Planning-Based Control of Software Agents

Daniel S. Weld
Department of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350
weld@cs.washington.edu

Introduction

The exponential growth of the Internet has produced a labyrinth of databases and services. Almost any type of information is available somewhere, but most users can't find it, and even experts waste untold time searching for appropriate information sources. Many researchers argue that software agents will remedy the situation by acting as personal assistants and automatically accessing multiple sources, integrating information, and acting on the user's behalf.

Indeed, this vision is so widely accepted (at least in the AI community) that I won't dwell on it further. Instead, I'll describe some of the software robots we've built at the University of Washington, explain why I think planning is a crucial technology for their control, summarize the lessons we've learned by their construction, and suggest directions for future work in the area. Note that this paper is not intended to be a comprehensive survey of important softbot projects -- there are far too many interesting systems for me to describe them all. I focus on University of Washington projects, because I am most familiar with this body of work and because the Washington softbots have emphasized planning-based control.

Motivation

Softbots (software robots) are like physical robots in several respects (Etzioni 1993a). Both have effectors, but instead of motor-driven wheels and grippers, softbots affect the world with commands such as `ftp`, `telnet`, and `mail`. Instead of sensors like sonar, softbots perception is based on Internet facilities such as `gopher` and `netfind`. Given the metaphor of "Internet as an information super-highway," a softbot becomes the competent chauffeur who knows every road and quietly performs routine errands. If it is to embody the ideal cyber-chauffeur, a softbot must satisfy several interrelated criteria (Etzioni & Weld 1994):

- **Integrated:** Users dislike being forced to remember the details of particular databases or the wide and growing variety of Internet services and utilities in order to use them effectively. Instead, the softbot should provide an integrated interface to all such services.
- **Goal oriented.** A user should be able to state *what* he or she wants accomplished. The softbot should determine *how* and *when* to achieve that goal, then perform the actions without supervision.
- **Expressive.** If common goals are impossible to specify, then a goal-oriented softbot is of limited use. Users should be able to request tasks such as "Get all of Tate's technical reports that aren't already stored locally." The softbot should handle this goal even though the underlying services (*e.g.*, `ftp`) does not handle this combination of universal quantification and negation.
- **Cooperative:** Instead of issuing a passive-aggressive error message in response to incorrect or incomplete specifications, a softbot should collaborate with the user in order to build a reasonable request.
- **Customizable:** Softbots should adapt both to the environment and to their users. A combination of experience and direct requests from the user should guide this process.

While all five attributes are important, I focus on the issues of integration, goal-orientation, and expressiveness in the remainder of this paper. The next section argues that AI planning algorithms are an ideal means for implementing these attributes.

Planning-Based Control

Since a planner takes a high-level goal as input and outputs a sequence of actions that will achieve the goal, planners are a natural way to construct a goal-oriented interface. Because planners take a database of action descriptions as another input, they also satisfy the objective of integration. By encoding each Internet information source as an action, the planner automatically integrates the available databases and services, dynamically choosing the utilities which are appropriate for the goal, backtracking and subgoaling as necessary.

A planner is only useful in an interface if it operates quickly enough to provide snappy performance.

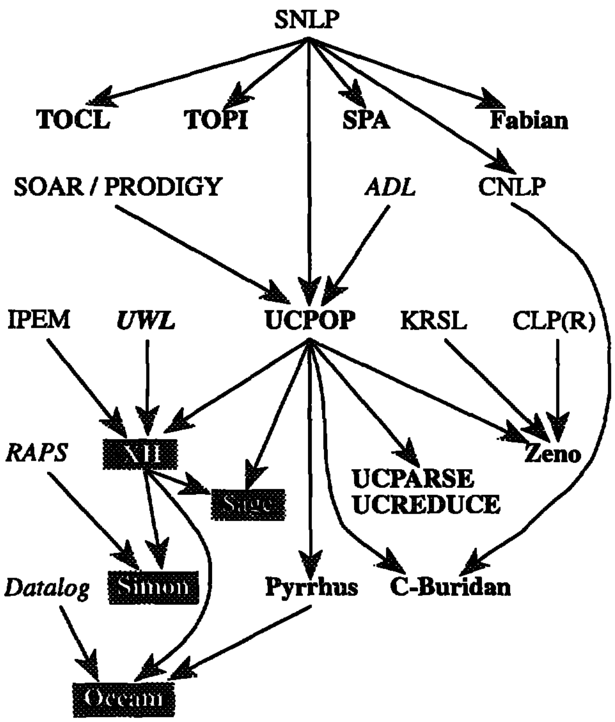

Figure 1: Partial family tree of planning algorithms (for clarity, many important planners are not shown). Arrows indicate significant intellectual ancestry; italics denotes an action language (as opposed to a planning system); boldface indicates development at the University of Washington. Black boxes surround planners that are used to control software agents.

