Adaptive Web Sites: an AI Challenge

Mike Perkowitz[*] **Oren Etzioni**
Department of Computer Science and Engineering, Box 352350
University of Washington, Seattle, WA 98195
{map, etzioni}@cs.washington.edu
(206) 616-1845 Fax: (206) 543-2969

Abstract

The creation of a complex web site is a thorny problem in user interface design. First, different visitors have distinct goals. Second, even a single visitor may have different needs at different times. Much of the information at the site may also be dynamic or time-dependent. Third, as the site grows and evolves, its original design may no longer be appropriate. Finally, a site may be designed for a particular purpose but used in unexpected ways.

Web servers record data about user interactions and accumulate this data over time. We believe that AI techniques can be used to examine user access logs in order to automatically improve the site. We challenge the AI community to create **adaptive web sites**: sites that automatically improve their organization and presentation based on user access data.

Several unrelated research projects in plan recognition, machine learning, knowledge representation, and user modeling have begun to explore aspects of this problem. We hope that posing this challenge explicitly will bring these projects together and stimulate fundamental AI research. Success would have a broad and highly visible impact on the web and the AI community.

1 Introduction

The World Wide Web is becoming a key medium for information dissemination, entertainment, and communication. Examples include personal home pages, on-line malls, university course information, and much more. Many web sites quickly sprout intricate collections of pages and hyperlinks as they begin to mirror the complexity of the information they convey.

Designing a rich web site so that it readily yields its information can be tricky. Unlike the oyster that contains a single pearl, a web site often contains myriad facts, images, and hyperlinks. Many different visitors approach a popular web site — each with his or her own goals and concerns. Consider, for example, the web site for a typical computer science department. The site contains an amalgam of research project descriptions, course information, lists of graduating students, pointers to industrial affiliates, and much more. Each nugget of information is of value to someone who would like to access it readily. One might think that a well organized hierarchy would solve this problem, but we've all had the experience of banging our heads against a web site and crying out "it's got to be here *somewhere*...".

The problem of good web design is compounded by several factors beyond the fact that different visitors have distinct goals. First, the same visitor may seek different information at different times. Second, many sites outgrow their original design, accumulating links and pages in unlikely places. Third, a site may be designed for a particular kind of use, but be used in many different ways in practice; the designer's *a priori* expectations may be violated. Too often web site designs are fossils cast in HTML, while web navigation is dynamic, time-dependent, and idiosyncratic. We challenge the AI community to address this problem by creating **adaptive web sites:** *web sites that automatically improve their organization and presentation by learning from user access patterns.*

In essence, web design is a problem in user interface design. However, in contrast with vendors of shrink-wrapped software, few web site designers can afford to subject their web sites to formal usability testing in special labs. Fortunately, web users interact directly with a server maintained by the inventors of the service or authors of the content being served. As a result, data on their behavior is recorded in web server logs (see Figure 1). Because this raw data is overwhelming for an overworked webmaster to process regularly, web server logs are ripe targets for automated analysis.

Our challenge then is this: how can we build a web site which improves itself over time in response to user interactions with the site? This challenge poses a number of

[*]This research was funded in part by Office of Naval Research grant 92-J-1946, by ARPA / Rome Labs grant F30602-95-1-0024, by a gift from Rockwell International Palo Alto Research, and by National Science Foundation grant IRI-9357772.

```
24hrlab-214.sfsu.edu - - [21/Nov/1996:00:01:05 -0800] "GET /home/jones/collectors.html HTTP/1.0" 200 13119
24hrlab-214.sfsu.edu - - [21/Nov/1996:00:01:06 -0800] "GET /home/jones/madewithmac.gif HTTP/1.0" 200 855
24hrlab-214.sfsu.edu - - [21/Nov/1996:00:01:06 -0800] "GET /home/jones/gustop2.gif HTTP/1.0" 200 25460
x67-122.ejack.umn.edu - - [21/Nov/1996:00:01:08 -0800] "GET /home/rich/aircrafts.html HTTP/1.0" 404 617
x67-122.ejack.umn.edu - - [21/Nov/1996:00:01:08 -0800] "GET /general/info.gif HTTP/1.0" 200 331
203.147.0.10 - - [21/Nov/1996:00:01:09 -0800] "GET /home/smith/kitty.html HTTP/1.0" 200 5160
24hrlab-214.sfsu.edu - - [21/Nov/1996:00:01:10 -0800] "GET /home/jones/thumbnails/awing-bo.gif HTTP/1.0" 200 5117
```

