INTELLIGENT AGENTS ON THE INTERNET: Fact, Fiction, and Forecast

Oren Etzioni & Daniel S. Weld*
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
{etzioni, weld}@cs.washington.edu

May 30, 1995

Abstract

Computer technology has dramatically enhanced our ability to generate, deliver, and store information. Unfortunately, our tools for locating, filtering, and analyzing information have not kept pace. A popular solution is intelligent agents. But what are they?

We provide a survey showing the myriad ways in which the evocative term "agent" is interpreted by researchers in the field. Following the "Information Superhighway" metaphor, an intelligent agent may be:

- A backseat driver who makes suggestions at every turn, or
- A taxi driver who drives you to your destination, or even
- A concierge whose knowledge and skills eliminate the need to personally approach the Superhighway at all.

We briefly describe a number of prototype Internet agents and elaborate on the Internet Softbot, a concierge under development at the University of Washington.

*This research was funded in part by Office of Naval Research Grants N00014-94-1-0060 and 92-J-1946, and by National Science Foundation grants IRI-9357772 and IRI-9303461. We acknowledge the members of the Internet Softbot group at the University of Washington: Tony Barrett, David Christianson, Terrance Goan, Keith Golden, Cody Kwok, Neal Lesh, Sujay Parekh, Mike Perkowitz, Richard Segal, Erik Selberg, and Ying Sun.

"The most profound technologies are those that disappear." — Mark Weiser [18]

1 Introduction

Electric motors have exerted a dramatic influence on our lifestyle; for example, there are more than twenty such machines in the average American home. But most people don't notice electric motors in day to day life; instead they focus on higher-level activities that the devices enable: washing clothes or listening to a CD. In contrast, the "Information Superhighway" looms larger and more prominent every day. This article explains how information agents can reverse this trend. Agents have two ways of making the Information Superhighway disappear:

- **Abstraction:** details of the technology underlying the agent and the resources that the agent accesses are abstracted away. The agent enables a person to state *what* information he requires; the agent determines *where* to find the information and *how* to retrieve it.
- **Distraction:** the agent's character or "persona" help to distract the person from what might be a tedious or complex task.

> **Forecast:** While cute agent personalities will have tremendous impact in the entertainment market, they will not be appreciated in operating systems, utilities, or business applications. Microsoft's "Bob" will become extinct for the same reason that talking cars were rejected by consumers.

For information agents, we believe that abstraction will win out over distraction. The pervasive interaction style for today's computers is *direct manipulation*, where users click on, drag, and drop icons. While direct manipulation is handy when the user is performing simple tasks such as printing files or invoking applications on her personal computer, it does not scale to searching massive networks for information. This point was recognized by many visionaries. Alan Kay writes "A retrieval 'tool' won't do because no one wants to spend hours looking through hundreds of networks with trillions of potentially useful items. This is a job for intelligent background processes that can successfully clone their users' goals and then carry them out. Here we want to *indirectly manage agents*, not directly manipulate objects." [12, page 203]. In an article entitled "Hospital Corners," Nicholas Negroponte uses the task of making one's bed as an illustration: "Today, notwithstanding the skill, I cherish the opportunity of delegating the task and have little interest in the *direct manipulation* of my bedsheets... Likewise, I feel no imperatives to manage my computer files, route my telecommunications, or filter the onslaught of mail messages, news, and the like. I am fully prepared to delegate these tasks to agents I trust as I tend to other matters ..." [15, page 347].

> **Fact:** The Internet society estimates that more than 30 million people world-wide have access to the Internet on any given day.

Both Negroponte and Kay suggest that direct manipulation is appropriate for tasks that are enjoyable. While some people enjoy browsing Internet content, there is a growing range of tedious tasks, such as searching for information on the Internet, which most people would gladly delegate to a competent *agent.*

Fact: There are at least 1,500 information brokerages (staffed by *human* information brokers) currently doing business nationwide.

