BABY STEPS

They can't fly yet, but intelligent agents and smart software are beginning to walk. Here's how they can make you work smarter.

KURT INDERMAUR

ROB SCHUSTER © 1995

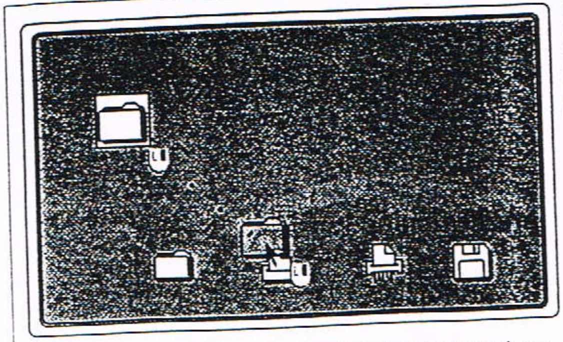
Think the idea of intelligent software agents is being oversold? Coach may convince you otherwise. Developed by Dr. Ted Selker at IBM's Almaden Research Center in California, this teaching assistant holds programming students by the hand as they learn the intricacies of Lisp. In a test using IBM programmers with no Lisp experience, student/Coach teams completed five times more training exercises involving database function calls than did students who didn't have the aid of an electronic tutor.

At the end of the training sessions, most Coach pupils said they liked Lisp; their untutored counterparts weren't so enthusiastic. Skeptics may scoff at agent hype—and the dearth of commercial programs—but if an agent can actually make using Lisp an enjoyable experience, maybe anything is possible.

Assisting in education and training has been a worthwhile aspect of agent research over the past decade. But the agents being developed and sold today cross a number of application boundaries. They're designed to filter and gather information from commercial data services and public domains like the Internet and to automate work flow. Because the range is broad, it helps to separate agents into three main categories: advisory, assistant, and Internet.

Advisory Agents

Advisory agents don't actually carry out tasks; like Coach, they offer instruction and advice to help you do your work. These agents are experts in a particular domain, but in the beginning, they have only rudimentary knowledge of you, your work patterns, and your preferences for managing your work life. As you go about your work, these agents learn such things about you as your level of expertise, your programming style, or your areas of personal and professional interest. Then, either



Coach uses AI techniques to help beginning programmers learn Lisp. Here, the icons along the bottom represent the "Big Picture": the main steps a programmer must follow to write database function calls. The dots trailing the file folder icon depict an animated image, which shows the user the actions he or she must take to move from one step to the next. This graphical shorthand keeps the agent small and fast and allows it to give advice unobtrusively.

when you ask, or when the agent decides it's time to chime in, it offers advice or information. The longer you work with an advisory agent, the better and more timely its advice becomes. At its best, such an agent could anticipate your goals and offer to complete a task the way you've done it in the past.

Selker, who is manager of USER (User Systems Ergonomics Research) at the IBM center, has been evolving Coach along these lines over the past decade. Coach, which is being converted from Lisp to C++, contains three independent but related sets of knowledge to do its work. The first is its user model, where the program builds and maintains information about a user's Lisp-programming proficiency, his or her mistakes (and the chosen fixes), and what worked and what didn't work in terms of coaching. Coach also tracks the rate of change in each user's proficiency (and whether it's getting better or worse), and for each learnable task, how long it's been since it was last used. If a user has done something correctly in the past and has trouble with it now, Coach can present an example from the user's own code.

The user model is persistent—it remains associated with a user from session to session. In a teaching environment, the user model could be made available to a teacher to help him or her understand just where a student might be having difficulties. In a commercial implementation, the coach agent might pass the user model onto a human customer-service representative if the agent determines a customer's problems are beyond the agent's capabilities.

