps of:

receiving a Downloadable;

generating a first Downloadable security profile that identifies suspicious code in the received Downloadable; and

linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.

42. A computer-readable storage medium storing program code for causing a data processing system on a network gateway to perform the steps of:

receiving a Downloadable with a linked first Downloadable security profile that identifies suspicious code in the Downloadable, the Downloadable security profile being linked to the Downloadable before the web server make the Downloadable available to the web client;

determining whether to trust the first Downloadable security profile; and

comparing the first Downloadable security profile against the security policy if the first Downloadable security profile is trustworthy.

43. An inspector system comprising:

means for receiving a Downloadable;

means for generating a first Downloadable security profile that identifies suspicious code in the received Downloadable; and

means for linking the first Downloadable security profile to the Downloadable before a web server makes the Downloadable available to web clients.

44. A network gateway system comprising:

means for receiving a Downloadable with a linked first Downloadable security profile that identifies suspicious code in the Downloadable, the Downloadable security profile being linked to the Downloadable before the web server make the Downloadable available to the web client;

means for determining whether to trust the first Downloadable security profile; and

means for comparing the first Downloadable security profile against the security policy if the first Downloadable security profile is trustworthy.