2 What is an Agent?

By *agent*, we mean someone who acts on your behalf. Information agents are loosely analogous to travel agents, insurance agents, *etc.*In the hope of demystifying the term, we enumerate a list of characteristics that have been proposed as desirable agent qualities.[1]

- **Autonomous:** an agent is able to take initiative and exercise a non-trivial degree of control over its own actions:
  - **Goal-oriented:** an agent accepts high-level requests indicating what a human wants and is responsible for deciding how and where to satisfy the requests.
  - **Collaborative:** an agent does not blindly obey commands, but has the ability to modify requests, ask clarification questions, or even refuse to satisfy certain requests.
  - **Flexible:** the agent's actions are not *scripted*; it is able to dynamically choose which actions to invoke, and in what sequence, in response to the state of its external environment.
  - **Self-starting:** unlike standard programs which are directly invoked by the user, an agent can sense changes to its environment and decide when to act.

  Temporal continuity: an agent is a continuously running process, not a "one-shot" computation that maps a single input to a single output, then terminates.
- **Character:** an agent has a well-defined, believable "personality" and emotional state.

[1] This list is based on the ideas originally proposed in [2,6,7,14].

- **Communicative:** the agent is able to engage in complex communication with other agents, including people, in order to obtain information or enlist their help in accomplishing its goals.
- **Adaptive:** the agent automatically customizes itself to the preferences of its user based on previous experience. The agent also automatically adapts to changes in its environment.
- **Mobile:** an agent is able to transport itself from one machine to another and across different system architectures and platforms.

> **Fiction:** General Magic has developed mobile intelligent agents actively roaming the Internet. In fact, General Magic has developed `telescript`, a language for specifying mobile, scripted agents. Actual intelligent agents are still under development. Furthermore, for mobility, each site receiving an agent has to be running a `telescript` interpreter. Few such sites are in existence today.

While no single agent currently has *all* these characteristics, several prototype agents embody a substantial fraction of the properties. Currently, there is little agreement about the relative importance of the different properties, but most agree that these characteristics are what differentiate agents from simple programs.

3 Different Types of Agents

Below we describe a selective sample of information agents currently under development, to give the reader the flavor of agents that will emerge in the next few years. For a more comprehensive collection, see [5,13]. The section is loosely organized by the agent's sophistication and the degree to which they make the Information Superhighway disappear.

People often feel lost or disoriented when navigating through the World Wide Web. One agent that may help alleviate this feeling is a *tour guide* [16]. A tour guide helps the user answer the question "Where do I go next?" Various tour guides have been implemented for relatively small hypertext systems, and prototype systems are being developed for the World Wide Web. For example, the WebWatcher, provides interactive advice to a Mosaic user regarding which hyperlink to follow next, and learns by observing the user's reaction to WebWatcher's advice [1]. Research on the WebWatcher focuses on mechanisms for learning good hyperlink recommendations for an individual user based on past interactions with that user.

Far from making the Web disappear, tour guides draw a person into the Web but attempt to provide a more friendly or even customized experience for the visitor. Of course, badly designed tour guides act much more like "backseat drivers," constantly making annoying suggestions.

The most popular agents on the Web are indexing agents such as Lycos, the WebCrawler, and InfoSeek. Indexing agents carry out a massive, autonomous search of the World Wide Web (scanning over a million documents), and store an index of words in document titles and document texts. The user can then query the agent by asking for documents containing certain key words. Indexing agents can deliver quick responses, but the state of the art in indexing agents has a number of limitations:

- Keyword queries can be rather awkward. For example, it would be difficult to find a reliable florist in Chicago or the weather forecast for Boston or the fax number for the White House using keyword queries.

- The indices are not personalized in any way. As a result, most queries get many false hits.

  The number of false hits appears to grow quickly with the size of the Web, rendering indexing agents unsatisfactory in the face of exponential growth in the amount of content.

  Indexing agents cannot completely index the Web and fail to consider information stored in databases, servers, and information services that are accessible from the Web, but aren't actually Web documents.

Current indexing agents are not selective in their search of the World Wide Web. Quite the contrary, they attempt to be as exhaustive as possible, given their resource limitations. To further elaborate the Information Superhighway metaphor, indexing agents are a bit like taxi drivers. They will take you to your destination (in this case, a particular location on the Web), but there they will drop you off and you are on your own. Furthermore, you may find that you are not at the destination that you wanted to reach.

Forecast: In the next two years, a new breed of indexing agents will emerge. These agents, which we might call "Automated Information Brokers" will choose between existing indices based on factors such as cost of access, coverage, and speed. Furthermore, an information broker will prune the results returned by indexing agents using clues about the contents of each page. For example, it is easy to tell what organization is behind a particular Web document. So we might ask our information broker to show us only product catalogs that originate with AT&T or US West. One such information broker is currently under development at the University of Washington.