Much of our planning research (figure 1) has been devoted to efficiency. The TOCL and TOPI planners helped show the potential power of the partial-order plan representation (Barrett & Weld 1994a). SPA introduced a new approach to case-based planning (Hanks & Weld 1995). The FABIAN planner extends the least-commitment approach to the choice of which action is used to support a subgoal (Friedman & Weld 1996). UCPARSE and UCREDUCE illustrate two alternative approaches for exploiting hierarchical task network information in the planning process (Barrett & Weld 1994b). In the future, we expect to adopt additional speedup learning and optimization techniques, *e.g.* (Minton 1990; Knoblock 1990; Etzioni 1993c; 1993b; Peot & Smith 1993; Smith & Peot 1993; Kambhampati & Chen 1993; Joslin & Pollack 1994; Veloso 1994; Ihrig & Kambhampati 1994; Schubert & Gerevini 1995; Joslin & Pollack 1995; Smith & Peot 1996).

In order to create an *expressive* interface, the softbot planner must handle a comprehensive goal language. Furthermore, the planner's action description language must be expressive in order to adequately represent a wide range of Internet utilities and services. The challenge of developing efficient planning algorithms capable of handling expressive languages has driven much of our planning research (figure 1). The UCPOP planner (Penberthy & Weld 1992; Weld 1994) accepts a large subset of the ADL action description language (Pednault 1989). UCPOP accepts universally quantified goals with nested quantifiers, conjunction, disjunction, and negation; action effects may be universally quantified and conditional.[1] From SOAR (Laird, Newell, & Rosenbloom 1987) via PRODIGY (Minton *et al.* 1989) we borrowed the idea of a metalevel control language for specifying search heuristics.

Several of our projects used UCPOP as a base planner with which to explore action languages of even greater expressiveness. Safety and tidiness constraints were discussed in (Weld & Etzioni 1994). The ZENO planner uses principles from logic programming (Jaffar *et al.* 1992) and operations research (Karloff 1991) to handle simultaneous actions involving continuous change and occurring over extended time intervals. ZENO supports metric constraints, exogenous events, domain axioms, and deadline goals with a sound and complete algorithm (Penberthy & Weld 1994; Penberthy 1993). PYRRHUS (Williamson & Hanks 1994; 1996) also handles deadline goals, but uses decision theoretic techniques to balance the cost of a plan against the degree of goal satisfaction (Haddawy & Hanks 1992). The BURIDAN planner adopted a probabilistic model of the world in which actions could have uncertain effects (Kushmerick, Hanks, & Weld 1995); C-BURIDAN (Draper, Hanks, & Weld 1994) extends this model with sensing actions and and approach to contingent action execution which builds on the approach of CNLP (Peot & Smith 1992).

A central problem in planner-based control of software agents is incomplete information (Olawsky & Gini 1990; Krebsbach, Olawsky, & Gini 1992). After all, no softbot could be familiar with the contents of *all* bulletin boards, FTP sites, and web pages. The possibility of incomplete information, means that an agent must distinguish between causation and observation. For example, it must differentiate between an action that erases the hard disk and one that detects such destruction! Since the process of information gathering is a crucial aspect of most applications of software agents, much of our work has focussed on the representation of sensors and planning algorithms that handle such representations. In theory the C-BURIDAN representation could be used, but the limitations of existing algorithms render probabilistic representations intractable for problems of Internet-sized scope. The

[1]Because of its portability and simplicity, UCPOP has been adopted by almost 100 sites worldwide. For information on acquiring the Common Lisp source code, see `http://www.cs.washington.edu/research/projects/ai/www/ucpop.html`