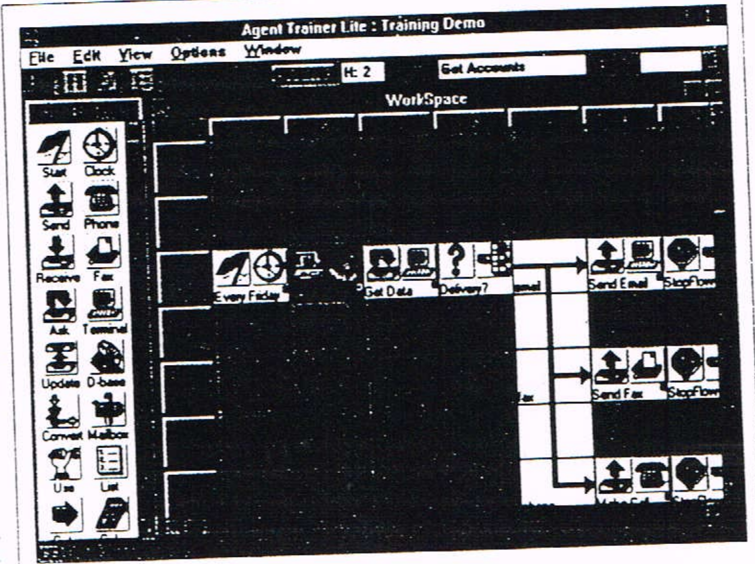
Second, Coach has knowledge about its subject matter. In the case of Lisp programming, Coach knows Lisp syntax, library functions, and concepts (e.g., evaluation, iteration, and recursion). This knowledge base grows over time, automatically incorporating user-defined functions into its repertoire. Such a facility would be a welcome addition to any programming environment, particularly in a team setting in which you're writing code for and using code from other programmers. Your agent might suggest using an existing function or object before you reinvent one yourself. Keeping the domain knowledge separate from knowledge about coaching has made it easy to apply Coach's framework to new domains. For example, Selker says Coach helped a summer intern inexperienced in programming create a Unix help system in just 10 weeks.

The third knowledge set consists of coaching rules that tie user knowledge and the domain together. These rules help Coach gauge a user's level of experience. Update rules determine when the program should refresh the user model to indicate that he or she has mastered a problem or when it's appropriate to present more advanced usage options for a particular feature. Consistency rules make sure that the user model doesn't contradict itself when the model is applied to related subjects. Finally, presentation rules determine how help will be presented to the user. For example, a user who's just starting out would want basic information, while an out-of-practice user may only need a reminder.

Assistant Agents

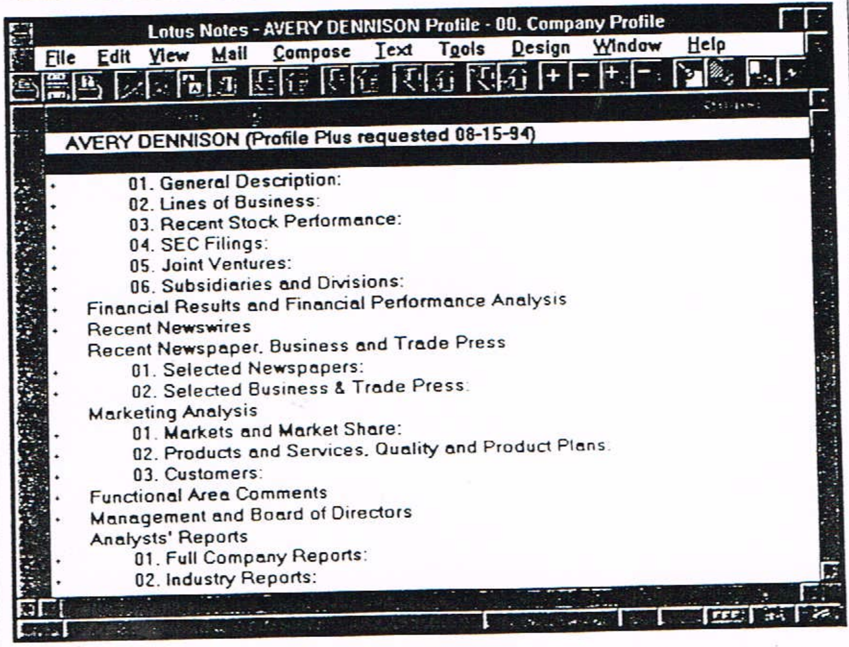
Assistant agents can be more ambitious than advisory agents because they often act without direct feedback from users. While this allows them to be much more powerful, it also raises a host of technical and even social issues that have yet to be resolved. The concerns you might have over privacy and stifled creativity with an agent that is only offering advice become much more acute when your agent is actually doing work for you.

