APPLIED THEORY

SUMMARY

- Explores a variety of technologies, including intelligent agents, information visualization, search engines, and collaborative filtering
- Examines related issues in the intersection of technologies and humans, including adaptation of e-commerce technology and the role of trust

Who is in Control?
The Logic Underlying the Intelligent Technologies Used in Performance Support

WHITNEY QUESENBERY

Neal Stephenson wrote in *Snow crash* that "the human mind can absorb and process an incredible amount of information—if it comes in the right format. The right interface. If you put the right face on it."

So here we are in the information age. Science fiction and cyberpunk writers have been imagining future worlds with deep information structures and an easy familiarity we can only dream of. Characters in the film *Johnny Mnemonic* move through vast data networks as though they are zooming through the stacks of a library. There's even a character called The Librarian in *Snow crash* who performs instant searches on a giant database, returning neatly organized packages of just the right information. But back here in the present day, we're overloaded. Computers have made it possible for us to know more and do more in less time and with less training. And do it with more data at our finger tips. Yet we are overwhelmed because we don't always have the right tools to put the right face on the data and turn it into usable information.

Performance support (also called EPSS, for electronic performance support system) emerged from the instructional design and training communities because corporate enterprise systems were difficult for people to use, and the training needed to make them productive was expensive and time consuming. A good definition is that "EPSS (Electronic Performance Support Systems) are systems that provide employees with the information, advice and learning experiences they need to get up to speed as quickly as possible and with the minimum of support from other people" (Raybould 1996).

One of the issues in designing performance support is managing information overload. Two approaches are the use of agents and the presentation of information in visual form (called *information visualization*). The former looks for ways that computer programs can do work for users, sorting through data on their behalf; the latter looks for ways to present information so that users can directly access it through direct manipulation. You can do both, but the selection of each has an impact on the interaction style and the degree to which users can directly control the system. It is therefore an issue that any performance support system designer should consider carefully. This is a logical extension of the goal of easy-to-use programs, adding the requirement that the user interface be actively informative and helpful.

Performance support systems also typically embody business rules and are designed to enable workers to fulfill their tasks within that context. The rules-based nature of many performance support systems argues in favor of agent technology to narrow down choices and help users make decisions. A successful EPSS also relies on seamless integration into the user interface and an understanding of what information a user needs at any moment. So in addition to understanding the performance to be achieved through the EPSS, the designers of EPSSs also need to

Manuscript received 28 December 2001; revised 27 June 2002; accepted 28 June 2002.

understand the technologies that can assist users in achieving this performance. This article provides an overview of those technologies. First, it explores a variety of technologies, including intelligent agents, information visualization, search engines, and collaborative filtering. Then, this article explores some related technology issues in the intersection of technologies and humans, including adaptation of e-commerce technology to EPSSs, and the roles of trust in automated systems, personalization and privacy, and control.

THE TECHNOLOGIES OF PERFORMANCE SUPPORT

EPSSs can provide more than traditional training and documentation. They can also advise users, coach them, and perform work. To do so, an EPSS employs a variety of complex software technologies. The next several sections provide an overview of those with the most direct application to EPSSs, including intelligent agents, information visualization, search engines, and collaborative filtering. Each section includes a definition of the technology, describes its application to performance support, and identifies theoretical and practical issues with each.

Intelligent agents

What agents are As their name implies, agents are helper programs that can take on tasks for users, acting autonomously on their behalf. Although research and concepts go back to the early days of computing, the first working agents were shown in early 1990s. The researchers at the Software Agents Group at the MIT Media Laboratory envisioned agents that could help users overcome information overload. They define agents somewhat generally as "computer systems to which one can delegate tasks" (MIT Media Lab 2002). A list of the benefits of agents includes several ways in which agents assist users (Maes 1994).

- They perform tasks on the user's behalf.
- They can train or teach the user.
- They help different users collaborate.
- They monitor events and procedures.

For an agent to be successful, it must follow a clear set of rules and must be able to communicate with the target of its work, such as database, news service, or other agents. Much of the research in agents has focused on how the agents themselves work to follow those rules. Only recently has the importance of the human interface emerged as a critical factor in the success or failure of an agent.

