From: AAAI Technical Report WS-02-02. Compilation copyright © 2002, AAAI (www.aaai.org). All rights reserved.

Pearl: A Mobile Robotic Assistant for the Elderly

Martha E. Pollack*
Laura Brown
Dirk Colbry
Cheryl Orosz
Bart Peintner
Sailesh Ramakrishnan
University of Michigan

Sandra Engberg
Judith T. Matthews
Jacqueline Dunbar-Jacob
Colleen E. McCarthy
University of Pittsburgh

Sebastian Thrun
Michael Montemerlo
Joelle Pineau
Nicholas Roy
Carnegie Mellon University

Abstract

The Nursebot project is a multi-disciplinary, multi-university effort aimed at developing mobile robotic assistants for the elderly. In this paper, we describe one such robot, Pearl. Pearl has two primary functions: (i) reminding people about routine activities such as eating, drinking, taking medicine, and using the bathroom, and (ii) guiding them through their environments. We provide a brief overview of the hardware platform, and we sketch the major software systems that enable Pearl to perform its two main functions. A prototype version of Pearl has been completely built, with all software implemented, and preliminary testing has been done at the Longwood Retirement Community in Oakmont, PA.

Introduction

The global trend of increasing longevity presents an enormous challenge to those engaged in developing technology to sustain independence and preserve quality of life among older adults. As more people live longer than ever before, the resulting demographic shift raises the profile of frailty and disability within the world's population (Davies 1999). In the United States, two out of every five dependents of persons between 18 and 64 years of age is an older adult (McDevitt & Rowe 2002), and one-third of community-residing, older adults indicate that they are severely limited by a disabling condition (McNeil 2001).

Though the vast majority of older adults live in the community, many reside with similarly frail relatives, or live alone with little or no outside support (AoA 2001). Family members are often widely dispersed and minimally involved in meeting the day-to-day needs of their elders. In-home services from community agencies are generally time-limited and, if not covered through insurance, prohibitively expensive for many older adults. Further, a pressing shortage of nursing personnel exists, particularly among nurses (Spratley *et al.* 2002) who might help older adults manage safely at home.

Frail and disabled older adults are at risk for hospitalization and premature institutionalization, due to the complex interplay among age-related deficits, manifestations and treatment of disease, and behavioral, social and economic factors (Maas *et al.* 2001). For example, cognitive impairment, including memory lapses and loss of executive function, may conspire with visual impairment and poverty to interfere with taking medication as prescribed. Urinary incontinence may result from bladder overactivity or loss of muscle tone or sphincter control and worsen when restricted joint movement makes getting to the bathroom difficult. Deconditioning that results from prolonged immobility may affect strength and balance, producing unsteadiness and predisposition for falls. Social isolation due to geographic location, lack of interpersonal contact, or psychopathology may correspond to poor eating habits, weight loss, and weakness.

Technology will never alleviate the full array of problems our burgeoning aged population faces, particularly those that require a human hand to touch, a human ear to listen, and human sensibilities to register empathy face-to-face. This paper describes a project in which we are designing and building robotic assistants that augment, rather than replace, human interaction. Of the many services that such a robot could provide (see (Engelberger 1999; Lacey & Dawson-Howe 1998)), our focus is on two tasks: (i) reminding people about routine activities such as eating, drinking, taking medicine, and using the bathroom, and (ii) guiding them through their environments. Our goal in this paper is to provide a picture of the overall system, and to give a sense of our vision of how it can help in the lives of elderly people. Due to space limitations, we cannot provide comprehensive technical details, but see our other papers, which describe particular aspects of the project (Baltus *et al.* 2000; Colbry, Peintner, & Pollack 2002; McCarthy & Pollack 2002; Pollack 2002; Pollack *et al.* 2002; Montemerlo *et al.* 2002; Pineau & Thrun 2002).

System Overview

The *Nursebot Project* was conceived in 1998 by a multidisciplinary team of investigators from three universities, consisting of researchers in both health care and computer science. The initial goal of the project was to develop mobile robotic assistants for elderly people living in their homes, particularly those with mild cognitive impairment. Over time, the goals have expanded to also include robotic assistants for elderly in other settings (particularly, assisted living and nursing homes), assistants for nurses caring for

*Corresponding Author: M. E. Pollack, Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109, pollackm@eecs.umich.edu

Figure 1: Pearl at the Longwood Retirement Community.

the elderly in these settings, and other sorts of (non-robotic) intelligent assistants for the elderly.

