Autominder: A Planning, Monitoring, and Reminding Assistive Agent

Martha E. Pollack* Colleen E. McCarthy† Ioannis Tsamardinos‡
Sailesh Ramakrishnan* Laura Brown* Steve Carrion* Dirk Colbry*
Cheryl Orosz* Bart Peintner*

*Artificial Intelligence Laboratory
University of Michigan
pollackm@umich.edu

†Department of Computer Science
University of Pittsburgh

‡Department of Biomedical Informatics
Vanderbilt University

Abstract.

The percentage of elderly people in the population is increasing at a phenomenal rate [14]. A significant challenge faced by many elderly is a decline in cognitive functioning, particularly in memory. In this paper, we describe Autominder, an automated agent designed to serve as a "cognitive orthotic", assisting an elderly client in carrying out the required activities of daily life (ADLs), by providing her with timely and appropriate reminders. In generating these reminders, the goal is to balance three objectives: (i) maximizing the client's compliance in performing ADL's; (ii) maximizing the level of caregiver and client satisfaction with the system; and (iii) avoiding making the client overly reliant on the system. Towards these ends, Autominder stores and updates plans representing a client's ADLs, tracks their execution, learns the typical behavior of the client with regard to the execution of these plans, and provides select reminders of the activities to be performed. Autominder is being designed as part of the Initiative on Personal Robotic Assistants for the Elderly [12], a project aimed at developing robotic systems to assist elderly persons with memory impairment.

1 Introduction

The percentage of elderly people in the population is increasing at a phenomenal rate in the United States [14], as well as in many other parts of the world. Indeed, the number of people residing in nursing homes in the U.S. is projected to double or triple by 2030. It has been shown that the quality of life for people remaining in their own homes is generally better than for those who are institutionalized [17]; moreover, the cost for institutional care can be much higher than the cost of care for a patient at home. Unfortunately, a significant challenge faced by many elderly people is a decline in cognitive functioning, particularly in memory. Such a decline can make it difficult for someone to organize and regularly perform their necessary activities of daily living (ADLs), such as taking medicine correctly, eating, drinking water, toileting, performing physical exercises (e.g., "Kegel" bladder exercises), performing routine hygiene, engaging in recreational activities (e.g., watching television, attending a Bingo game), and going to medical appointments.[1] This inability to adequately perform ADLs can necessitate institutionalization.

In this paper, we describe Autominder, an automated agent designed to serve as a "cognitive orthotic", assisting an elderly client in carrying out the required activities of daily life by providing her with timely and appropriate reminders. In generating these reminders, the goal is to balance three objectives: (i) maximizing the client's compliance in performing ADL's; (ii) maximizing the level of satisfaction with the system of both the client and the caregiver(s); and (iii) avoiding making the client overly reliant on the system and possibly decreasing, rather than increasing, her independence. Towards these ends, Autominder stores and updates plans representing a client's ADLs, tracks their execution, learns the typical behavior of the client with regard to the execution of these plans, and provides carefully chosen and timed reminders of the activities to be performed. Autominder relies on a number of AI techniques, including interleaved planning and execution, sophisticated temporal reasoning, and reasoning under uncertainty.

Autominder is being designed as part of the Initiative on Personal Robotic Assistants for the Elderly (Nursebot)[12], a multi-university collaborative project.[2] The initial focus of this initiative is the design of an autonomous robot, currently called Pearl, that will "live" in the home of an elderly person. Autominder is a central element of Pearl's software. Several prototype versions of Autominder have been fully implemented in Java and Lisp. Although the most recent version has not yet been installed on Pearl, an earlier version was used in an exploratory field test with elderly users in June, 2001.

In the next section, we provide a thorough overview of Autominder's architecture, and of the existing and novel AI techniques we are using. Section 3 briefly discusses the issue of the kinds of platforms–robotic or software–on which Autominder might be installed. Section 4 describes related work on cognitive orthotics, and finally, Section 5 summarizes and points to ongoing and future work on this topic.