A far more selective agent emerging on the Internet is an agent for guiding people to answers to frequently asked questions (FAQs) [10,21]. We call these agents

"FAQ-Finders," after the University of Chicago agent. People tend to ask the same questions over and over again. In response, newsgroups, support staffs, and other organizations developed files of frequently asked questions and their answers. For example, the Gopher FAQ file contains answers to questions such as "What is Gopher?" and "Where can I get Gopher software?" A person approaching the Internet with a question may be convinced that the answer is out there somewhere in some FAQ file, but may be unable to locate the appropriate file. FAQ-Finders attempt to address this problem by indexing large numbers of FAQ files and providing an interface where people can pose their questions in natural language, and the text of the question is used to index into the appropriate answer. In contrast to indexing agents, FAQ-Finders only retrieve answers to questions that appear in the FAQ files they index. Because of the semi-structured nature of the files, and because the number of files is much smaller than the number of documents on the World Wide Web, FAQ-Finders have the potential to be much more reliable than general-purpose indexing agents.

Fiction: Eventually all information will be available on the Internet.

All of the systems described thus far presuppose that the information being sought is directly available on the Internet. But as Kautz, et al. [11] argue, much of the information people are looking for is not available on-line but only "in people's heads." Therefore, another task which can be facilitated by information agents is "expertise location:" putting the questioner in touch with appropriate experts.

Both tour guides and indexing agents operate by suggesting locations on the Web to the user. The suggestions are based on a relatively weak model of what the user wants, and what information is available at the suggested location. In contrast, the Internet Softbot represents a more ambitious attempt to both determine what the user wants and understand the contents of information services in order to provide better responses to information requests. The agents we have described thus far access unstructured documents or semi-structured information such as FAQ files. The Internet Softbot [6] tackles a different component of information on the Internet: structured information services such as weather map servers, stock quote servers, Federal Express's package tracking service, the `NetFind` service, *etc.*Because the information is structured, the Softbot need not rely on natural language or information retrieval techniques to "understand" the information provided by a service. Instead, the Softbot relies on a model of the service to give it the precise semantics associated with information provided by the service. As a result, the Softbot is able to answer focused queries with relatively high reliability. The next section explains the principles underlying the Softbot's operation.

4 Case Study: Internet Softbot

To help make the previous discussion concrete we briefly describe the University of Washington's Internet Softbot. We do not mean to imply that the Softbot is the ideal software agent, but it illustrates several important themes.

The key insight behind the Softbot architecture is visible in the name which stems from "Software robot." Instead of a mobile robot's arms and wheels, the Softbot has software commands such as `ftp`, `lpr` and `mail` for effectors; instead of a sonar array or television camera, the Softbot uses commands such as `ls`, `finger`, and `netfind` to gather information. Internally, the Softbot is implemented with a least-commitment planner that provides behavioral control. Several technical innovations were required to render this approach successful in the complex world of the Internet (see Section 4.3).

4.1 Objective: A Concierge

The Internet Softbot is a prototype implementation of a high-level assistant, such as a hotel concierge. In contrast to systems for assisted browsing or information retrieval, the Softbot accepts high-level user goals and dynamically synthesizes the appropriate sequence of Internet commands to satisfy those goals. The Softbot executes the sequence, gathering information to aid future decisions, recovering from errors and retrying commands if necessary. We think of Softbot as a concierge because it eliminates the need for a person to drive the Information Superhighway at all; that job is delegated to the Softbot. The following are examples of the types of requests that the Softbot has handled:

- **Notification Requests:** The Softbot can monitor a variety of events (*e.g.*, disk utilization, user activity, bboards, remote `ftp` sites *etc.*) and communicate its findings to the user in several different ways (*e.g.*, a beep, a message displayed on the screen, an e-mail message, or output at a printer). The Softbot determines autonomously how to monitor the events (*i.e.*, which commands to execute and how often to execute them) as well as how to notify the user of its findings.

