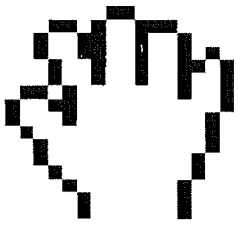
SIGCHI

April 1998
Volume 30, Number 2
ISSN 0736-6906



A Farewell to the Apple Advance Technology Group

acm PRESS

A Quarterly Publication of the ACM Special Interest Group on **Computer-Human Interaction**

SIGCHI Bulletin (ISSN 0736-6906) is published quarterly by ACM, 1515 Broadway, New York, NY 10036, USA. Periodicals postage paid at New York, NY 10001, and at additional mailing offices. POSTMASTER: Send address changes to SIGCHI Bulletin, ACM, 1515 Broadway, New York, NY 10036. Printed in USA.

Annual subscription cost of $17.19 is included in the member dues of $30 (for students, cost is included in $10); the non-member annual subscription is $57.

Scope. The scope of Special Interest Group on Computer-Human Interaction (SIGCHI) is the study of human factors in the human-computer interaction process, including research, design, development, and evaluation of interactive computing systems. The focus is on human communication and interaction with computer systems. SIGCHI provides a forum for the exchange of ideas among computer scientists, behavioral and cognitive scientists, system designers, and end users, and it serves as a clearinghouse of information for the field of human factors and user psychology research and development.

Membership. You are invited to join and participate in SIGCHI functions. Membership in SIGCHI, which includes a subscription to the *SIGCHI Bulletin*, is open to ACM members and non-members. For subscriptions and to become a member of SIGCHI (or for change of address), complete the form in the back of this issue. Correspondence regarding subscriptions, back issues, or membership should be sent to ACM Member Services, 1515 Broadway, New York, NY 10036, USA; email: ACMhelp@acm.org

Submissions. Materials for editorial consideration should be submitted to the appropriate editor. *Electronic submissions are strongly encouraged.* Please include an email address if possible, as well as a postal address and short biography. Materials suitable for publication include technical submissions, correspondence, announcements of research reports and publications, conference and workshop reports, news items, annotated bibliographies, book reviews and other items of general interest.

Deadlines. Copy deadlines are the 1st of the third month preceding date of issue e.g., October 1 for the January issue. For further submission instructions please refer to http://www.acm.org/sigchi/bulletin/

Opinions expressed in signed articles and letters are those of the writer and do not necessarily express the position of the ACM or SIGCHI. Reports and technical papers in the Bulletin are unrefereed working papers, unless otherwise stated. Materials may be reproduced for noncommercial purposes, if credit is given to the *SIGCHI Bulletin* and ACM/SIGCHI.

Advertising. ACM accepts recruitment advertising under the basic premise that the advertising employer does not discriminate on the basis of age, color, race, religion, gender, sexual preference, or national origin. ACM recognizes, however, that laws on such matters vary from country to country and contain exceptions, inconsistencies or contradictions. This is as true of laws of the United States of America as it is of other countries. Thus ACM policy requires each advertising employer to state explicitly in the advertisement any employment restrictions that may apply with respect to age, color, race, religion, gender, sexual preference, or national origin. Observance of the legal retirement age in the employer's country is not considered discrimination under this policy.

For advertising information, please contact Walter Andrzejewski (Advertising Manager) at ACM, 1515 Broadway, New York, NY 10036 USA; +1-212-869-7440; Fax: +1-212-869-0481. Email: acm-advertising@acm.org.

An Overview of the ATG Intelligent Systems Program

James R. Miller

The potential of intelligent user interfaces have been obvious for many years one only has to look at the *Knowledge Navigator* video or any of a large number of science fiction novels or films to appreciate what it could be like to interact with computers in ways analogous to how people interact with other people. However, the computing industry, throughout this time, has remained incapable of building systems with these capabilities. The technology demands are great, as are the human interaction design problems that must ultimately be solved to yield a system that is truly useful to people.