What's the difference between an agent and a smart program? The general answer is that an agent goes beyond a simple program to

1. Function independently
2. Exist over time

Although there is no single definition of an agent (or even consensus on the required characteristics), some of the properties most often attributed to an agent are shown in Table 1.

Much of the research in agents has focused on how the agents themselves work to follow those rules.

What agents are not Most programs fail to meet the criteria for being agents because they are not temporally continuous—that is, they do not persist over time—and because their output does not affect what it will do later. But the lines are blurry. "A spell checker adjunct to a word processor is typically not an agent [for the reasons given in the preceding paragraph]. However, a spell checker that watched as the user typed and corrected on the fly might well be an agent" (Franklin and Graesser 1996). This statement suggests that many of the "modern" improvements in software are in some respect agents that make these programs more helpful to users. In another early example, Maes cites a mail sorting agent as an example of the kinds of tasks agents might perform (Maes 1994). Tools that move messages to designated mailboxes based on the sender, title, or other characteristics are now generally available in most e-mail programs. However we don't think of them as agents but simply as a software function or filter. They are also typically constrained to operating within the e-mail program and are not really autonomous.

A wizard—a program that guides a user through a complex task—is a little closer to meeting the criteria for an agent, but it also fails to meet those criteria fully. Wizards are neither adaptive nor continuous, but are typically designed as a more supportive interface for a specific task. They embody logic and may be highly interactive, but they do not function independently. A wizard-like format, however, might make a good interface to provide an agent with its instructions.

This ability for the user to set the rules for the agent is a critical distinction between an agent and a program with embedded rules. Most definitions of agents include the ability of an agent to learn over time and adapt its behavior in response to events it observes or to a changing environment. This brings us to the issue of human control.

Human control and information visualization

What information visualization is Ben Shneiderman, founder of the University of Maryland Human-Computer Interaction Laboratory, has been a vocal champion of direct human control of computers. His research has focused on

TABLE 1: PROPERTIES ATTRIBUTED TO AGENTS (ADAPTED FROM FRANKLIN AND GRAESSER 1996)

Property	Meaning
Reactive	Responds in a timely fashion to changes in the environment
Autonomous	Exercises control over its own actions
Goal-oriented	Does not simply act in response to the environment
Temporally continuous	Is a continuously running process
Communicative	Socially able to communicate with other agents, perhaps including people
Learning, adaptive	Changes its behavior based on its previous experience
Mobile	Able to transport itself from one machine to another
Flexible	Is not scripted
Character	Demonstrates a believable "personality" and emotional state

discovering better ways to display information so people can make choices for themselves rather than relying on a hidden algorithm to do so. Much of this work has focused on techniques to make a large collection of data visible and understandable in a small display space. This work is called information visualization. An example of information visualization is the MarketMap, which displays a summary of the stock market performance in a single screen (http://www.smartmoney.com/marketmap/).

In advocating for information visualization and other aspects of usable interfaces, Shneiderman also promotes the principle of "direct manipulation" to allow human control of the program. Shneiderman's criteria for direct manipulation in a user interface are

- That the object of interest is continuously represented
- That the user interacts through physical actions (or button presses) rather than through a complex command syntax
- That the interaction is characterized by a series of rapid incremental and reversible operations whose impact is immediately visible

These criteria allow for systems that "give us the qualitative feeling that we are directly engaged with control of the objects—not with the programs, not with the computer, but with the semantic objects of our goals and intentions" (Shneiderman 1987).

The controversy surrounding human control and agents In 1996 and 1997, Maes and Shneiderman held a series of debates, reported in an article in the magazine *Interactions* (Shneiderman and Maes 1997), which brought the comparison between the two approaches into sharp relief. The most important difference was their underlying philosophy. Shneiderman is unwilling to delegate critical interpretive tasks to computers, reserving them for people. He also took issue with the tendency to anthropomorphize agents, saying "I make the basic assertion that people are not machines and machines are not people. I do not think that human-to-human interaction is a good model for the design of the user interface." Maes, speaking in support of agents, suggested that with the vast amount of fluid information we manage today, "we need to be able to delegate to what could be thought of as 'extra eyes or extra ears' that are on the lookout for things that you may be interested in."