2 Autominder Architecture

Autominder grew out of our earlier work on plan management, in particular, the Plan Management Agent (PMA), a prototype intelligent calendar tool [15]. PMA consists primarily of a plan manager, a system that stores a client's plans, updating them as the client adds, deletes, or modifies constraints on those plans, and/or executes actions in them. A central task for PMA is to ensure that there are no conflicts amongst the client's plans, instead suggesting alternative ways to resolve potential conflicts. An extension of PMA's main component now serves as the Plan Manager (PM) for Autominder. There are two additional components essential to Autominder: a Client Modeler(CM) and a Personal Cognitive Orthotic (PCO). The overall architecture is illustrated in Figure 1. What is not apparent in the figure is that the system is event-driven and all communication between components is routed through a message-handling component.

---

[1] In fact, the list of activities we are covering extends beyond the set usually included under the heading of ADLs. We should also note that in the early versions of the Autominder, we are not directly issuing reminders about medicine-taking, due to safety concerns: we want to ensure the correctness of Autominder before seeking FDA approval to include medicine reminders.

[2] The initiative includes researchers from the University of Pittsburgh School of Nursing and Department of Computer Science, Carnegie Mellon University Robotics Institute and Human-Computer Interaction Department, and the University of Michigan Department of Electrical Engineering and Computer Science.

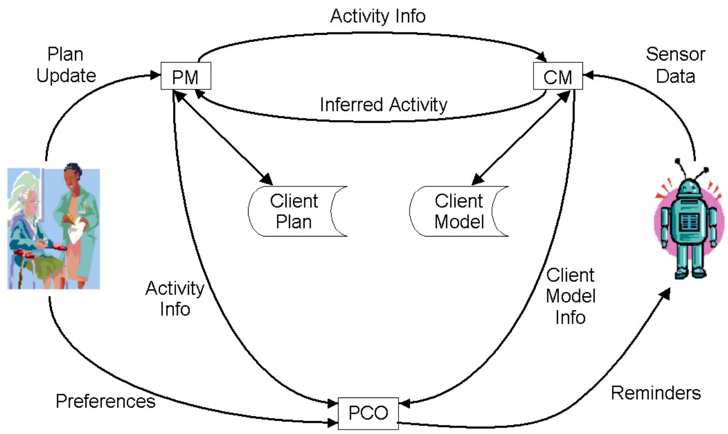

Figure 1: Autominder Architecture (simplified: Message Handler Omitted)

However, to make the flow of information clearer, we have omitted this component from the diagram, and just show the intended source and destination of each type of message.

In Autominder, the caregiver initially inputs a description of the activities the client is supposed to perform, as well as any constraints on, or preferences regarding, the time or manner of their performance. Subsequently, updates to the plan (e.g., a new doctor's appointment) can be made by a caregiver and, with certain restrictions, the client herself. Plan information flows directly to the PM, which, like PMA, checks for consistency and provides ways of resolving potential conflicts (e.g., using the toilet before leaving for the doctor's office).

Pearl, the robot on which Autominder is installed, has various sensors–camera, microphone, infrared, etc. - and it sends sensory information to the CM. Note that the pixels-to-predicates problem is solved by software outside of the Autominder: the Autominder receives reports of the form "client went to kitchen" or "toilet flush heard". The CM uses the sensor information, along with the client's plan itself, to infer whether there is an indication that a planned activity has been initiated or has ended (e.g., 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). Over time, the CM also constructs a model of the client's typical plan execution patterns (e.g., that the client usually remembers to take medicine in the morning, but frequently forgets in the evening). It is important to distinguish between the *client plan*, which models the activities that the caregiver would like the client to perform and is maintained by the PM, and the *client model*, which models the system's expectations of what the client has done and will do.

The final component of Autominder is the PCO, which uses both the client plan and the client model to determine what reminders should be issued and when. Ultimately, the PCO will also make use of information provided by the caregiver or client about their preferences as to how and when the activities should be executed, consistent with the requirements of the plan.

2.1 The Plan Manager

The primary job of the Plan Manager (PM) is to maintain an up-to-date model of the plan (the ADLs) that the client should execute. Initially, a routine daily plan is submitted to the PM. This plan may then be changed in one of three ways: (i) by the addition of new activities [3]; (ii) by the modification or deletion of (constraints on) activities already in the plan; (iii) by the execution of one of the planned activities. In the first two cases, PM performs plan merging [21, 8, 20, 19]: to ensure that the change does not introduce a conflict. In the third case, it propagates the constraints affected by activity execution, as described in the example above.