- **Enforcing Constraints:** In addition to information gathering requests, the Softbot can act on the world around it. For example, one can ask it to ensure that all files in a shared directory are group-readable, that the contents of an archival directory are compressed, *etc.*

- **Locating and Manipulating Objects:** The Softbot can compile source code, convert documents from one format to another, and access remote databases. Since users find it convenient to communicate partial descriptions of people, objects, and information sources, the Softbot automatically executes commands until it has disambiguated the request. For example, a request to print a file quickly causes the Softbot to determine the queue size and status (*e.g.*, "jammed") of the nearby printers. When given the name of a person (but not their email address), the Softbot uses a combination of `whois`, `netfind`, `staffdir`, `finger`, and other utilities to determine an electronic address.

The above sample tasks are neither exhaustive nor mutually exclusive, but illustrate our main point: the Softbot allows a user to specify *what* to accomplish, leaving the decision of *how* and *where* to accomplish it to the Softbot. Thus the Internet Softbot is an excellent example of a "goal-oriented" agent.

4.2 Specifying Goals

Goal-orientation will only be useful if it is easier for a user to specify her request than to carry out the activity herself. Three criteria must be satisfied to make specifying goals convenient:

- **Expressive Goal Language:** If common goals are impossible to specify, then a goal-oriented software agent is of limited use. To avoid this pitfall, the Softbot accepts goals containing complex combinations of conjunction, disjunction, negation, and nested universal and existential quantification. This allows specification of tasks such as "Get all of Kambhampati's technical reports that aren't already stored locally." Note that the Softbot can handle this goal even though the underlying services (*e.g.*, `ftp`) does not handle this combination of universal quantification and negation. The Softbot determines which of Kambhampati's reports are new and issues `ftp` commands to obtain them.

- **Convenient Syntax:** Despite the expressive power of mathematical logic, many users are unable (or unwilling) to type long, complex, quantifier-laden sentences. For this reason, the Softbot supplies a forms-based graphical user interface and automatically translates forms into the logical goal language. User can quickly construct complex quantified goals with a few mouse clicks and a minimum of typing. (Natural language input is an alternative approach pursued by many researchers, but not yet incorporated into the Softbot).

  Mixed-Initiative Refinement Dialog: Even with a well-engineered interface, it is difficult to specify orders perfectly. For this reason, human assistants commonly engage their superiors in dialog "Do you *really* need it delivered by 8am? It would cost much less to arrange for delivery by 9am instead." The current Softbot interface has only a limited support for iterative refinement of goals, but we are designing a new interface that will allow the Softbot to pose questions (while it continues to work) and allow the user to add information and constraints if she so desires.

4.3 Softbot Architecture

In total, the Softbot architecture consists of four major modules — task manager (approximately 10% of the code), XII planner (25%), model manager (25%), and Internet domain models (30%) — in addition to miscellaneous support code (10%).

- **The Task Manager** resembles an operating system scheduler. All important Softbot activities (*i.e.*, both cognitive tasks such as planning and active tasks such as connecting to a `gopher` server) are controlled by this mechanism. As a result, the Softbot can schedule activities for future execution.

  The XII Planner is an extension of UCPOP [17,19] that handles planning with incomplete information [4]. XII searches the space of plans — partially ordered sets of partially specified actions which have both observational and causal effects — until it finds a plan that will achieve the current goal. Since incomplete information requires that the Softbot gather information in order to determine a plausible course of action, XII interleaves planning and execution. For example, when requested to "Get all of Kambhampati's technical reports that aren't stored locally," XII first executes actions to locate Kambhampati's host, then connects to see what reports are available, then executes actions to see which are stored locally, and only then plans the detailed `ftp` actions that will achieve the goal. See [9] for a description of the innovations required in order to plan efficiently in the face of incomplete information.

- **The Model Manager** is a specialized database that stores everything the Softbot has observed about the world. In order to support the XII planner, the model manager implements pattern-directed queries via unification. The most novel aspect of the model manager is its capacity for local closed-world reasoning [3]. Closed-world reasoning — the ability to draw conclusions based on the assumption that one knows about the existence of all relevant objects — is essential for goal-directed behavior [8]. For example, when directed to find the cheapest direct flight, travel agents assume that after accessing their collection of databases they have information about *every* relevant flight. Underlying the Softbot model manager is the insight that closed-world reasoning is both essential and dangerous. Clearly, the Internet is so vast that the Softbot can't assume that it knows the contents of *every* database on *every* host. However, after accessing the SABRE reservation system, the Softbot must be able to conclude that it has *local* closed world information: it knows the price of every flight between the cities in question. The Softbot model manager performs fast inference on local closed-world information — if the user later specifies that the carrier must be United Airlines, then the Softbot need not access SABRE again. But if the Softbot is informed of an unspecified fare increase or the creation of a new flight, then it will retract its conclusion of local closed world information and gather more information.