To date, we have developed two autonomous mobile robots, along with software systems to enable these robots to assist elderly people. Figure 1 shows Pearl, our current robot, interacting with elderly residents of the Longwood Retirement Community in Oakmont, PA, where we have conducted initial field tests. Pearl is equipped with a differential drive system, two on-board Pentium PCs, wireless Ethernet, SICK laser range finders, sonar sensors, microphones for speech recognition, speakers for speech synthesis, touch-sensitive graphical displays, actuated head units, and stereo camera systems. Particular care has been paid to the design of Pearl's visual appearance, especially its head unit.[1]

On the software side, Pearl features off-the-shelf autonomous mobile robot navigation systems, speech recognition and speech synthesis software, fast image capture and compression software for online video streaming, and face detection and tracking software. Additionally, Pearl includes software modules, described below, that support the primary tasks of providing reminders and assisting with navigation.

Cognitive Orthotic Functions

The first main function of our system is to serve as a cognitive orthotic, providing elderly people with reminders about their daily activities. The idea of using computer technology to enhance the performance of cognitively disabled people dates back nearly forty years (Englebart 1963). Early aids included talking clocks, calendar systems, and similar devices that were not very technologically sophisticated; yet many are still in use today. More recent efforts at designing cognitive orthotics have enabled reminders to be provided using the telephone (Friedman 1998), personal digital assistants (Dowds & Robinson 1996) and pagers (Hersh & Treadgold 1994). Work has also been done on improved modeling of clients' activities, notably in the work of Kirsch and Levine (Kirsch *et al.* 1987), and in the PEAT system (Levinson 1997). This latter system is a hand-held orthotic device that uses AI planning technology to model the client's plan, and provide visual and audible cues about its execution.

In the Nursebot project, our goal is to make principled decisions about what reminders to issue when, balancing several potentially competing objectives: (i) ensuring that the client is aware of activities she (the cleint) is expected to perform, (ii) increasing the likelihood that she will perform at least the required activities (such as taking medicine), (iii) avoiding annoying the client, and (iv) avoiding making the client overly reliant on the system. To attain these goals, the system must be flexible and adaptive, responding to the actions taken by the client. Consider, for instance, a forgetful, elderly client with urinary incontinence who is supposed to be reminded to use the toilet every three hours, and whose next reminder is scheduled for 11:00 a.m. Suppose that our robot system observes the client enter the bathroom at 10:40 a.m., and concludes that toileting has occurred. The system should not then issue a reminder at 11:00, as previous planned, but should instead set a later reminder. Again, in making this decision, flexibility is required. A strict three-hour interval may not be optimal: the client's favorite television program might be on from 1:30 to 2:00. In that case, it might be better to issue the reminder at 1:25, and provide a justfication that mentions the television program (e.g., "Mrs. Smith, Why don't you use the toilet now? That way I won't interrupt you during your show.")

The software component we have developed to provide the cognitive orthotic functions is called Autominder, and is depicted in Figure 2. As shown, Autominder has three main components: a Plan Manager (PM), which stores the client's plan of daily activities, and is responsible for updating it and identifying any potential conflicts in it; a Client Modeler (CM), which uses information about the client's observable activities to track the execution of the plan; and a Personal Cognitive Orthotic (PCO), which reasons about any disparities between what the client is supposed to do and what she is doing, and makes decisions about when to issue reminders.

To initialize the system, the caregiver for an elderly client inputs a description of the client's daily activities, as well as any constraints on, or preferences regarding, the time or manner of their performance. This plan may then be changed in one of four ways: (i) the client or caregiver may add new activities; (ii) the client or caregiver may modify or delete activities aleady in the plan[2] (iii) the client may execute one of the planned activities; or (iv) the simple pas-

[1]Pearl was largely designed and built by Greg Baltus.

[2]In later versions of the system, we will need to implement mechanisms that ensure that the client is allowed to change some activities, e.g., social engagements, but is blocked from modifying others, e.g., medicine-taking.