UWL representation (Etzioni *et al.* 1992) provides a simple method for encoding sensing actions and information gathering goals (also known as knowledge preconditions (Moore 1985; Morgenstern 1987)). Borrowing ideas from the IPEM planner (Ambros-Ingerson & Steel 1988), the XII planner integrates planning and execution using an extension of the UWL language that provides quantified sensing operations of the form "For all files in the directory, observe their name and their length." (Golden, Etzioni, & Weld 1994; 1996). The Internet softbot, RODNEY (Etzioni & Weld 1994), is controlled by XII as explained in the next section. Our most recent planner is called OCCAM (Kwok & Weld 1996); it builds upon the systems described above and borrows ideas from database theory (Ullman 1988; 1989; Rajaraman, Sagiv, & Ullman 1995; Levy, Srivastava, & Kirk 1995).

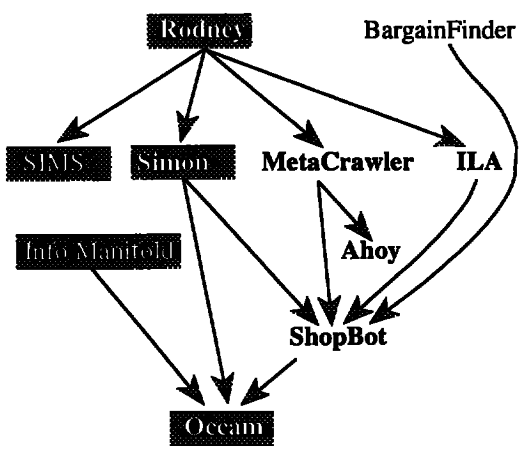

Figure 2: Partial softbot family tree (not intended as a comprehensive listing of software agent projects). Arrows indicate significant intellectual ancestry; boldface indicates development at the University of Washington. Black boxes surround agents that have general-purpose, deliberative control.

The Softbot Story

Figure 2 shows an abbreviated genealogy of our software agents. RODNEY, the patriarch, consists of four major modules — task manager (approximately 10% of the code), XII planner (25%), Internet resource models (30%), and model manager (25%) — in addition to miscellaneous support code (10%); see figure 3. 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. The XII planner was described briefly in the previous section; see (Golden, Etzioni, & Weld 1994; 1996) for more information. The Internet domain models provide RODNEY with background information about the Internet — primarily a set of approximately 100 action descriptions as encoded in an extension of UWL. The model manager is a specialized database that stores everything that RODNEY has observed about the world. The most novel aspect of the model manager is its capacity for local closed-world reasoning (Etzioni, Golden, & Weld 1994). 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 (Ginsberg 1987). 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 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, RODNEY 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 model manager performs fast inference on local closed-world information — if the user later specifies that the carrier must be United Airlines, then RODNEY need not access SABRE again. But if RODNEY 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.

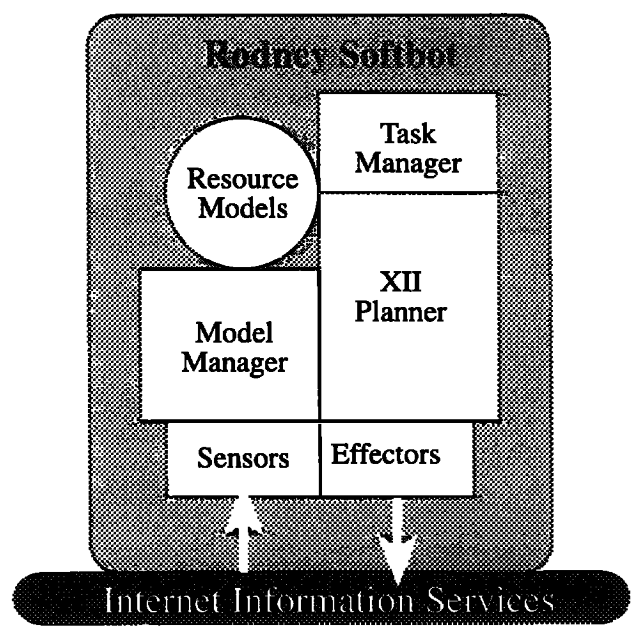

Figure 3: Architecture of the RODNEY softbot.