Figure 1: Typical user access logs, these from a computer science web site. Each entry corresponds to a single request to the server and includes originating machine, time, and URL requested. Note the series of accesses from each of two users (one from SFSU, one from UMN).

difficult, but not impossible, questions:

- **What kinds of generalizations can we draw from user access patterns and what kinds of changes could we make?** Suppose we maintain a web site containing information about various automobiles, organized by manufacturer. We observe that visitors who look at the *Ford Windstar* minivan page also tend to look at the *Dodge Caravan* and *Mazda MPV* minivan pages. We might therefore create a new page for minivans, which cuts across the existing manufacturer-based organization and provides a new *view* of the site.
- **How do we design a site for adaptivity?** We might specifically design parts of the site to be changeable. For example, we might present our users with a "tour guide" (as in [Armstrong *et al.*, 1995]) and have changes to the site be presented as the agent's suggestions. Alternatively, we might annotate our HTML with directives stating where and how changes can be made. Or we may provide semantic information about the entire site, allowing the agent to reason about the relationships between everything, perhaps representing the entire site as a database (see [Fernandez *et al.*, 1997]).
- **How do we effectively collaborate with a human webmaster to suggest and justify potential adaptations?** Suppose the human webmaster is still responsible for the final product. Instead of changing web pages directly, our system might accumulate observations and suggested changes and present them to the webmaster, clearly explaining its observations and justifying the changes it recommends.
- **How do we move beyond one-shot learning algorithms to web sites that continually improve with experience?** Over time, our adaptive web site will accumulate a great deal of data about its users and should be able to use its rich history to continually evolve and improve.

Our department maintains a web site for its introductory computer science course. This site contains schedules, announcements, assignments, and other information important to the hundreds of students who take the course every quarter. Enough information is available that important documents can be hard to find or entirely lost in the clutter. Imagine, however, if the site were able to determine what was important and make that information easiest to find. Important pages would be available from the site's front page. Important links would appear at the top of the page or be highlighted. Timely information would be emphasized, and obsolete information would be quietly moved out of the way.

There are several factors that make this challenge both appropriate and timely for the AI community. First, the growing popularity and complexity of the web underscores the importance of the challenge. Second, virtually all existing web sites are *not* adaptive, yet data to support the learning process is readily available in web server logs. Clearly, here is an opportunity for AI! Finally, a number of disconnected projects in machine learning [Armstrong *et al.*, 1995], data mining, knowledge representation, plan recognition [Kautz, 1987; Pollack, 1990], and user modeling [Fink *et al.*, 1996] have begun to explore aspects of the problem. Framing the problem explicitly in this paper could help bring these disparate approaches together.

We pose our challenge as a particular task to be accomplished by any means available. Many advances in artificial intelligence, both practical and theoretical, have come about in response to such task-oriented approaches. The quest to build a better chess-playing computer, for example, has led to many advances in search techniques (e.g., [Anantharaman *et al.*, 1990]). The autonomous land vehicle project at CMU [Thorpe, 1990] has resulted in not only a highway-cruising vehicle but also breakthroughs in vision, robotics, and neural networks. The quest to build autonomous software agents has similarly led to both practical and theoretical advances. For example, the Internet Softbot project has yielded both deployed softbots and advances in planning, knowledge representation, and machine learning [Etzioni, 1996].

We believe that the goal of creating self-improving web sites is a similar task: one whose accomplishment will require breakthroughs in different areas of AI. In this paper we discuss possible approaches to this task and how to evaluate the community's progress. In section 2, we present two basic approaches to creating an adaptive web site. We illustrate both with ongoing research and examples. In section 3, we discuss how to evaluate research on this challenge, discussing practical alternatives as well as open questions. Throughout, we pose **Challenge** questions intended to suggest research directions and illustrate where the open questions lie.

2 Approaches to adaptive web sites

Sites may be adaptive in two basic ways. First, the site may focus on *customization*: modifying web pages in real time to suit the needs of individual users. Second, the site may focus on *optimization*: altering the site itself to make navigation easier for all. We illustrate these two basic approaches with examples drawn from current AI research. Whether we modify our web pages online or offline, we must use information about user access patterns and the structure of our site. Much of this information is available in access logs and in the site's HTML, but this may not be sufficient; we also discuss how to support adaptivity with *meta-information* — information about page content. Finally, we examine other issues that arise in designing adaptive web sites.