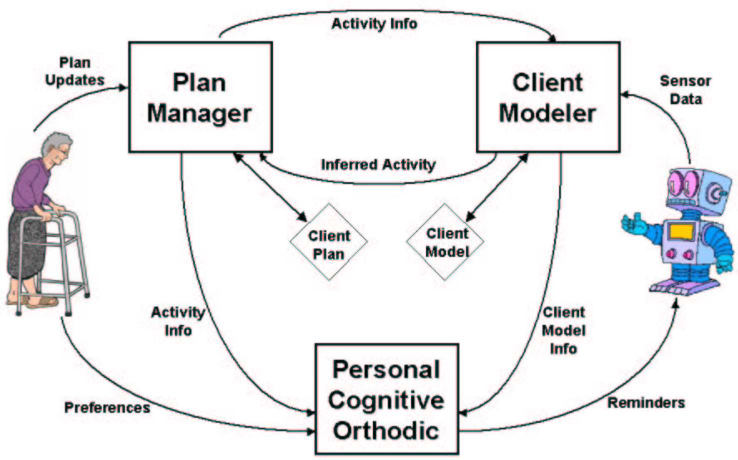

Figure 2: Autominder Architecture

sage of time may cause automatic changes to be made in the plan.[3] Whenever a change occurs, the PM updates the client plan, performing plan merging (Tsamardinos, Pollack, & Horty 2000) and constraint propagation as needed. To adequately represent client plans, it is essential to support a rich set of temporal constraints; we achieve this goal by modeling client plans as Disjunctive Temporal Problems (DTPs) and reasoning about them using efficient algorithms developed in our group (Tsamardinos 2001).

As the elderly client goes about her day, sensor information is gathered by the robot and sent to the CM, which uses this information to try to infer what activities the client is performing. For instance, going to the kitchen around the normal dinner time may indicate that the client is beginning dinner. If the likelihood is high that a planned activity is being executed, the CM reports this to the PM, which can then update the client's plans by recording the time of execution and propagating any affected constraints to other activities (e.g., if the client is supposed to take medicine no less than two hours after eating, the time for medicine-taking can be made more precise upon learning that the client is having dinner). The client model is represented with a new reasoning formalism called a Quantitative Temporal Bayes Net (QTBN), which we developed to handle the need both to reason about fluents and about probabilistic temporal constraints (Colbry, Peintner, & Pollack 2002).

The final component of Autominder is the PCO (McCarthy & Pollack 2002), which uses both the client plan and the client model to determine what reminders should be issued and when. The PCO identifies activities that may require reminders based on their importance and their likelihood of being executed on time as modeled in the CM. It also determines the most effective times to issue each required reminder, taking account of the expected client behavior, and any preferences explicitly provided by the client and the caregiver. Finally, the PCO provides justifications as to why particular activities warrant a reminder.

The PCO treats the generation of a reminder plan as a satisficing problem. It is relatively easy to create a reminder plan that is minimally acceptable: such a plan simply issues a reminder at the earliest possible start time of each activity. However, this plan is likely to do a poor job of satisfying the caregiver and client, and it does not attend at all to the objective of avoiding overreliance on the part of the client. Producing a higher-quality reminder plan is more difficult, as one must consider whether each reminder is really necessary, and also take account the client's expected behavior, her preferences, and interactions amongst planned activities. The PCO handles this problem by adopting a local-search approach called Planning-by-Rewriting (PbR) (Ambite & Knoblock 2001). It begins by creating the initial reminder plan as just suggested (reminders at the earliest possible time), and then performs local search, using a set of plan-rewrite rules to generate alternative candidate reminding plans. For example, the system contains a rule that deletes reminders for activities that have low importance and that are seldom forgotten by the client. Another rule spaces out reminders for activities for the same type of action: for instance, instead of issuing eight reminders in a row to drink water, the PCO will attempt to spread these reminders out through the day.

A prototype version of the Autominder has been fully implemented, in Java and Lisp for Wintel machines. It has been tested in the laboratory, as well as integrated with Pearl's other software components and included in a field test conducted at Longwood in June of 2001. We intend to conduct interviews later this year with caregivers and residents at Longwood in order to develop more detailed models of the daily plans of several residents, and then to field test a version of Autominder that encodes those plans.

Safe Navigation with Elderly People

The second major goal for our system is to help elderly people navigate their environments. This goal is particularly important for assisted living facilities, where nursing staff spend significant amounts of time escorting elderly residents from one location to another. The number of activities requiring navigation is large, ranging from regular daily events (e.g., meals), appointments (e.g., doctor appointments, physiotherapy, hair cuts), social events (e.g., visiting friends, cinema), to simply walking for the purpose of exercising. Many elderly people move at extremely slow speeds (e.g., 5 cm/sec), making the task of helping people around one of the most labor-intensive in assisted living facilities. Furthermore, the help provided is often not of a physical nature, as elderly people usually select walking aids over physical assistance by nurses, thus preserving some independence. Instead, nurses often provide important cognitive help, in the form of reminders, guidance and motivation, in addition to valuable social interaction.