Consequentially, the Intelligent Systems Program chose a different approach: that of *partial understanding*. Since the dream of truly intelligent interaction lies beyond our capabilities, we worked to approximate this dream with technologies that were available to us. This led to a different approach to the intelligent interface question and to our work: we thought less about problems that have resisted solution for decades, and more about the real needs that users have and how they can best be satisfied. The tools of the intelligent interfaces community – parsers, inference engines, knowledge representation, language understanding – remained central our work, lurking beneath the surface. However, they were applied in ways that work today, and that yield successful solutions to user needs. One of these solutions – *Apple Data Detectors*[1] [5] – is shipping as an Apple product today; we hope and expect that other solutions based on this approach will join it soon.

Over the lifetime of the program, we applied this general strategy of partial understanding to several problem areas. These addressed different questions about user tasks and interaction styles, and presented different opportunities for technologies to help solve those problems. Several of the papers in this volume describe our experiences with this approach; in particular:

- *Apple Data Detectors and beyond*: Our work on Apple Data Detectors was based, in part, on a desire to move towards a broader notion of what constitutes a document, and how documents can be transformed from today's simple streams of characters into highly interactive computational objects. Our work on *LiveDoc* [4] and *Drop Zones* [3] outline increasingly powerful systems that take us in this direction by finding meaningful components of documents and making them the locus of task-based interaction.
- *Language-based knowledge mining*. Perhaps, at some time in the future, computers will be able to derive the same kinds of understandings of written documents that humans can. But what do we do until then? One approach, which we have pursued in our work on natural language processing, tries to find a middle ground between these statistical techniques and AI-based understanding techniques. The idea here is to use a low-level linguistic analysis of the document to identify terms from the document that are probably central to that document's meaning. It is then possible to use these terms to reconstitute how the terms are interrelated and, from there, some of the document's high-level structure. The paper on SCOOP [2] describes this set of language technologies and some sample applications of them; the paper on dynamic document presentation [1] shows how this approach can be used as a central part of an innovative interactive system.
- *Networked communities and distance education*: Finding the content of documents does us little good if we can't communicate that content to the people who need it. Hence, we also

1. Downloadable at http://applescript.apple.com/data_detectors

studied how communities grow up around bodies of information, and what technologies can insure the successful dissemination of that information to the members of that community. One experience with such a community is described in [6], both in terms of the technologies we found to be useful for providing community support, and in how the participants in the community felt about these kinds of interaction and collaboration.

It's important to be clear about the perspective we chose for this work. This was not a rejection of artificial intelligence as a source of useful technologies. It was, however, a fairly explicit rejection of the belief that we could count on the imminent solution of "the AI problem" – that, suddenly, systems with the near-human capabilities envisioned for so long would suddenly become available. But this was not so much pessimism as pragmatism and practicality. We were not working to find great new applications of AI technologies, but to identify problems in peoples' everyday lives and to propose solutions for them. If we chose to let go of the full-bore AI dream, we were perhaps choosing instead the freedom that comes from making use of the technologies and strategies that we have today, and that can be applied to real problems with immediate as well as long-term payoff. And, if the AI problem gets solved next week, there are plenty of ways to fit them into what you'll read about here....

References

1. Boguraev, B. & Bellamy, R. (1988). Dynamic document presentation. *SIGCHI Bulletin*, this volume.
2. Boguraev, B., Kennedy, C., & Brawer, S. (1988). An architecture for content analysis of documents and its use in information and knowledge management tasks. *SIGCHI Bulletin*, this volume.
3. Bonura, T., & Miller, J. R. (1998). Drop Zones: An extension to LiveDoc. *SIGCHI Bulletin*, this volume.
4. Miller, J. R., & Bonura, T (1998). From documents to objects: An overview of LiveDoc. *SIGCHI Bulletin*, this volume.
5. Nardi, B. A., Miller, J. R., & Wright, D. J. (1998). Collaborative, programmable intelligent agents. *Communications of the ACM*, Vol. 41, No. 3 March,1998.
6. Nardi, B. A., Reilly, B., & Steinbeck, R. (1988). An online digital photography course for high school teachers. *SIGCHI Bulletin*, this volume.