Although RODNEY was a success in most respects, there were two unsatisfying aspects to its operation: the unwieldy size of the UWL encodings of Internet services, and the difficulty of specifying search control knowledge in the metalevel control language that XII inherited from PRODIGY via UCPOP. Although the planner handled many goals effectively, it was more brittle than we liked. To some extent this was understandable, since RODNEY's model of roughly a hundred UNIX commands and Internet services made it one of the largest planning domain theories reported in the literature, comparable to the those of OPLAN (Currie & Tate 1991) and SIPE (Wilkins 1988a; 1988b; 1990).

In any case, we wanted to do even better. Research led us in several directions. One tack eschewed general purpose, planner-based control as being too difficult; the resulting METACRAWLER (Selberg & Etzioni 1995) is a special purpose softbot optimized to deal with multiple search engines in parallel. Arguably, it is a huge success, handling 40,000 hits a day; but is it AI? Perhaps the METACRAWLER is best thought of as enabling technology for subsequent AI research. For example, the AHOY homepage finder (Shakes, Langheinrich, & Etzioni 1996) uses a heuristic evaluation function to filter METACRAWLER output for URLs likely to represent the type of page sought by the user. The raw performance of the METACRAWLER enables an intriguing bootstrapping technique: user evaluations of AHOY performance provide high quality data for use as training data in a learning algorithm; when the learner produces an improved heuristic function, the improved accuracy attracts more users and hence more evaluations for an upward performance spiral.

Other projects pursued the dream of creating general-purpose softbots. Since one problem with RODNEY was the time-consuming process of hand coding resource models, the ILA learning system (Perkowitz & Etzioni 1995) addressed the task of automatically learning these models by computer-directed interaction with Internet sites. Motivated by the task popularized by BARGAINFINDER (Krulwich 1995), the SHOPBOT comparison shopping agent (Doorenbos, Etzioni, & Weld 1996) used ILA-style learning technology to autonomously develop models of electronic vendors. We also explored with the idea of procedural control of subgoaling behavior (in contrast with the SOAR-like metalevel declarative representation in RODNEY. The resulting system, which we called SIMON (Kwok *et al.* 1995), resembled RAPS (Firby 1987). Although SIMON was somewhat faster and a bit more predictable than RODNEY, it was just as hard (if not harder) to knowledge engineer. Encoding strategies for handling universally quantified goals was particularly time-consuming in our RAPS framework.

Meanwhile, at ISI, the SIMS project (Knoblock, Arens, & Hsu 1994) had developed a database-integration softbot that used planning-based control similar to that pioneered by XII. Like XII the SIMS planner (called SAGE (Knoblock 1995)) was based on UCPOP, but SAGE allows parallel execution which XII lacks. And at AT&T, researchers were developing the INFORMATION MANIFOLD (Levy, Srivastava, & Kirk 1995) which used an elegant and compact representation of information sources, stemming from database theory. In many respects this representation is less expressive than the extended UWL language used by RODNEY and SIMON; for example, it can not represent causal effects or world-state preconditions. On the other hand, since reduced expressiveness often translates into tractability, we have borrowed some of these ideas in the design of our OCCAM softbot. The OCCAM planner is fully implemented (Kwok & Weld 1996), and work proceeds on the execution system which uses the decision theoretic techniques developed in PYRRHUS (Williamson & Hanks 1994) to gather information in the most effective manner.

The Future

As the Internet environment grows ever more complex, users will demand increasingly flexible agents to assist them. While special purpose agents will always have a role, the future belongs to softbots with general-purposed deliberative control. However, in order to realize the dream of such powerful agents acting in a domain as vast and dynamic as the Internet, the planning community must continue to make fast research progress on a variety of fronts. Automatic construction of powerful search control knowledge, *e.g.* (Etzioni 1993b; Smith & Peot 1996), is a crucial challenge — especially for planners with expressive languages. Algorithms from constraint satisfaction and operations research should be integrated with modern planning representations (Penberthy & Weld 1994; Joslin & Pollack 1995; Kautz & Sleman 1996). New approaches for reasoning about incomplete information and sensing actions should be developed; local closed world reasoning algorithms (Etzioni, Golden, & Weld 1994) should be extended to handle worlds where exogenous change is common. Probabilistic approaches must be investigated to a greater extent. Tractable applications of decision theory have great promise for evaluating the value of information and determining the best information sources to query.