[3]For instance, suppose that the client can eat lunch anytime between 11:00 a.m. and 1 p.m., but also expects her son to call at noon to check in with her. If we expect that lunch will take at least 20 minutes, and that it should not be interrupted by the phone call, then if lunch has not begun by 11:10 a.m., the plan will be changed to specify that lunch should not begin until after the phone call.

To enable Pearl to guide elderly people through their environments, we have extended previous navigation systems (Burgard *et al.* 1998; Thrun *et al.* 2000), by adding modules that are concerned specifically with interacting with people. The problem of locating people is that of determining their x-y-location relative to the robot. Previous approaches to people tracking in robotics were feature-based: they analyzed sensor measurements (images, range scans) for the presence of features (Gavrila 1999; Schultz *et al.* 2001) as the basis of tracking. In our case, the diversity of the environment mandated a different approach. Pearl detects people using map differencing: the robot learns a map, and people are detected by significant deviations from the map.

Mathematically, the problem of people tracking is a combined posterior estimation and model selection problem. Let N be the number of people near the robot. The posterior over the people's positions is given by

$$p(y_{1,t}, \ldots, y_{N,t} | z^t, u^t, m) \tag{1}$$

where $y_{n,t}$ with $1 \leq n \leq N$ is the location of a person at time t, z^t is the sequence of all sensor measurements, u^t is the sequence of all robot controls, and m is the environment map. To use map differencing, the robot has to know its own location. The location and total number of nearby people detected by the robot is clearly dependent on the robot's estimate of its own location and heading direction. Hence, Pearl estimates a posterior of the type:

$$p(y_{1,t}, \ldots, y_{N,t}, x^t | z^t, u^t, m) \tag{2}$$

where x^t denotes the sequence of robot poses (the path) up to time t. If N was known, estimating this posterior would be a high-dimensional estimation problem, with complexity that is between quadratic and cubic in N for Kalman filters (Bar-Shalom & Fortmann 1998) or exponential in N for particle filters (Doucet, de Freitas, & Gordon 2001). Neither of these approaches is, thus, applicable: Kalman filters cannot globally localize the robot, and particle filters would be computationally prohibitive.

Luckily, under certain reasonable conditions the posterior (2) can be factored into $N + 1$ conditionally independent estimates:

$$p(x^t | z^t, u^t, m) \prod_n p(y_{n,t} | z^t, u^t, m) \tag{3}$$

This factorization opens the door for a particle filter that scales linearly in N. Our approach is similar (but not identical) to the Rao-Blackwellized particle filter described in (Doucet *et al.* 2000). First, the robot path x^t is estimated using a particle filter, as in the Monte Carlo localization (MCL) algorithm (Dellaert *et al.* 1999) for mobile robot localization. However, each particle in this filter is associated with a set of N particle filters, each representing one of the people position estimates $p(y_{n,t} | z^t, u^t, m)$. These *conditional* particle filters represent people position estimates *conditioned* on robot path estimates—hence capturing the inherent dependence of people and robot location estimates. The data association between measurements and people is done using maximum likelihood, as in (Bar-Shalom & Fortmann 1998). Under the (false) assumption that this maximum likelihood estimator is always correct, our approach can be shown to converge to the correct posterior, and it does so with update time linear in N. In practice, we found that the data association is correct in the vast majority of situations.

Adaptive velocity is another area that we are working on. Initial experiments with fixed velocity led almost always to frustration on the people's side, in that the robot was either too slow or too fast. Thus, we plan to use estimates of a person's walking speed, to adapt the velocity of the robot, and thereby maintain roughly a constant distance to the person.

Of course, safety is a critical concern when navigating in the presense of elderly people. Our robot's laser range system measures obstacles 18 cm above ground, but is unable to detect any obstacles below or above this level. In the assisted living facilities, we found that people are easy to detect when standing or walking, but hard when on chairs. Thus, the risk of accidentally hitting a person's foot due to poor localization is particularly high in densely populated regions such as the dining areas.