- **The Internet Domain Models** provide the Softbot with background information about the Internet. The most important part of these models are declarative encodings of the actions available to the Softbot (*i.e.*, descriptions of UNIX commands and Internet databases and services). Also included are background information about people and machines in addition to search control heuristics.

4.4 Properties

Referring back to the discussion of Section 2, we note that the Softbot has many, but not all, of the properties associated with agent-hood. The Softbot is highly autonomous; it is goal-directed, flexible and self-starting. In principle, the Softbot has temporal continuity, but in practice it rarely survives for more than a few days before needing rebooting. We are currently striving to augment the Softbot's rudimentary collaborative, communicative, and adaptive characteristics, but the Softbot is not mobile nor do we see any reason for it to be so. Given the "reach" of its effectors, there appears to be no need for the Softbot to transport itself to another host machine; indeed, mobility might compromise the Softbot's ability to keep user goals and preferences confidential.

5 Conclusion

Although this paper described working prototypes of several significant software agents, considerable research is necessary before intelligent agents fulfill their potential capabilities. Below, we list some of the research challenges that seem most pressing in service of goal-orientation, collaborativity, and adaptivity.

- **Algorithms for planning with incomplete information.** The dramatic dynamism of the Internet suggests that these planners must be capable of gathering information as part of the planning process. The Internet's vast size implies that planners must be extremely efficient, have excellent search control heuristics, or exploit past experiences using case-based techniques. Planning technology has been successfully used to guide the Internet Softbot, but the problem of planning with incomplete information is certainly not "solved." The Internet Softbot requires extensive, hand-coded, search control rules in order to handle the goals listed in Section 4.1. If effective search control rules could be automatically learned, the Softbot would be capable of handling many more goals.

- **Multi-agent communication techniques.** Software agents must communicate with each other, and with their human masters. In particular, agents should be able to refine requests and queries through evolving dialog. It is likely that all effective communication and collaboration techniques will rely on shared encodings of large amounts of common sense information about computers, networks, useful services, and human activities. But the communication algorithms should allow one agent to convey new vocabulary (or the capabilities of a third agent) to another.

- **Learning algorithms.** Machine learning techniques will allow an agent to adapt to its user's desires as well as to a changing environment.

  Cyberperception. Just as physical animals are aware of changes in their environment, software agents must be able to sense their surroundings. Machine load, network traffic, and human activities greatly affect which actions a software agent should execute. Agents must also be able to take unstructured and semistructured documents and extract facts from them (perhaps using image processing and information retrieval algorithms). Until full natural language processing is a reality, semantic mark-up formats will be crucial. We need to provide the information service authors with a formal language for describing the contents of their information sources in a way that is accessible to software agents.

  Softbot safety systems. Without some guarantee that it holds its master's goals and interests inviolate, a powerful software agent is more dangerous than desirable. We need techniques capable of ensuring that an agent will not inadvertently harm its owner while in service of an explicit request [20].

Forecast: The Internet and the World Wide Web will eventually disappear. They will become invisible layers, much in the same way that registers and machine instructions have been supplanted by higher programming languages in today's computers.

In the future, naive users, busy executives, and people requesting information over a low bandwidth channel (*e.g.*, from handheld computers via cellular links) will refuse to be thrust directly onto the Information Superhighway. They will prefer to delegate responsibility to reliable, intelligent agents that can handle simple tasks without supervision and can provide concise answers to complex queries.