2.1 Customization

Customization is adjusting the site's presentation for an individual user. Customization allows fine-grained improvement, since the interface may be completely tailored to each individual user. One way for a site to respond to particular visitors is to allow *manual customization*: allowing users to specify display options that are remembered during the entire visit and from one visit to the next. The Microsoft Network (at http://www.msn.com), for example, allows users to create home pages with customized news and information displays. Every time an individual visits her MSN home page, she sees the latest pickings from the site presented according to her customizations.

Path prediction, on the other hand, customizes *automatically* by attempting to guess where the user wants to go and taking her there more quickly. A path prediction system must answer at least the following questions.

- **What are we predicting?** We may try to predict the user's next step. For example, if we can predict what link on a page a particular user will follow, we might highlight the link or bring it to the top of the page. Alternatively, we may try to predict the user's eventual goal; if we can determine what page at the site a visitor is looking for, we can present it to her immediately.
- **On what basis do we make predictions?** We might use only a particular individual's actions to predict where she will go next. On the other hand, we might generalize from multiple users to gather data more quickly.
- **What kinds of modifications do we make on the basis of our predictions?** We may do as little as highlighting selected links (by making them bold or putting graphics around them, for example) or as much as synthesizing a brand new page that we *think* the user wants to see.

The WebWatcher [Armstrong *et al.*, 1995] (see http://www.cs.cmu.edu/ ~webwatcher/) learns to predict what links users will follow on a particular page as a function of their specified interests. WebWatcher observes many users over time and attempts to learn, given a user's current page and stated interests, where she will go next. A link that WebWatcher believes you are likely to follow will be highlighted graphically and duplicated at the top of the page. Visitors to a site are asked, in broad terms, what they are looking for. Before they depart, they are asked if they found what they wanted. WebWatcher uses the paths of people who indicated success as examples of successful navigations. If, for example, many people who were looking for "personal home pages" follow the "people" link, then WebWatcher will tend to highlight that link for future visitors with the same goal.

Instead of predicting a user's next action based on the actions of many, we might try to predict the user's ultimate goal based on what she has done so far. *Goal recognition* [Kautz, 1987; Pollack, 1990] is the problem of identifying, from a series of actions, what an agent is trying to accomplish. Lesh and Etzioni [Lesh and Etzioni, 1995] pose this problem in a domain-independent framework and investigate it empirically in the Unix domain: by watching over a user's shoulder, can we figure out what she is trying to accomplish (and offer to accomplish it for her)? They model user actions as planning operators. Assuming users behave somewhat rationally, they use these actions' precondition/postcondition representation to reason from what a user has done to what she must be trying to do. In the web domain, we observe a visitor's navigation through our site and try to determine what page she is seeking. If we can do this quickly and accurately, we can then offer the desired page immediately.

Challenge: Can we formalize user navigation of the web as a planning process that is amenable to goal recognition? Do user actions on the web carry enough evidence of their purpose?

The AVANTI Project [Fink *et al.*, 1996] (see http://zeus.gmd.de/ projects/avanti.html) focuses on dynamic customization based on users' needs and tastes. As with the WebWatcher, AVANTI relies partly on users providing information about themselves when they enter the site. Based on what it knows about the user, AVANTI attempts to predict both the user's eventual goal and her likely next step. AVANTI will prominently present links leading directly to pages it thinks a user will want to see. Additionally, AVANTI will highlight links that accord with the user's interests. AVANTI is illustrated on a hypothetical Louvre Museum web site. For example, when a disabled tourist comes to the site, links regarding disabled access and tourist information are emphasized. AVANTI relies on users providing some information about themselves in an initial dialogue; the site then uses this information to guide its customization throughout the user's exploration of the site. AVANTI also attempts to guess where the user might go based on what she has looked at so far. For example, if our disabled tourist looks at a number of paintings at the site, AVANTI will emphasize paintings links as it continues to serve pages. As with the WebWatcher, we might ask if we can avoid AVANTI's requirement that users explicitly provide information.