Following an idea in (Burgard *et al.* 1998), we restricted the robot's operation area to avoid densely populated regions, using a manually augmented map of the environment. To stay within its operating area, the robot needs accurate localization, especially at the boundaries of this area. While our approach yields sufficiently accurate results on average, it is important to realize that probabilistic techniques never provide hard guarantees that the robot obeys a safety constraint. To further decrease the risk of a dangerous encounter, we augmented the robot localization particle filter by a sampling strategy that is sensitive to the increased risk in the dining areas (see (Poupart, Ortiz, & Boutilier 2001; Thrun, Langford, & Verma 2002)). By generating samples in high-risk regions, we minimize the likelihood of being mislocalized in such regions, or worse, the likelihood of entering prohibited regions undetected. Tests involving real-world data collected during robot operation show that the robot was well-localized in high-risk regions.

Control Architecture

High-level control of Pearl is required to arbitrate amongst information gathering and performance-related actions, as well as to negotiate the different goals generated by the different specialized modules. High-level robot control has been a popular topic in AI, and decades of research have led to a collection of well-studied architectures (e.g., (Arkin 1998; Brooks 1985; Gat 1996)). However, existing architectures rarely take uncertainty into account during planning. Consequently, we use a hierarchical variant of a partially observable Markov decision process (POMDP) (Kaelbling, Littman, & Cassandra 1998) as Pearl's high-level control architecture. Initial experiments have provided solid evidence that the consideration of uncertainty leads to measurably better control strategies at this level, due to the significant levels of noise in robot perception, which arise in Pearl both from the navigation sensors (e.g., the laser rangefinder) and the interaction sensors (e.g., speech recognition and the touchscreen).

Pearl's control decision is based on the full probability distribution generated by the state estimator, such as in

State features	*Feature values*
RobotLocation	home, room, physio
UserLocation	room, physio
UserPresent	yes, no
ReminderGoal	none, physio, vitamin, checklist
UserMotionGoal	none, toPhysioWithRobot
UserInfoGoal	none, wantTime, wantWeather
Observation features	*Feature values*
Speech	yes, no, time, weather, go, unknown
Touchscreen	t_yes, t_no, t_time, t_weather, t_go
Laser	atRoom, atPhysio, atHome
Reminder	g_none, g_physio, g_vitamin, g_checklist

Table 1: State Variables for the Nursebot Domain

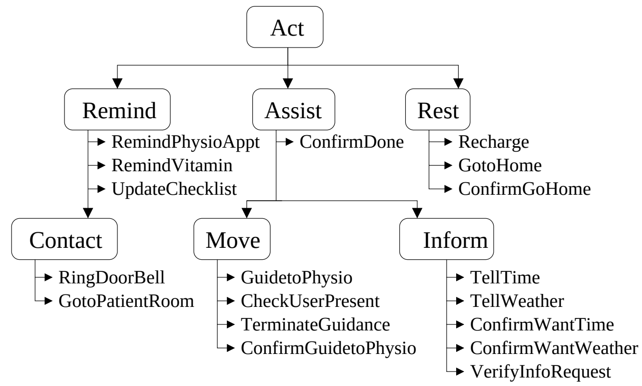

Figure 3: Sample Action Hierarchy

Equation (2) above. In our application, this distribution includes a number of multi-valued probabilistic state and goal variables, shown in Table 1, which encode information about the robot's location, the client's location, the existence of a reminder goal, observed input from the interaction sensors, and so on. Altogether, 576 distinct states are represented. In response, Pearl can select from 19 distinct actions, falling into three broad categories: communication actions (e.g., issue a reminder, check that the client is present, tell the client the time or the weather), movement actions (e.g., guide the client from one location to another), and miscellaneous actions (e.g., recharge the battery, do nothing). Each discrete action invokes a well-defined sequence of operations on the part of the robot; for instance, the action of telling the client the weather maps to *SpeechSynthesis="Tomorrow's weather should be sunny, with a high of 80."*.)

Unfortunately, POMDPs of the size required for the Nursebot application are an order of magnitude larger than today's best exact POMDP algorithms can tackle (Kaelbling, Littman, & Cassandra 1998). However, this application yields a highly structured POMDP, where certain actions are only applicable in certain situations. To exploit this structure, we developed a *hierarchical* version of POMDPs, which breaks down the decision making problem into a collection of smaller problems that can be solved more efficiently. Our approach is similar to the MAX-Q decomposition for MDPs (Dietterich 1998), but defined over POMDPs.