Their goals were similar in many ways, even if their approaches to a solution differed. Both, for example, were looking for ways to expose unexpected or serendipitous relationships among bits of information, whether they were found through a person examining a visual presentation or by an agent working behind the scenes and presenting what it found to the user. They also agreed that some kinds of low-level tasks can and should be automated. Shneiderman would limit it to management of details "under the hood . . . as long as it does not interfere with the user's

prediction of behavior"; Maes was predictably more willing to allow agents to automate complex interactive tasks and to accept the idea that she would not necessarily understand all the details of its work. She compared this attitude to her willingness to forego learning auto repair, and to trust her mechanic to do this work on her behalf.

This debate is a tangled one and depends in part on the definition of *agency*. Consider two features of the modern automobile: the outside temperature sensor and anti-lock brakes. The temperature sensor monitors the temperature around the car and triggers an alarm at a preset temperature when dangerous road conditions may be present. This sensor is not an agent because it takes no action, makes no decisions, but simply notifies the driver when its criteria are met.

The anti-lock brakes, on the other hand, not only monitor conditions, but make decisions about the action to take for the most effective way to safely stop the car. Anti-lock brakes can be thought of as a simple agent. They are autonomous (exercising control over their own actions), reactive (responding to changes in the environment), and communicative, but not goal-oriented (in the sense of not simply responding to the environment) or adaptive. Most importantly for the proponents of direct manipulation, they act in response to a human action, and provide instant feedback on their own actions. The user interface is a simple one—direct manipulation of the brake pedal—even though the agent's actions are complex.

One key area of agreement in the Maes-Shneiderman debates (1997) was the importance of the user interface—"user understanding is central, and user control is vital." We will return to this issue later, but first it is worth looking "under the covers" at some of the techniques that both direct manipulation interfaces and agents use to manage and filter the large information spaces that they handle.

Search engines

What search agents are In some ways, the search engines that are so pervasive on the Web can be seen as a kind of agent. One popular search site even emphasizes the agent-like nature in its ads, mascot (a retriever), and brand line: "Go Get It" (http://www.lycos.com). This tag plays on the comfortable metaphor of a dog as an agent for humans, suggesting that the software is as trustworthy and well trained as a good hunting dog.

The weakness of a pure text retrieval system is its lack of semantic interpretation. Even natural language interfaces such as AskJeeves (http://www.askjeeves.com) are limited in their ability to interpret colloquial language. While "human language thrives when using the same term to mean somewhat different things," computers do not (Berners-Lee and others 2001). This limitation makes it difficult to use search agents in situations where judgment or interpretation is needed. The "problem" with the Web is that the information on it is for people rather than data for machines—that is, the information is not in a structured format that can be processed automatically. For a search engine or performance support system to reach outside of a single system, it needs to be able to communicate with other systems—that is, it needs a common language for data interchange.

The weakness of a pure text retrieval system is its lack of semantic interpretation.

One of these projects of the World Wide Web Consortium (W3C) is the Resource Description Framework (RDF), a "lightweight ontology system to support the exchange of knowledge on the Web" (W3C 2002). RDF proposes a system of metadata to make the information on the Web "machine understandable" and is part of a larger project called the Semantic Web. This project was introduced to the general public in an article in *Scientific American* (Berners-Lee and others 2001). "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." The authors describe a grand vision that "the real power of the Semantic Web will be realized when people create many programs that collect web content from diverse sources, process the information and exchange the results with other programs."

How search engines work Search engines supply many of the benefits identified with agents, but they are not autonomous, unless you consider the company that maintains the search index or classification as part of the agent/system. Google (http://www.google.com), for example, has a 1.3 billion word index that is refreshed every 28 days. This index enables Google to search for specific terms quickly. Most search engines use some kind of index—searching would be a physical impossibility without it—but each one also attempts to add value in how it prioritizes the pages that it finds to put the most meaningful at the top of the list.

According to Google's director of research, Monika Henzinger, ". . . we don't only return those documents, we also return documents where other people point—have a hyperlink—to this page" (Mieszkowski 2001). In establishing relevance, Google relies on the popularity of a site (based in part on the number of links *to* its pages) as an indication of its importance, mixing a recommendation scheme with pure information retrieval.