Current commercial agents that assist users have more or less avoided these issues because they're designed for specific domains. Smart mailboxes (see "Smart Networks," October 1994 BYTE) and search engines are two well-known examples of these applications. In addition, Edify (Santa Clara, CA) offers products for human-resource and voice-response systems. Edify's Electronic Workforce is an agent-supporting environment with a development platform and applications. The electronic workforce has three components: the agents, a run-time environment, and a visual programming tool that's used to train agents. The product incorporates agents to automate human-resources queries and customer-service telephone support and handle the behind-the-scenes paperwork of an employee review



In the agent-trainer facility in Edify, you click on any of the graphical images to fill in details. For example, behind the clock image is a scheduling calendar, which in this case is set up to schedule an event every Friday.

process. Charles Jolissaint, Edify's chief technology officer, says that agents allow companies to augment traditional support applications with automated, 24-hour-a-day services. For example, banking customers can request that they be notified immediately (via fax, telephone, or pager) if their account balance drops below a certain amount, or buyers can be faxed a copy of the shipping manifest at the moment their order is shipped.



AVERY DENNISON (Profile Plus requested 08-15-94)

- 01. General Description:
- 02. Lines of Business:
- 03. Recent Stock Performance:
- 04. SEC Filings:
- 05. Joint Ventures:
- 06. Subsidiaries and Divisions:
- Financial Results and Financial Performance Analysis
- Recent Newswires

Recent Newspaper, Business and Trade Press

- 01. Selected Newspapers:
- 02. Selected Business & Trade Press:

Marketing Analysis

- 01. Markets and Market Share:
- 02. Products and Services, Quality and Product Plans
- 03. Customers:
- Functional Area Comments
- Management and Board of Directors

Analysts' Reports

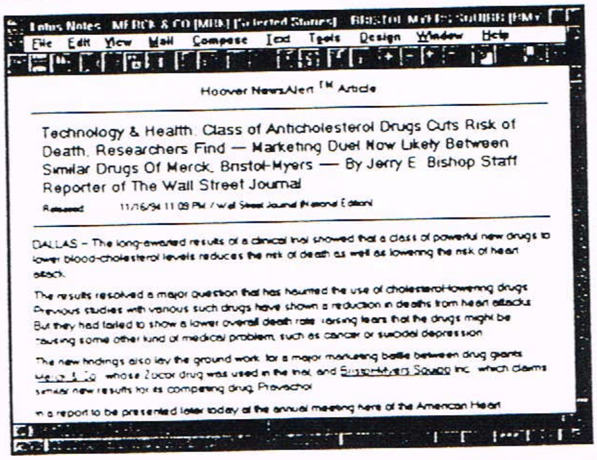
- 01. Full Company Reports:
- 02. Industry Reports:

Hoover can find and report back information from Notes databases and commercial data services. This screen shows a custom company-profile template, which the program can fill in for a one-time ad hoc query or automatically on a regular basis if the subject is a company you track over time.

Edify's Workforce agents don't start life with specialized knowledge. Developers walk an agent through its paces to train, debug, and deploy it in the agent run-time environment. Because the program uses a visual programming environment, developers never have to drop down to an underlying scripting language. The database backend can be any database that supports OS/2 clients, and the run-time environment itself is an applications server that can be distributed over multiple OS/2 nodes.

The run-time environment includes a resource manager and a scheduler, which both optimize and prioritize the system's use of resources. The high level of system support and the capabilities of the operating environment mean that developers can focus on developing their own agents rather than worrying about contention for telephone lines or other resources. Operating-system support in the Electronic Workforce run-time environment makes agent development easier and, perhaps more important, reduces the risks of running an ill-behaved agent.

Another example of an assistant agent is a mailbox that can manage all your electronic communications, including telephone, fax, E-mail, and pagers. From the first day you use it, the mailbox would convert one message format to another, depending on the device you use to access your mailbox (e.g., convert voice mail or a WordPerfect document into plain text for your pager). Then, rather than (or in addition to) asking for explicit rules for handling your mail, your mailbox agent would observe how you process your messages and offer to set up rules of its own. Junk mail might automatically be routed to a low-priority folder, urgent messages could be forwarded to you at home, and your electronic inbox could be prioritized based on the source and content of the messages you receive.