5.1 For Further Information (Sidebar)

We have described a wide range of agents including backseat drivers, tour guides, taxi drivers, culminating in a concierge — an agent that eliminates the need to personally approach the Information Superhighway. For additional details about several intelligent-agent projects, browse the following Web sites:

- A tour guide for the Web being developed at Carnegie-Mellon University. `http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html`

- The FAQFinder project at the University of Chicago. `http://cs-www.uchicago.edu/burke/faqfinder.html`
- An information mediator called SIMS is being developed at the Information Sciences Institute. `http://www.isi.edu/sims/sims-homepage.html`
- The autonomous agent group at MIT's Media Lab. `http://agents.www.media.mit.edu/groups/agents/`
- The Nobotics Research Group at Stanford University. `http://robotics.stanford.edu/groups/nobotics/home.html`

  The Internet Softbot project at the University of Washington. `http://www.cs.washington.edu/research/softbots`

References

[1] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. Webwatcher: A learning apprentice for the world wide web. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, pages 6–12, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[2] Phlip R. Cohen, Adam Cheyer, Michelle Wang, and Soon Cheol Baeg. An open agent architecture. In *Working Notes of the AAAI Spring Symposium: Software Agents*, pages 1–8, Menlo Park, CA, 1994. AAAI Press. To order a copy, contact sss@aaai.org.

[3] O. Etzioni, K. Golden, and D. Weld. Tractable closed-world reasoning with updates. In *Proc. 4th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 178–189, June 1994.

[4] O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. An Approach to Planning with Incomplete Information. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning*, October 1992. Available via FTP from `pub/ai/` at `ftp.cs.washington.edu`.

[5] O. Etzioni, P. Maes, T. Mitchell, and Y. Shoham, editors. *Working Notes of the AAAI Spring Symposium on Software Agents*, Menlo Park, CA, 1994. AAAI Press. To order a copy, contact sss@aaai.org.

[6] O. Etzioni and D. Weld. A softbot-based interface to the internet. *CACM*, 37(7):72–76, July 1994.

[7] Oren Etzioni, Neal Lesh, and Richard Segal. Building softbots for UNIX (preliminary report). Technical Report 93-09-01, University of Washington, 1993. Available via anonymous FTP from `pub/etzioni/softbots/` at `cs.washington.edu`.

[8] M. Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, San Mateo, CA, 1987.

[9] K. Golden, O. Etzioni, and D. Weld. Omnipotence without omniscience: Sensor management in planning. In *Proc. 12th Nat. Conf. on A.I.*, pages 1048–1054, July 1994.

[10] Kristen Hammond, Robin Burke, Charles Martin, and Steven Lytinen. Agent amplified communication. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, pages 69–73, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[11] Henry A. Kautz, Al Milewski, and Bart Selman. Agent amplified communication. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, pages 78–84, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[12] Alan Kay. User interface: A personal view. In Brenda Laurel, editor, *The Art of Human Computer Interface Design*, pages 191–207. Addison-Wesley, 1990.

[13] Craig Knoblock, Alon Levy, Su-Shing Chen, and Gio Widerhold, editors. *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[14] Craig A. Knoblock and Yigal Arens. An architecture for information retrieval agents. In *Working Notes of the AAAI Spring Symposium: Software Agents*, pages 49–56, Menlo Park, CA, 1994. AAAI Press. To order a copy, contact sss@aaai.org.

[15] Nicholas Negroponte. Hospital corners. In Brenda Laurel, editor, *The Art of Human Computer Interface Design*, pages 347–353. Addison-Wesley, 1990.

[16] Tim Oren, Gitta Salomon, Kristee Kreitman, and Abbe Don. Guides: Characterizing the interface. In Brenda Laurel, editor, *The Art of Human Computer Interface Design*, pages 367–381. Addison-Wesley, 1990.

[17] J.S. Penberthy and D. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 103–114, October 1992. Available via FTP from `pub/ai/` at `ftp.cs.washington.edu`.

[18] Mark Weiser. The computer for the 21st century. *Scientific American*, 1995. Special Issue: The Computer in the 21st Century.

[19] D. Weld. An introduction to least-commitment planning. *AI Magazine*, pages 27–61, Winter 1994. Available via FTP from `pub/ai/` at `ftp.cs.washington.edu`.

[20] D. Weld and O. Etzioni. The first law of robotics (a call to arms). In *Proc. 12th Nat. Conf. on A.I.*, July 1994. Available via FTP from `pub/ai/` at `ftp.cs.washington.edu`.

[21] Steven D. Whitehead. Auto-faq: An experiment in cyberspace leveraging. In *Proceedings of the Second International WWW Conference*, volume 1, pages 25–38, Chicago, IL, 1994. (See also: `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/whitehead/whitehead.html`).