Collaborative filtering

What collaborative filtering is Another approach to using community recommendation to establish relevance

and priority is collaborative filtering. Some of the early research on this technique was conducted at Bellcore. One of the conclusions drawn from this research was that to make a selection (for example of a movie), "choosing as other like-minded, similarly-situated people have successfully chosen in the past is a good strategy—in effect using other people as filters and guides" (Hill and others 1995).

The advantage to collaborative filtering systems is that they need to know very little about the content of the database. It is enough for them to look for patterns of preferences. Agent-based recommendation systems, on the other hand, need accurate metadata about the items in the database and rules about the relative weight of each piece of information to be effective. The solution proposed by collaborative filtering is to compare each person's ratings of items with those of hundreds of others, on the assumption that the greater the overlap in personal ratings, the more valid the recommendations for nonrated items would be.

The Bellcore project used the Internet to collect movie ratings from October 1993 through May 1994 and used this data to recommend movies to participants. For most users, the results were very positive, though there were some who did not agree with the movie selections presented to them. The Bellcore researchers believe that collaborative filtering "outperforms by far a standard source of movie recommendations: nationally recognized movie critics" (Hill and others 1995).

The growing use of collaborative filtering The use of collaborative filtering has grown and is still ongoing in such projects as the GroupLens Research project at the University of Minnesota (GroupLens Research 2002). Its MovieLens is still online and offers a data set with over 100,000 ratings. One of this project's innovations is the transparency that results from adding explanations of the reasoning and data behind a recommendation.

> *In the process of deciding to accept a recommendation from a friend, we might consider the quality of previous recommendation . . . However, if there is any doubt, we will ask "why?" and let the friend explain their reasoning behind a suggestions. Then we can analyze the logic of the suggestion and determine for ourselves if the evidence is strong enough. (Herlocker and others 2000)*

The GroupLens Research project also points out that it is important for users to be able to look into the "black box" if recommendation systems are to be trusted for domains where there is higher risk than spending two hours watching a movie one dislikes.

In contrast to collaborative filtering, most other selection and recommendation programs require detailed metadata about each of the items in the database. One is the University of Maryland Human-Computer Interaction Lab's Film Finder (Shneiderman 1999). In this program, users looking for a film enter the criteria for a query by adjusting sliders, buttons, and check boxes. The results are displayed in a real-time scatter-plot display (called the "starfield") containing a small iconic point for each matching film. With a click on any of the points, details about the film are displayed in a small popup window.

Beyond the interface issues in completing a transaction successfully, e-commerce sites share some of the system issues with both performance support systems and agents.

This project was one of a series of experiments with a visualization concept called "dynamic queries" that used direct manipulation techniques in a search interface. In a typical search interface, the user enters criteria for the items to be found and "runs" the search. Then the search program displays items matching those criteria for selection specified by the user. Dynamic queries replace this two-step interaction, with a continuously visible display showing both the search criteria and a visualization of the currently matched items. Changes in the criteria and results are instantly displayed, thus encouraging the exploration of even a complex data set. Similar techniques have been developed by others, and this approach is increasingly popular in programs handling complex information. The downside of this approach is that detailed information about each item in the database must be entered in advance, making it most useful for structured information.

Collaborative filtering has also moved into the mainstream of e-commerce. Amazon (http://www.amazon.com), for example, offers two examples. Personal recommendations allow users to rate books in various categories, and receive recommendations of other, similar books. Amazon also includes a recommendation feature for each of their books that offers a list of three to four books also bought by people who bought the current one.

ISSUES IN USING THESE TECHNOLOGIES

Although these technologies offer a number of benefits, they pose a variety of issues, some obvious, some not. The next several sections explore some of the key issues, including adaptation of e-commerce technology to EPSSs, and the roles of trust in automated systems, personalization and privacy, and control.

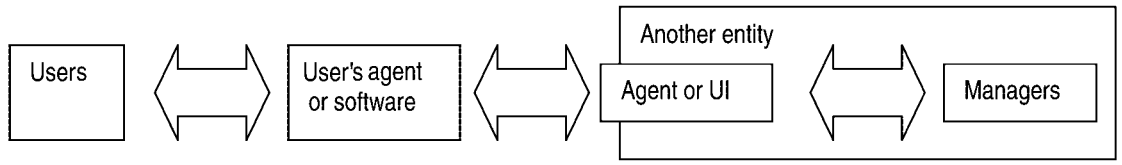

Figure 1. Trust relationships.