2.2 Optimization

Whereas customization focuses on individuals, *optimization* tries to improve the site as a whole. Instead of making changes for each user, the site learns from all users to make the site easier to use. This approach allows even new users, about whom we know nothing, to benefit from the improvements.

We may view a web site's design as a particular point in the vast space of possible designs. Improving the site, then, corresponds to searching in this space for a "better" design. Assuming we have a way of measuring "better", we may view this as a classical AI search problem. One possible quality metric would be to measure the amount of *effort* a visitor needs to exert on average in order to find what she is looking for at our site. Effort is defined as a function of the number of links traversed and the difficulty of finding those links. For example, a site whose most popular local page is buried five links away from the front page could be improved by making that page accessible from a readily obvious link on the front page. We can navigate through this space by performing transformations on the site — adding or removing links, rearranging links, creating new web pages, etc. If we guarantee that each transformation improves the quality of the site, we are performing a hillclimbing search.

Challenge: How large is this search space and what is an appropriate search strategy? Can we restructure the space to avoid searching large portions of it?

In [Perkowitz and Etzioni, 1997] we sketch the design of a system with a repertoire of transformations that aim to improve a site's organization; transformations include rearranging and highlighting links as well as synthesizing new pages. Our system learns from common patterns in the user access logs and decides how to transform the site to exploit those patterns and make the site easier to navigate. For example, the web site for our department's introductory computer science course contains a web page for each homework assignment given during the course. After each assignment's due date, a solution set for that assignment is made available. Our system would observe that after an assignment's due date many visitors look at the solution set; in fact, the most recent solution set is one of the most popular pages at the site. This observation would lead the system to *promote* the solution set by giving it a prominent link on the front page. Promotion — making the link to a page more prominent — is a simple but effective transformation. We have implemented a form of promotion on an existing web site and have found that approximately 10% of our 10,000-15,000 daily page accesses are through automatically generated links; roughly 25% of all visitors click through at least one such link. Of course, we note that promoting a link may be a self-fulfilling prophecy — making a page more prominent may increase its popularity, artificially inflating the site's apparent success at adaptation.

A more ambitious transformation is *clustering* — synthesizing a brand new web page that contains links to a set of related objects. From available data, the system must infer that a set of pages at the site are related and group them together. This inference might be based on content (e.g., when a number of pages cover the same topic) or on user navigation patterns (e.g., when visitors to one page are particularly likely to visit certain others). As final exams approach, students tend to look at multiple solutions sets on each visit. Even though the solution pages are not linked together directly, visitors navigate from one to another (via intervening pages) on their own. This pattern suggests that the solution sets form a meaningful group in our visitors' heads, which does not appear on our web site — solution sets are only linked to from their respective assignment pages. Our system would create a new page with a link to each solution set and make this new page available to visitors to the site. We are currently implementing clustering transformations based on user navigation data.

2.3 Meta-information

A web site's ability to adapt can be hampered by the limited knowledge about its content and structure provided by HTML. For example, suppose that a page contains a list of links. Is it appropriate to add a new link at the top of the list? The answer depends on the contents of the list — an adaptive site should not add a link to a course's home page to a list of links to faculty home pages; furthermore, if the list is in alphabetical order then a new item can only be added at the appropriate point. Clearly, a site's ability to adapt could be enhanced by providing it with *meta-information*: information about its content, structure, and organization. In this section, we discuss means of providing an adaptive site with this sort of information.

One way to provide meta-information is to represent the site's content in a formal framework with precisely defined semantics such as a database or a semantic network. This approach is pioneered by the STRUDEL web-site management system [Fernandez *et al.*, 1997] which attempts to separate the information available at a web site from its graphical presentation. Instead of manipulating web sites at the level of pages and links, web sites may be specified using STRUDEL's view-definition language. In addition, web sites may be created and updated by issuing STRUDEL queries. For example, a corporation might create home pages for its employees by merging data from its "manager" and "employee" databases. A page would be created for every person in either database. Furthermore, each manager's page would have links to her employees, and vice-versa.

This approach would facilitate adaptivity because STRUDEL would enable a site to reason about its logical description and detect cases where adaptations would violate the existing logic. Furthermore, an adaptive site could easily transform itself by issuing STRUDEL queries; STRUDEL provides the mechanisms to automatically update the site appropriately. The drawback of the STRUDEL approach is that it requires the site's entire content to be encoded in a set of databases or in *wrappers* that map web pages and other information sources into STRUDEL. The cost of constructing such wrappers for existing web sites, and particularly for relatively unstructured sites, appears to be high.