Lotus Notes - MERCK & CO [MRK] [Selected Stories] - BRISTOL MYERS SQUIBB [BMY
File
Edit
View
Mail
Compose
Text
Tools
Design
Window
Help

Hoover NewsAlert TM Article

Technology & Health: Class of Anticholesterol Drugs Cuts Risk of Death, Researchers Find — Marketing Duel Now Likely Between Similar Drugs Of Merck, Bristol-Myers — By Jerry E. Bishop Staff Reporter of The Wall Street Journal

Released 11/16/94 11:09 PM / Wall Street Journal (National Edition)

DALLAS – The long-awaited results of a clinical trial showed that a class of powerful new drugs to lower blood-cholesterol levels reduces the risk of death as well as lowering the risk of heart attack.

The results resolved a major question that has haunted the use of cholesterol-lowering drugs. Previous studies with various such drugs have shown a reduction in deaths from heart attacks. But they had failed to show a lower overall death rate, raising fears that the drugs might be causing some other kind of medical problem, such as cancer or suicidal depression.

The new findings also lay the ground work for a major marketing battle between drug giants Merck & Co., whose Zocor drug was used in the trial, and Bristol-Myers Squibb Inc., which claims similar new results for its competing drug, Pravachol.

In a report to be presented later today at the annual meeting here of the American Heart

Hoover can also locate and retrieve news stories with topics of interest to individual users. Hoover's developers say that future refinements in the program will center on making the interface easier to use.

In the process of prototyping just such a mail agent, Pattie Maes, assistant professor and an agent researcher with MIT's Media Laboratory (Cambridge, MA) identified two important factors in the design of a good agent: competence and trust. To be competent, the agent must determine what tasks to do, when to do them, and the best way to perform the jobs. Maes observed that an agent that asks too many questions, interrupts too frequently, or does more than the user wants puts users off. To be trusted, an agent's actions shouldn't surprise the user. An agent should ask before acting if the task at hand is new or if an action might bring about unanticipated results.

While it's relatively easy to describe competence and trust, it's much more difficult to develop an agent application with those characteristics, Maes notes. These characteristics must be molded for each application, which requires extensive user prototyping to get the right balance between independence and intrusiveness.

Henry Lieberman, also with the Media Laboratory, recommends that agent programmers always give people control over their agents' intrusiveness. Some people want to know what their agents are doing all the time and to approve every action before it's begun. Other people are content to let their agents work without direct control. As people gain confidence in their agents' competence, they will likely want to change the degree of control they exercise over their agents.

Assistant agents can make users more productive by reducing mundane tasks; by nature, they can be

WWW AGENTS

Martijn Koster, of Nexor, a communications software company in the U.K., maintains a list of WWW (World Wide Web) agents (http://web.nexor.co.uk/mak/doc/robots/robots.html). This Web page also includes guidelines for agent developers and links to some of David Eichmann's work (see the main text). Most of the indexing agents make their indexes publicly available on the WWW. The list includes the following:

- **JumpStation/JumpStation II Robot**
  Constructs an index of documents by title, header, and subject. Author: Jonathon Fletcher (J.Fletcher@stirling.ac.uk).
- **Lycos**
  Research aid uses a finite memory model of the WWW to guide directed searches. Author: Dr. Michael L. Mauldin (fuzzy@cmu.edu) at Carnegie Mellon University.
- **NorthStar Robot**
  Another indexing/searching agent for the WWW. Authors: Fred Barrie (barrie@unr.edu) and Billy Brown.
- **Repository-Based Software Engineering Project Spider**
  A combination agent and indexer; traverses the WWW and indexes the full text of what it finds. Author: Dr. David Eichmann (eichmann@rbse.jsc.nasa.gov).
- **WebCrawler**
  Creates a content-based index of documents it finds on the WWW and satisfies specific user search requests. Author: Brian Pinkerton (bp@biotech.washington.edu).
- **W4 (World Wide Web Wanderer)**
  Measures the growth of the WWW. Author: Matthew Gray (mkgray@mit.edu).