Adaptation of e-commerce technology to EPSSs

Although e-commerce occupies its own niche, it is relevant to performance support systems because one of the changes it is bringing to commerce (especially to business-to-business purchasing) is enabling customers to perform tasks previously handled by human sales agents. Besides the simple examples of placing orders automatically, there are other services that allow agents to manage insurance applications through a Web site, or professors to construct a custom textbook and make it available for purchase by students on services such as McGraw-Hill's Primis (http://www.mhhe.com/primis/).

Many of these e-commerce or extranet capabilities put powerful enterprise systems in the hands of untrained users. In doing so, the burden of clear communication of the rules and options has shifted from training for employee-operators to the user interface and, in the process, has increased the usability requirements for these increasingly complex tasks.

Beyond the interface issues in completing a transaction successfully, e-commerce sites share some of the system issues with both performance support systems and agents. All three, for example, rely on access to a knowledge base of rules and other information. What makes it possible for some e-commerce systems to replace a sales-person is that the software is given more responsibility for the tasks it assists with, and therefore requires a closer relationship between the humans and the software tools. It also means that the users must trust the software in the same way they trusted (or perhaps, in the same ways that they did not trust) human sales agents.

Trust

The more control people cede to others, the more they have to trust them. In enterprise software, a relationship between the software owner and the employee using it already exists: these tools are provided by the company as a means for employees to do their work. Issues of trust and coercion are part of the larger employee-employer relationship, rather than established solely through the software. In this case, users must trust that the software will work properly, and help them do their jobs well and provide adequate support for those tasks, but they do not typically need to worry about whether it will work in ways that are contrary to their own interests. They may, however, not feel trusted by their employer if they are not sufficiently empowered to do their work by overly restrictive software or inadequate performance support.

In contrast, public agents and e-commerce tools must earn trust from each individual. In the e-commerce context, a usability research study from Nielsen/Norman Group defined trust as "the person's willingness to invest time, money and personal data in an e-commerce site in return for goods and services that meet certain expectations" (Nielsen and others 2000). One way that companies try to establish expectations that they can be trusted is through their brands. Brand is much more than a company's logo; it is the consumer's mental model of their relationship with a company. This model is built up through many media: advertising, word-of-mouth, and direct experience. But at the moment of use, the "user interface conveys the brand, accurately or not, and establishes the bounds of a relationship" (Baron 2001). This relationship is fragile. As the Nielsen report puts it, "trust is hard to build—and easy to lose. A single violation of trust can destroy years of slowly accumulated credibility" (Nielsen and others 2000). What's important here is that the usability and interaction style of any interface, including agents and performance support systems, are critical factors in its success or failure.

The intelligent software agents discussed earlier face an even harder battle to create trust because the details of their actions may be hidden from the user. A user must not only trust the agent itself but all the other entities with which the agent interacts, as shown in Figure 1.

As the number of entities the agent must interact with becomes larger, the trust equation becomes more complex. A break-down in any of the transactions can spoil the entire chain and make the results questionable, if not untrustworthy. For example, people may trust their online travel service to show the best prices it can find, but do people know whether the information they get from the airlines actually reflects the best prices available? How well would people trust the service if they learned that it did not have access to special sale prices at all airlines, or if it received a payment for suggesting routes through a specific city?

This problem is not confined to the online world. There are many kinds of live service agents, from hotel

concierges to real estate purchase agents. These agents may be used for their expertise or knowledge base, often in an area where the customer has limited knowledge or experience. A critical factor in establishing trust is how transparent the agent's action is—that is, how easily the user can see inside the transaction or independently validate the results. This transparency is just as important in establishing trust with a software agent as with a human vendor.

Personalization and privacy

Agents and other systems need to know not only how to communicate with each other, but also about the preferences and personal characteristics of the person they "work for." The question of how this information will be used once collected has raised privacy concerns that must also be considered as part of a design process.

Concern over privacy runs deep. In one study, 65% of consumers were "very concerned" or "extremely concerned" about privacy online (Berinato 2000). This concern suggests a general lack of trust that is translated into measurable actions. For example, among those who have sought pharmaceutical information online, 47% have not submitted any personal health information to a site, 42% have not registered to gain access to more tailored information, and 30% have not accessed health information online at all due to privacy concerns (CyberDialog 2000).