The basic idea of the hierarchical POMDP is to partition the action space—not the state space, since the state is not fully observable—into smaller chunks. Therefore the cornerstone of our hierarchical algorithm is an *action hierarchy*. Figure 3 illustrates an action hierarchy used on Pearl.

Formally, an action hierarchy is a tree, where each leaf is labeled by an action from the target POMDP problem's action set. Each *primitive action* $a \in A_0$ must be attached to at least one leaf (e.g. *RingDoorBell*, *GotoPatientRoom*, etc.) In each internal node (shown as circles in figure 3) we introduce an *abstract action*. Each of these provides an abstraction of the actions in the nodes directly below it in the hierarchy (e.g. *Contact* is an abstraction of *RingDoorBell* and *GotoPatientRoom*.)

A key step towards hierarchical problem solving is to use the action hierarchy to translate the original full POMDP task into a collection of smaller POMDPs. The goal is to achieve a collection of POMDPs that individually are smaller than the original POMDP, yet collectively define a complete policy. Thus, given an internal node $\bar{a}_i$ in the action hierarchy, we define a corresponding subtask $\mathbf{P}_i$. The subtask is a well-defined POMDP composed of:

- a state space S_i, identical to the full original state space S_0;
- an observation space Ω_i, identical to the full original observation space Ω_0;
- an action space A_i, containing the children nodes (both *primitive* and *abstract*) immediately below $\bar{a}_i$ in the subtask.

For example, the action hierarchy in figure 3 divides the problem into seven subtasks, where subtask P_{remind} has action set: A_{remind}=$\{a_{RemindPhysioAppt}, a_{RemindVitamin}, a_{UpdateChecklist}, \bar{a}_{Contact}\}$, and so on.

Once the action hierarchy has been defined, we can independently optimize an independent policy for each subtask, such that we obtain a collection of corresponding *local policies*, which taken as a whole, constitute a global policy for action. During execution, the controller then simply monitors the state of the system (i.e., calculates the posterior) and looks up the appropriate control in the local policy set (Pineau & Thrun 2002).

Conclusion

We have described Pearl, a mobile robot system being designed to assist elderly people in navigating their daily activities and their environment. Early prototype versions of our system have been field tested in a residential retirement community, and we have a number of additional experiments planned for that setting. While the project we are describing is ambitious, it is the consensus of our team that success will be dependent upon melding knowledge of technology with knowledge of human behavior, aging, and disability.

Acknowledgments

Primary support for the Nursebot project has come from the National Science Foundation (IIS-0085796). We would like to thank the caregivers and residents of the Longwood Retirement Community for allowing us to study our robot systems at their facility.

References

Ambite, J. L., and Knoblock, C. A. 2001. Planning by rewriting. *Journal of Artificial Intelligence Research* 15:207–261.

2001. A profile of older americans: 2000. U. S. Department of Health and Human Services, Washington, D. C.

Arkin, R. 1998. *Behavior-Based Robotics*. Boston, MA: MIT Press.

Baltus, G.; Fox, D.; Gemperle, F.; Goetz, J.; Hirsch, T.; Margaritis, D.; Montemerlo, M.; Pineau, J.; Roy, N.; Schulte, J.; and Thrun, S. 2000. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment*.

Bar-Shalom, Y., and Fortmann, T. E. 1998. *Tracking and Data Association*. Academic Press.

Brooks, R. 1985. A robust layered control system for a mobile robot. Technical Report AI memo 864, Massachusetts Institute of Technology, Cambridge, MA.

Burgard, W.; Cremers, A.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1998. The interactive museum tour-guide robot. In *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*.

Colbry, D.; Peintner, B.; and Pollack, M. E. 2002. Execution monitoring with quantitative temporal bayesian networks. In *Proceedings of the 6th International Conference on AI Planning and Scheduling*.

Davies, A. M. 1999. Aging and health in the 21st century. In *Aging and Health: A Global Challenge for the Twenty-First Century*. Geneva, Switzerland: World Health Organization.

Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Dietterich, T. 1998. The maxq method for hierarchical reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Doucet, A.; de Freitas, N.; Murphy, K.; and Russell, S. 2000. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 176–183.

Doucet, A.; de Freitas, J.; and Gordon, N., eds. 2001. *Sequential Monte Carlo Methods In Practice*. New York: Springer Verlag.

Dowds, M. M., and Robinson, K. 1996. Assistive technology for memory impairment: Palmtop computers and TBI. In *Braintree Hospital Traumatic Brain Injury Neurorehabilitation Conference*.