Acknowledgments

As ATG came to an end, I was joined in the Intelligent Systems Program by Bran Boguraev, Tom Bonura, Eric Hulteen, Bonnie Nardi, and Dave Wright. Thanks are also due to our colleagues from previous years, including Yu-Ying Chow, Laile Di Silvestro, Dan Russell, and Bob Strong, as well as a number of contractors and interns, including (and with apologies to those I may have forgotten) Ken Anderson, Elizabeth Bratt, Sascha Brawer, Jon Doyel, Bruce Horn, Chris Kennedy, Henry Lieberman, Pattie Maes, Dave McDonald, Max Metral, Justina Ohaeri, Jeremy Sandmel, Shilpa Shulka, Oliver Steele, Heinrich Schwarz, Jason Swartz, Marc Verhagen, Yin Yin Wong, and Dave Yost.

About the Author

Jim Miller, until recently, was the program manager for Intelligent Systems in Apple's Advanced Technology Group. He is currently exploring consumer applications of Internet technology as part of *Miramontes Computing*.

Author's Address

Jim Miller
Miramontes Computing
828 Sladky Avenue
Mountain View CA 94040, USA
email: jmiller@millerclan.com
Tel: +1-650-967-2102

From Documents to Objects

An Overview of LiveDoc

James R. Miller and Thomas Bonura

One of the changes that the World Wide Web has brought to the computing industry is a new way of thinking about documents. Traditionally, documents have been seen as simple streams of characters, like those in a document editor. Applications that manage these documents may do more or less interesting things to the characters, but they rarely attempt to interpret any of the meaning of the document. There's obviously meaning there, but it only becomes apparent when read or otherwise manipulated by a human. In contrast, the Web has brought with it the concept of a document that has been authored in such a way that important bits of information are explicitly identified within the document. This identification exposes some of the meaning of the document, albeit at a fairly low level, so that various kinds of actions – primarily "show me this related document" – are offered to users and made easy for them to carry out.

The gap that separates these two notions of *document* is the need for the human authoring of the Web document. More to the point, it's the need for a human to identify the meaningful components of the document and the actions that make sense for those components. There is a real opportunity to advance the computing field here, by bringing these two worlds together: by enabling an ordinary document, built with any application, to automatically offer users access to some of the meaningful bits of its content, and by helping users carry out appropriate actions on these objects.

Bridging the Gap through Structure Detection

This premise led to a collection of projects within Apple's Advanced Technology Group – within the Intelligent Systems Program, in particular – on the idea of *structure detection*. The work was based on the observation that, while automatically computing a high-level understanding of an arbitrary document is beyond our present ability, many meaningful bits of information are computationally quite easy to recognize: recognizing an e-mail address ("fred@apple.com") or a URL ("http://www.apple.com") takes little more than a context-free grammar, if not merely a regular expression parser. A first step to bridging the document gap described above is then to construct a means of passing text from a user's document into a parser for matching against a collection of recognizers, each of which is looking for some meaningful type of information. These identifications imply simple interpretations of the bits of information that were found: URLs are found by the URL grammar, e-mail addresses are found by the e-mail address grammar, and so on. Then, actions appropriate to each kind of object can be offered, supporting users in their work on those objects and on the document as a whole.

Our overall intent here – to examine document content, identify likely user actions, and provide simple ways of selecting and executing those actions – is not unlike that of the authors of other "intelligent" critic and advisory systems [e.g., 4, 6]. However, our work on structure detection differs from these systems in a number of ways:

- Syntactically-regular information structures, and the tasks that follow from them, can be found in almost any user domain. Hence, the total number of structures and tasks for which structure detection assistance would be helpful is too large for any single person or organization to try to satisfy. Therefore, we have paid special attention to the importance of allowing application developers and even end-users to define and extend the set of detectors and actions. This drove us to design

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.