The legal and ethical policies that underlie use of information technology are far from mature. Laws on core issues such as privacy, free speech and copyright are still being tested in courts both within the U.S. and around the world. The Internet has created porous boundaries, making it difficult to determine which country's laws apply in situations that cross national borders. The very electronic nature of information creates opportunities for error. Where paper-based information requires effort to assemble and use it, electronic information can be easily misused, especially in situations where the software does not offer good human control.

Business norms for use of personal information are still being formed. In at least one dot.com bankruptcy case, the company ended up in court over its attempt to sell as an asset customer data collected under a privacy agreement (Enos and Blakey 2000).

In addition to the problems of establishing ethical behavior for those who collect or access information, it is also essential that every person understand what personal data they are surrendering and the conditions under which it will be used. The W3C Privacy Preferences Project is attempting to address this problem by creating a standard privacy document that can be translated into a machine-readable rule to assist the user in making privacy-related decisions (XNS Public Trust Organization 2000). There are also business groups looking at ways to more effectively structure and communicate privacy guidelines to consumers. One coalition is backing a proposal to create a plain English summary of privacy policies that will make it easier to compare them (France 2002).

While we are being pessimistic about the ability to control the use of information, here's a really scary notion: the ultimate agent is a virus.

Some have pointed out that "privacy" is largely an illusion, that personal information has always been collected (consider the kinds of primary source records that historians sift through to build a picture of an era). The difference is that all this information has never been so easily accessed nor as available to so many people. The line between "helpful" and "intrusive" is a difficult one to define, and its location often depends on social context. Privacy may be an important intersection between agents and performance support systems: the amount of personal information that is revealed by one person—or made available to another—may define that boundary.

Control

The vision of agents as a solution to information overload is based on a lot of optimistic assumptions. The most important of these is that we will be able to trust and control the systems we create.

In an article called "When 'smart' refrigerators go bad" (Plotnikoff 1999), the author imagines, "A worst case scenario? When those home devices get tired of talking to the grocery store and start building a meaningful online dialogue with my insurance carriers." He goes on to envision a conversation in which his refrigerator refuses to release a delectable leftover because his cholesterol is too high. Another popular columnist, John Dvorak, suggests that allowing appliances (hardware agents?) to call for repairs is an open invitation to crime (Dvorak 2000). Information that is free for one is free for all.

While we are being pessimistic about the ability to control the use of information, here's a really scary notion: the ultimate agent is a virus. Franklin and Graesser even include viruses in their taxonomy of agents. One of the first viruses to emerge into public consciousness was Robert Morris's worm in 1988. It wasn't even intended as a virus. It was written as a worm—a self-replicating, self-propagating program. Unfortunately, it contained a bug and began

replicating and infecting machines faster than he had intended. The damage was severe enough that it crashed machines all over the country before his messages on how to stop it could be distributed ("Summary of the Robert Morris case" n.d.).

New technology and features may require trading some privacy and control for the convenience of delegating some kinds of work to agents. But whatever approach or technology is used, it is important that people can use it with "a feeling of mastery," through an interface that is "controllable, consistent and predictable," and an interaction style that leaves the user in control (Ben Shneiderman, quoted in Beardley 1999). For humans to maintain control over an agent, we must have the ability

- To understand and validate the rules that govern the agent
- To verify that the agent has operated according to those rules
- To discover any other forces or rules at work on the agent

Without these abilities, we are the sorcerer's apprentice, setting loose systems without the means to control them or determine their results.

TECHNOLOGY AND THE IMPORTANCE OF THE USER EXPERIENCE

Performance support starts with the goal of assisting humans in managing tasks, especially those that involve managing large amounts of information or complex rules. One design solution is to delegate control to intelligent software agents, using rules and semantic data to instruct them. Another is to find ways to present information and options in a way that leaves the human user in control, such as through information visualization, collaborative filtering, and search.

Whatever choices the designer makes, the EPSS must allow users to see not only the choices they are making, but the implications of those choices. Agents must be designed so that their actions are visible and can be examined—or changed. Designers of performance support systems will need to balance options to make selections that enhance the user experience and provide the sense of mastery and control that is part of the EPSS "brand promise." T**C**

REFERENCES

Baron, M. 2001. Private communication to Whitney Quesenbery.