Much remains to be done, but with hard work we can ensure the ubiquity of planning technology in cyberspace. To paraphrase an ancient Chinese saying: *a planning-based softbot is worth a thousand shell-scripts.*

Acknowledgments

Oren Etzioni's vision converted me into a softboticist, and he collaborated on most of the systems described in this paper. Steve Hanks, also a valued collaborator, helped teach me much about planning. In the course of five years, well over a dozen people helped design, build, and test the softbots mentioned herein; I dare not try and list them all. I've learned from many, many people both here at Washington and around the world; I would like thank them all profusely. My research is funded in part by Office of Naval Research Grant N00014-94-1-0060, by National Science Foundation Grant IRI-9303461, by ARPA / Rome Labs grant F30602-95-1-0024, and by gifts from Apple Computer and Rockwell International Palo Alto Research.

References

Allen, J.; Hendler, J.; and Tate, A., eds. 1990. *Readings in Planning.* San Mateo, CA: Morgan Kaufmann.

Ambros-Ingerson, J., and Steel, S. 1988. Integrating planning, execution, and monitoring. In *Proc. 7th Nat. Conf. on A.I.*, 735–740.

Barrett, A., and Weld, D. 1994a. Partial order planning: Evaluating possible efficiency gains. *Artificial Intelligence* 67(1):71–112.

Barrett, A., and Weld, D. 1994b. Task-decomposition via plan parsing. In *Proc. 12th Nat. Conf. on A.I.* Available via FTP from pub/ai/ at ftp.cs.washington.edu.

Currie, K., and Tate, A. 1991. O-plan: the open planning architecture. *Artificial Intelligence* 52(1):49–86.

Dent, L.; Boticario, J.; McDermott, J.; Mitchell, T.; and Zabowski, D. 1992. A personal learning apprentice. In *Proc. 10th Nat. Conf. on A.I.*, 96–103.

Doorenbos, B.; Etzioni, O.; and Weld, D. 1996. A scalable comparison-shopping agent for the world-wide web. Technical Report 96-01-03, University of Washington, Department of Computer Science and Engineering. Available via FTP from pub/ai/ at ftp.cs.washington.edu.

Draper, D.; Hanks, S.; and Weld, D. 1994. Probabilistic planning with information gathering and contingent execution. In *Proc. 2nd Intl. Conf. on A.I. Planning Systems.*

Etzioni, O., and Weld, D. 1994. A softbot-based interface to the internet. *CACM* 37(7):72–76. See http://www.cs.washington.edu/research/softbots.

Etzioni, O.; Hanks, S.; Weld, D.; Draper, D.; Lesh, N.; and Williamson, M. 1992. An Approach to Planning with Incomplete Information. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning.* San Francisco, CA: Morgan Kaufmann. Available via FTP from pub/ai/ at ftp.cs.washington.edu.

Etzioni, O.; Golden, K.; and Weld, D. 1994. Tractable closed-world reasoning with updates. In *Proc. 4th Int. Conf. on Principles of Knowledge Representation and Reasoning*, 178–189. San Francisco, CA: Morgan Kaufmann.

Etzioni, O. 1993a. Intelligence without robots (a reply to brooks). *AI Magazine* 14(4). Available via anonymous FTP from pub/etzioni/softbots/ at cs.washington.edu.

Etzioni, O. 1993b. Acquiring search-control knowledge via static analysis. *Artificial Intelligence* 62(2):255–302.

Etzioni, O. 1993c. A structural theory of explanation-based learning. *Artificial Intelligence* 60(1):93–140.

Firby, R. J. 1987. An investigation into reactive planning in complex domains. In *Proc. 6th Nat. Conf. on A.I.*, 202–206.

Friedman, M., and Weld, D. 1996. Least-commitment action selection. In *Proc. 3rd Intl. Conf. on A.I. Planning Systems.*

Ginsberg, M., ed. 1987. *Readings in Nonmonotonic Reasoning.* San Mateo, CA: Morgan Kaufmann.

Golden, K.; Etzioni, O.; and Weld, D. 1994. Omnipotence without omniscience: Sensor management in planning. In *Proc. 12th Nat. Conf. on A.I.*, 1048–1054. Menlo Park, CA: AAAI Press.