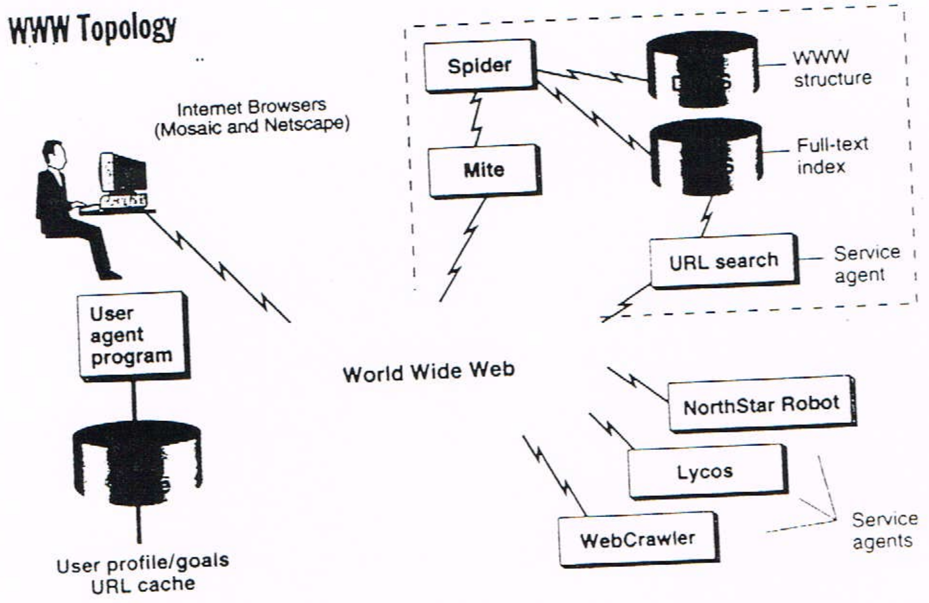
less intrusive than advisory agents. However, assistant agents' independence makes them more dangerous. We have less control over assistant agents but may still be responsible for their actions. As a result, issues of security, liability, and payment for services become important. Your agents may incur charges or do something you might have implied but did not explicitly approve, and you could be held accountable for those actions. As rapidly growing networks create more opportunities for agent software, they also increase the cost of an ill-behaved agent (for details about how agent-based operating environments are addressing security, see "Free Agents" on page 105). Users and service providers will want to be certain that agents remain under control.

SandPoint (Cambridge, MA) has grappled with the issues facing assistant agents in developing its Hoover search engine. This program works in conjunction with Lotus Notes to perform ad hoc or regularly scheduled searches of Notes' databases and commercial data services and then present the results to people in a Notes-format document. SandPoint has acquired access rights to commercial data sources, and it gives both systemwide and user-level control over how much money an agent should spend on given queries. Tom Henry, SandPoint vice president, says the biggest challenge now is to create a simple user interface so that information search and retrieval using software agents will become as natural for us as picking up the phone or reading a newspaper. Software agents will help accomplish this by knowing where to look and how to find information, according to Henry.

Internet Agents

Agents and their related issues are especially relevant on the Internet. Along with the explosive growth of the WWW (World Wide Web) has come the demand for tools to help us manage the vast amounts of available information, and agents (variously known as WebCrawlers, Spiders, and Robots) often fit the bill (see the text box "WWW Agents").

Some Internet agents attempt to present an integrated view of the Internet as a whole, but the most common to date are information gatherers (see the figure "WWW Topology"). These agents traverse the WWW and then report what they find to a home location. They collect information



A growing demand for information-gathering and filtering tools on the Internet has spawned a host of agents variously known as WebCrawlers, Spiders, and Robots. Some Internet agents attempt to present a view of the Internet as a whole, but the most common, shown here as service agents, travel the WWW and then report what they find.

The spider is an indexer that examines HTML documents with the help of a "mite," which extracts URLs. The indexer then uses the URLs stored in the database to bring documents to the home site, index them, and store the results in the database.