Beardlesy, T. 1999. "Humans unite!" *Scientific American* (March). http://www.sciam.com/1999/0399issue/0399profile.html

Berinato, S. 2000. "Forrester exec injects security summit with harsh truths." *eWeek* (7 December). http://www.zdnet.com/filters/printerfriendly/0,6061,2595089–2,00.html

Berners-Lee, T., J. Hendler, and O. Lassila. 2001. "The semantic Web." *Scientific American* (May). http://www.sciam.com/2001/0501issue/0501berners-lee.html

CyberDialog (now Fulcrum Analytics). 2000. "Cybercitizen health 2000."

Dvorak, J. 2000. "Smart homes, dumb ideas." *PC magazine* (27 June). http://www.zdnet.com/filters/printerfriendly/0,6061,2595089-2,00.html

Enos, L., and E. Blakey. 2000. "Toysmart will not sell customer names . . . for now." *E-commerce times* (28 July). http://www.ecommercetimes.com/perl/printer/3894/

France, M. 2002. "Privacy: The long and the short of it." *Business week* (24 June):12.

Franklin, S., and A. Graesser. 1996. "Is it an agent, or just a program?: A taxonomy for autonomous agents." In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages.* http://www.msci.memphis.edu/~franklin/AgentProg.html

GroupLens Research. 2002. http://www.cs.umn.edu/Research/GroupLens/

Herlocker, J., J. Konstan, and J. Riedl. 2000. "Explaining collaborative filtering recommendations." In *ACM 2000 Conference on Computer Supported Cooperative Work.* http://www.cs.umn.edu/Research/GroupLens/explain-CSCW.pdf

Hill, W., L. Stead, M. Rosenstein, and G. Furnas. 1995. "Recommending and evaluating choices in a virtual community of use." In *Proceedings of CHI'95.* http://www.acm.org/sigchi/chi95/proceedings/papers/wch_bdy.htm

Maes, P., 1994. "Agents that reduce work and information overload." *Communications of the ACM* 37, no 7. http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html

MIT Media Lab. 2002. "Software Agents Group." http://agents.www.media.mit.edu/groups/agents/

Nielsen, J., R. Molich, C. Snyder, and S. Farrell. 2000. "E-commerce user experience: trust." http://www.nngroup.com/reports/ecommerce/

Mieszkowski, K. 2001. "Google a go-go." *Salon.* http://www.salon.com/tech/feature/2001/06/21/google_henziger/print.html

Plotnikoff, D. 1999. "What if your fridge talked back to you?" *Contra Costa times* (14 March): D7.

Raybould, B. 1996. "What is an electronic performance support system." *EPSS. com.* http://www.epss.com/lb/artonlin/articles/EPSSdef.htm

Shneiderman, B. 1987. *Designing the user interface.* Reading, MA: Addison-Wesley.

Shneiderman, B. 1999. "Dynamic queries, starfield displays, and the path to Spotfire." http://www.cs.umd.edu/hcil/spotfire/

Shneiderman, B., and P. Maes. 1997. "Debate: Direct manipulation vs. interface agents." *Interactions: New visions of human-computer interaction* 4, no. 6:42.
"Summary of the Robert Morris case." n.d. *Computer abuse: Worms, viruses and Trojan horses.* http://www.eos.ncsu.edu/eos/info/computer_ethics/abuse/wvt/worm/

Stephenson, Neal. *Snow crash*. New York, NY: Bantam Spectra.

XNS Public Trust Organization. 2001. "XNS and Internet privacy." http://www.xns.org/xns/whitepapers/privacysummary.html

W3C Semantic Web Project. 2002. "Resource Description Format." http://www.w3.org/RDF/

WHITNEY QUESENBERY designs interfaces for projects from Web sites to applications and helps companies improve the usability of their products. She has long been concerned with the relationship of people and computers. As a principal at Princeton, NJ-based Cognetics, her projects ranged from the original HP LaserJet 4 Travel Guide to hospital management software, Web applications, and corporate information tools. For STC, Whitney has been the manager of the Usability SIG and is the Webmaster for its popular Web site (www.stcsig.org/usability). She is also the director of strategic outreach for the Usability Professionals' Association. Contact information: whitneyq@acm.org