Golden, K.; Etzioni, O.; and Weld, D. 1996. Planning with execution and incomplete information. Technical Report 96-01-09, University of Washington, Department of Computer Science and Engineering. Available via FTP from pub/ai/ at ftp.cs.washington.edu.

Haddawy, P., and Hanks, S. 1992. Representations for Decision-Theoretic Planning: Utility Functions for Dealine Goals. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning.*

Hanks, S., and Weld, D. S. 1995. A domain-independent algorithm for plan adaptation. *Journal of Artificial Intelligence Research* 319–360. Available via FTP from pub/ai/ at ftp.cs.washington.edu.

Ihrig, L., and Kambhampati, R. 1994. Derivation replay for partial-order planning. In *Proc. 12th Nat. Conf. on A.I.*

Jaffar, J.; Michaylov, S.; Stuckey, P.; and Yap, R. 1992. The CLP($\mathcal{R}$) Language and System. *ACM Transactions on Programming Languages and Systems* 14(3):339–395.

Joslin, D., and Pollack, M. 1994. Least-cost flaw repair: A plan refinement strategy for partial-order planning. In *Proc. 12th Nat. Conf. on A.I.*

Joslin, D., and Pollack, M. 1995. Passive and active decision postponement in plan generation. In *Proc. 2nd European Planning Workshop.*

Kambhampati, S., and Chen, J. 1993. Relative utility of EBG based plan reuse in partial ordering vs. total ordering planning. In *Proc. 11th Nat. Conf. on A.I.*

Kambhampati, S.; Knoblock, C.; and Yang, Q. 1995. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial order planning. *Artificial Intelligence* 76:167–238.

Karloff, H. 1991. *Linear Programming.* Boston: Birkhäuser.

Kautz, H., and Sleman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. Personal Communication.

Kautz, H.; Selman, B.; Coen, M.; Ketchpel, S.; and Ramming, C. 1994. An experiment in the design of software agents. In *Proc. 12th Nat. Conf. on A.I.*

Knoblock, C., and Levy, A., eds. 1995. *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments.* Stanford University: AAAI Press. To order a copy, contact sss@aaai.org.

Knoblock, C.; Arens, Y.; and Hsu, C.-N. 1994. Cooperating agents for information retrieval. In *Proceedings of the Second International Conference on Cooperative Information Systems.*

Knoblock, C. 1990. Learning abstraction hierarchies for problem solving. In *Proc. 8th Nat. Conf. on A.I.*, 923–928.

Knoblock, C. A. 1995. Planning, executing, sensing, and replanning for information gathering. In *Proc. 15th Int. Joint Conf. on A.I.*

Krebsbach, K.; Olawsky, D.; and Gini, M. 1992. An empirical study of sensing and defaulting in planning. In *Proc. 1st Intl. Conf. on A.I. Planning Systems*, 136–144.

Krulwich, B. 1995. Bargain finder agent prototype. Technical report, Anderson Consulting. `http://bf.cstar.ac.com/bf/`.

Kushmerick, N.; Hanks, S.; and Weld, D. 1995. An Algorithm for Probabilistic Planning. *Artificial Intelligence* 76:239–286.

Kwok, C., and Weld, D. 1996. Planning to gather information. Technical Report 96-01-04, University of Washington, Department of Computer Science and Engineering. Available via FTP from pub/ai/ at `ftp.cs.washington.edu`.

Kwok, C.; Goan, T.; Etzioni, O.; and Weld, D. 1995. Simon user guide. Technical report, University of Washington. `http://www.cs.washington.edu/homes/ctkwok/simon-manual.html`.

Laird, J.; Newell, A.; and Rosenbloom, P. 1987. SOAR: An architecture for general intelligence. *Artificial Intelligence* 33(1).

Levy, A. Y.; Srivastava, D.; and Kirk, T. 1995. Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems, Special Issue on Networked Information Discovery and Retrieval* 5 (2).

Maes, P., and Kozierok, R. 1993. Learning interface agents. In *Proceedings of AAAI-93.*

Maes, P. 1994. Agents that reduce work and information overload. *Comm. of the ACM* 37(7):31–40, 146.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proc. 9th Nat. Conf. on A.I.*, 634–639.