A lighter-weight approach is to annotate an existing web site with meta-content tags. In this approach, a formal description of the content coexists with HTML documents. We may choose how much of the site to annotate and how complex our annotations will be. Yet, meta-content annotation still facilitates reasoning about the connections between parts of the site and still provides guidance as to where and how to make changes.

One approach of this type is Apple's Meta-Content Format (see http://mcf.research.apple.com). MCF is an attempt to establish a standard for meta-content annotation for the web. When a user visits an MCF-enhanced site with an MCF-enabled browser, she can choose to navigate the site in a three-dimensional representation of the site's structure, as determined from the site's MCF annotation. SHOE [Luke *et al.*, 1997] (at http://www.cs.umd.edu/projects/plus/SHOE/), takes a different tack. SHOE is a language for adding simple ontologies to web pages. SHOE adds basic ontological declarations to HTML; a page can refer to a particular ontology and declare classifications for itself and relations to other pages. In their example, a man's home page is annotated with information about him, such as the fact that he is a person, his name, his occupation, and his wife's identity (she has her own home page). SHOE is designed to facilitate the exploration of agents and the workings of search tools, but ontological annotation could also support adaptation.

While lighter-weight than STRUDEL, meta-content tagging also has clear disadvantages. First, because the meta-content annotation is separated from the actual content, it has to be updated manually as the content changes. Second, since the meta-content is attached to existing HTML, it provides no direct support for automatic adaptation; Any adaptation must still modify the original HTML.

Each of the approaches described so far require a fair amount of effort to build and maintain the content descriptions. If we wish only to facilitate adaptation, this effort may be overkill. An alternative that we are actively investigating is to use an extremely lightweight annotation system designed specifically for adaptivity. These annotations would be in the form of directives to the adaptive system telling it where it may (or may not) make changes and what kinds of changes it might make. For example, we might add a *list* tag to HTML to allow us to describe the elements in a list and how they are ordered. A list might be declared as `<list order="unordered">`, which tells the system it may reorder the list in any way it chooses. Or a list might be declared `<list order="popularity">`, in which case the system will draw upon data from access logs to determine how to present the list. A list declared `<list order="alphabetical">` or `<list order="chronological">` can be modified by additions or deletions so long as its original ordering constraint is preserved.

We present tags of this sort as part of an "Adaptive HTML" language called A-HTML in [Perkowitz and Etzioni, 1997]. Our intention is to extend HTML to a higher level of abstraction, allowing a web designer to describe objects in terms of their time-relevance, organization, and interrelationships. Note that this approach does not require the global establishment of an A-HTML standard; the adaptive site uses a server capable of interpreting A-HTML and translating it into standard HTML at runtime. Only the resulting HTML is served in response to page requests.

2.4 Open Questions

The quest for a self-improving web site raises a number of related questions. An adaptive site will be active twenty-four hours a day, seven days a week. The site will constantly be ingesting and analyzing data, adjusting its concepts and models, and updating its own structure and presentation. Over time, this constant cycle will reflect many hours of experience and refinement. In the past, AI research has focused on single trials and short-lived entities: systems that run their experiments and shut down, to start again the next day with a blank slate. Although such an approach may be applied to the adaptive site challenge, the most intelligent site will surely be one that continually accumulates knowledge about pages, users, content, and itself.

User interface design is difficult enough for human beings to perform well. Yet an adaptive web site will have to take into account all the artistry of good design in its self-improvements. We can limit the scope of the system's ability to change itself, thus ensuring that it cannot do too much harm, but this means we also limit its scope for improvement. On the other hand, giving the system free rein for radical transformation might mean giving it free rein for radical screwup.

Challenge: How do we formalize the concept of good design? How do we limit the potential for harm without overly limiting the potential for good?

We might instead put the AI system in the role of advisor to a human master. Instead of making changes under cover of night, our AI system must now intelligently present suggestions to a human being, complete with explanation and justification. Such a solution frees us from the problem of changing details without changing design but presents us with a new interface challenge.

Challenge: How does our adaptive web site communicate its suggestions to a webmaster?