Engelberger, G. 1999. Services. In Nof, S. Y., ed., *Handbook of Industrial Robotics*. John Wiley and Sons, 2nd edition. chapter 64., 1201–1212.

Englebart, D. C. 1963. A conceptual framework for the augmentation of man's intellect. In *Vistas in Information Handling*. Spartan Books. 1–13.

Friedman, R. H. 1998. Automated telephone conversation to assess health behavior and deliver behavioral interventions. *Jounral of Medical Systems* 22:95–101.

Gat, E. 1996. Esl: A language for supporting robust plan execution in embedded autonomous agents. In *Working notes of the AAAI Fall Symposium on Plan Execution*. Boston, MA: AAAI.

Gavrila, D. M. 1999. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU* 73(1):82–98.

Hersh, N. A., and Treadgold, L. 1994. Neuropage: The rehabilitation of memory dysfunction by prosthetic memory and cueing. *NeuroRehabilitation* 4:187–197.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99–134.

Kirsch, N.; Levine, S. P.; Fallon-Kureger, M.; and Jaros, L. A. 1987. The microcomputer as an 'orthotic' device for patients with cognitive deficits. *Journal of Head Trauma Rehabilitation* 2:77–86.

Lacey, G., and Dawson-Howe, K. 1998. The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems* 23:245–252.

Levinson, R. 1997. Peat–the planning and execution assistant and trainer. *Journal of Head Trauma Rehabilitation*.

Maas, M. L.; Buckwalter, K. C.; Hardy, M. D.; Tripp-Rimer, T.; Titler, M. G.; and Specht, J. P. 2001. *Nursing Care of Older Adults: Diagnoses, Outcomes, and Interventions*. St. Louis, MO: Mosby, Inc.

McCarthy, C. E., and Pollack, M. E. 2002. A plan-based personalized cognitive orthotic. In *Proceedings of the 6th International Conference on AI Planning and Scheduling*.

McDevitt, T. M., and Rowe, P. 2002. The united states in international context: 2000. Technical Report Current Population Reports, C2KBR/01-11, U.S. Census Bureau, Washington, D. C.

McNeil, J. 2001. Americans with disabilities. Technical Report Current Population Reports, P70-73, U. S. Census Bureau, Washington, D. C.

Montemerlo, M.; Pineau, J.; Roy, N.; and Thrun, S. 2002. Experiences with a mobile robotic guide for the elderly. In *18th National Conference on Artificial Intelligence*.

Pineau, J., and Thrun, S. 2002. High-level robot behavior control using pomdps. In *AAAI-02 Workshop on Cognitive Robotics*.

Pollack, M. E.; McCarthy, C. E.; Ramakrishnan, S.; Tsamardinos, I.; Brown, L.; Carrion, S.; Colbry, D.; Orosz, C.; and Peintner, B. 2002. Autominder: A planning, monitoring, and reminding assistive agent. In *7th International Conference on Intelligent Autonomous Systems*.

Pollack, M. E. 2002. Planning technology for intelligent cognitive orthotics. In *Proceedings of the 6th International Conference on Automated Planning and Scheduling*.

Poupart, P.; Ortiz, L.; and Boutilier, C. 2001. Value-directed sampling methods for monitoring POMDPs. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*.

Schultz, D.; Burgard, W.; Fox, D.; and Cremers, A. 2001. Tracking multiple moving targets with a mobile robot using particle filters and statistitical data association. In *Proceedings of ICRA-2001*.

Spratley, E.; Johnson, A.; Sochalski, J.; Fritz, M.; and Spencer, W. 2002. The registered nurse population: Findings from the 2000 national sample survey of registered nurses. Health Resources and Service Administration, Washington, D. C.

Thrun, S.; Beetz, M.; Bennewitz, M.; Burgard, W.; Cremers, A.; Dellaert, F.; Fox, D.; Hähnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; and Schulz, D. 2000. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research* 19(11):972–999.

Thrun, S.; Langford, J.; and Verma, V. 2002. Risk sensitive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press.

Tsamardinos, I.; Pollack, M. E.; and Horty, J. F. 2000. Merging plans with quantitative temporal constraints, temporally extended actions, and conditional branches. In *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling*.

Tsamardinos, I. 2001. *Constraint-Based Temporal Reasoning Algorithms with Applications to Planning*. Ph.D. Dissertation, University of Pittsburgh Intelligent Systems Program.