Minton, S.; Carbonell, J. G.; Knoblock, C. A.; Kuokka, D. R.; Etzioni, O.; and Gil, Y. 1989. Explanation-based learning: A problem-solving perspective. *Artificial Intelligence* 40:63–118. Available as technical report CMU-CS-89-103.

Minton, S. 1990. Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence* 42(2-3).

Moore, R. 1985. A Formal Theory of Knowledge and Action. In Hobbs, J., and Moore, R., eds., *Formal Theories of the Commonsense World.* Norwood, NJ: Ablex.

Morgenstern, L. 1987. Knowledge preconditions for actions and plans. In *Proceedings of IJCAI-87*, 867–874.

Olawsky, D., and Gini, M. 1990. Deferred planning and sensor use. In *Proceedings, DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control.* Morgan Kaufmann.

Pednault, E. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proc. 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, 324–332.

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

Penberthy, J., and Weld, D. 1994. Temporal planning with continuous change. In *Proc. 12th Nat. Conf. on A.I.*

Penberthy, J. 1993. *Planning with Continuous Change.* Ph.D. Dissertation, University of Washington. Available as UW CSE Tech Report 93-12-01.

Peot, M., and Smith, D. 1992. Conditional Nonlinear Planning. In *Proc. 1st Intl. Conf. on A.I. Planning Systems*, 189–197.

Peot, M., and Smith, D. 1993. Threat-removal strategies for partial-order planning. In *Proc. 11th Nat. Conf. on A.I.*, 492–499.

Perkowitz, M., and Etzioni, O. 1995. Category translation: Learning to understand information on the internet. In *Proc. 15th Int. Joint Conf. on A.I.*

Rajaraman, A.; Sagiv, Y.; and Ullman, J. 1995. Answering queries using templates with binding patterns. In *Principles of Database Systems.*

Schubert, L., and Gerevini, A. 1995. Accelerating partial order planners by improving plan and goal choices. In *Proceedings of the 7th IEEE Int. Conference on Tools with Artificial Intelligence.*

Selberg, E., and Etzioni, O. 1995. Multi-Service Search and Comparison Using the MetaCrawler. In *Proc. 4th World Wide Web Conf.*, 195–208. See `http://www.cs.washington.edu/research/metacrawler`.

Shakes, J.; Langheinrich, M.; and Etzioni, O. 1996. Ahoy: The home page finder. Technical report, University of Washington. See http://www.cs.washington.edu/research/ahoy.

Smith, D., and Peot, M. 1993. Postponing threats in partial-order planning. In *Proc. 11th Nat. Conf. on A.I.*, 500–506.

Smith, D., and Peot, M. 1996. Suspending recurison in causal link planning. In *Proc. 3rd Intl. Conf. on A.I. Planning Systems.*

Stefik, M. 1981a. Planning and meta-planning (MOLGEN: Part 2). *Artificial Intelligence* 14(2):141–169.

Stefik, M. 1981b. Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence* 14(2):111-139.

Tate, A. 1977. Generating project networks. In *Proc. 5th Int. Joint Conf. on A.I.*, 888–893.

Ullman, J. 1988. Database and knowledge-base systems. In *Database and knowledge-base systems*, volume 1. Computer Science Press.

Ullman, J. 1989. Database and knowledge-base systems. In *Database and knowledge-base systems*, volume 2. Computer Science Press.

Veloso, M. 1994. Flexible strategy learning: Analogical replay of problem solving episodes. In *Proc. 12th Nat. Conf. on A.I.*, 595–600.

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

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

Wilkins, D. 1988a. Causal reasoning in planning. *Computational Intelligence* 4(4):373–380.

Wilkins, D. E. 1988b. *Practical Planning.* San Mateo, CA: Morgan Kaufmann.

Wilkins, D. 1990. Can AI planners solve practical problems? *Computational Intelligence* 6(4):232–246.

Williamson, M., and Hanks, S. 1994. Optimal planning with a goal-directed utility model. In *Proc. 2nd Intl. Conf. on A.I. Planning Systems.*

Williamson, M., and Hanks, S. 1996. Flaw selection strategies for value-directed planning. In *Proc. 3rd Intl. Conf. on A.I. Planning Systems.*