3 Evaluation

Although the problem of measuring the quality of a web site design is thorny, we have identified several preliminary approaches. Progress on the design of adaptive web sites will include more sophisticated methods of evaluating a site's usability. We propose a basic metric for how usable a site is: how much effort must a user exert on average in order to find what she wants? As discussed in section 2.2, effort can be defined as a function of the number of links traversed and the difficulty of finding the links on their pages. The standard daily access log may be used to approximately measure user effort.

However, standard log data is not sufficient to know everything about visitor navigation. For example, standard logs do not distinguish between individuals connecting from the same location or record which link a user followed. However, software is available to provide more complete information. WebThreads, for example (see http://www.webthreads.com), allows a site to track an individual user's progress, including both pages visited and links followed. Along with analysis of our site's structure, data from a system like WebThreads is sufficient for us to measure user effort.

Analysis of our user logs provides much information about how users interact with the site. In addition, we may use controlled tests with subjects. Such tests have the advantage of allowing us to observe users as they interact with the site – we get much more information than is encoded in user access logs. As subjects perform tasks such as finding information, downloading software, or locating documents, we may gather data such as:

- Whether the subject succeeded at the task (or realized it was not solvable).
- How long the subject took to solve the goal.
- How much exploration was required.

Careful observation of test subjects would complement the limited access data we get on all of the site's regular visitors. Of course, we can also rely on intermediate measures such as encouraging users to fill out feedback forms and send e-mail messages.

4 Conclusion

This paper posed the challenge of using AI techniques to radically transform web sites from today's inert collections of HTML pages and hyperlinks to intelligent, evolving entities. Adaptive web sites can make popular pages more accessible, highlight interesting links, connect related pages, and cluster similar documents together. An adaptive web site can perform these self-improvements autonomously or advise a site's webmaster, summarizing access information and making suggestions. The improvements can happen in real-time as a visitor is navigating the site, or offline based on observations culled from many visitors.

This paper juxtaposed a number of disconnected projects from knowledge representation, machine learning, and user modeling that are investigating aspects of the problem. We believe that posing the challenge explicitly, in this paper, will help to cross-fertilize existing efforts and alert new researchers to the problem. Success in the next two years will have a broad and highly visible impact on the web and the AI community.

References

[Anantharaman *et al.*, 1990] T. Anantharaman, M. Campbell, and F. Hsu. Singular extensions: adding selectivity to brute-force searching. *Artificial Intelligence*, 43(1):99–109, 1990.

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

[Etzioni, 1996] O. Etzioni. Moving up the information food chain: softbots as information carnivores. In *Proc. 14th Nat. Conf. on AI*, 1996.

[Fernandez *et al.*, 1997] Mary Fernandez, Daniela Florescu, Jaewoo Kang, Alon Levy, and Dan Suciu. System demonstration - strudel: A web-site management system. In *ACM SIGMOD Conference on Management of Data*, 1997.

[Fink *et al.*, 1996] J. Fink, A; Kobsa, and A. Nill. User-oriented adaptivity and adaptability in the avanti project. In *Designing for the Web: Empirical Studies*, Microsoft Usability Group, Redmond (WA)., 1996.

[Kautz, 1987] H. Kautz. *A Formal Theory Of Plan Recognition.* PhD thesis, University of Rochester, 1987.

[Lesh and Etzioni, 1995] Neal Lesh and Oren Etzioni. A sound and fast goal recognizer. In *Proc. 14th Int. Joint Conf. on AI*, pp. 1704–1710, 1995.

[Luke *et al.*, 1997] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In *Proceedings of the First International Conference on Autonomous Agents*, 1997.

[Perkowitz and Etzioni, 1997] M. Perkowitz and O. Etzioni. Adaptive sites: Automatically learning from user access patterns. Technical Report UW-CSE-97-03-01, University of Washington, Department of Computer Science and Engineering, March 1997.

[Pollack, 1990] M. Pollack. Plans as complex mental attitudes. In P. Cohen, J. Morgan, and M. Pollack, eds., *Intentions in Communication*, pp. 77–101. MIT Press, Cambridge, MA, 1990.

[Thorpe, 1990] C. Thorpe, ed. *Vision and Navigation: the Carnegie Mellon Navlab.* Kluwer Academic Publishing, Boston, MA, 1990.