To adequately represent the client plans, it is essential to support a rich set of temporal constraints: for example, we may need to express that the client should take a medication within 15 minutes of waking, and then eat breakfast between 1 and 2 hours later. We model client plans as Disjunctive Temporal Problems (DTP) [13, 18] and use an efficient algorithm for checking their consistency, which we developed in [19]. The DTP is an expressive framework for temporal reasoning problems that extends the well-known Simple Temporal Problem (STP) [5] by allowing disjunctions, and the Temporal Constraint Satisfaction Problem (TCSP) [*ibid.*] by removing restrictions on the allowable disjunctions. Formally, a DTP is defined to be a pair $<V, C>$, where

- V is a set of variables (or nodes) whose domains are the real numbers, and
- C is a set of disjunctive constraints of the form: $C_i : x_1 - y_1 \leq b_1 \vee \ldots \vee x_n - y_n \leq b_n$, such that x_i are y_i are both members of V, and b_i is a real number.

A solution to a DTP is an assignment to each variable in V such that all the constraints in C are satisfied. If a DTP has at least one solution, it is *consistent*.

Within the PM, we assign a pair of DTP variables to each activity in the client's plan: one variable represents the start time of the activity, while the other represents its end time. We can easily encode a variety of constraints, including absolute times of events, relative times of events, and event durations, and can also express ranges for each of these.

To propagate new constraints and to check for consistency, the PM uses our Epilitis system [19]. The approach to DTP solving taken in the literature has been to convert the original problem to one of selecting one disjunct, $x_j - y_j \leq b_j$, from each constraint $C_i \in C$, and then checking that the set of selected disjuncts forms a consistent STP. Checking the consistency of and finding a solution to an STP can be performed in polynomial time using shortest-path algorithms [5]. The computational complexity in DTP solving derives from the fact that there are exponentially many sets of selected disjuncts that may need to be considered; the challenge is to find ways to efficiently explore the space of disjunct combinations. This has been done by casting the disjunct selection problem as a constraint satisfaction processing (CSP) problem [18, 13] or a satisfiability (SAT) problem [1]. Epilitis combines and extends the previous approaches, in particular by adding no-good learning, and achieves a speed-up of two orders of magnitude on a range of benchmark problems [19]. For typical problems we have so far studied in the Autominder domain, performance is well within the acceptable range, typically taking less than 10 seconds.

[3] Note that the PM includes a library of precomputed methods for common activities, so that information need only be provided about "top-level" activities, such as going to a doctor's appointment. Lower level activities, such as arranging for transportation, will then automatically be inserted in the plan.

2.2 Client Modeler

The second major component of Autominder is the Client Modeler (CM). As the Autominder client goes about her day, sensor information is sent to the CM. The CM is then responsible for two tasks: (i) inferring what planned activities the client has performed, given sensor data; and (ii) learning a model of the client's expected behavior. These tasks are synergistic, in that the client model developed is used in the inference task, while the results of the inference are used to update the model.

The client's expected behavior is represented with a new reasoning formulism called a Quantitative Temporal Dynamic Bayes Net (QTDBN). Essentially, a QTDBN combines a standard Bayes net which reasons about all temporal aspects of the client's activities, and a dynamic Bayes net (DBN) which reasons about the activities currently being executed. Together, they represent an entire day of activities. Nodes in each time slice of the DBN are random variables representing all of the following:

1. the incoming sensor data (e.g., client has moved to kitchen);
2. the actual execution of planned activities (e.g., client has started breakfast); and
3. whether a reminder for each activity has already been issued.

Initially, the model is derived from the client plan, by making two assumptions: first, that all activities in the plan will, with high probability, be executed by the client without reminders within the time range specified in the plan, and second, that the actual time of an activity can be described by a uniform probability density function over the range associated with that activity.

The CM uses sensor data and the current time to update the model. Each time sensor data arrives, the CM performs Bayesian update. If an activity execution node's probability rises above a threshold, the activity is believed to have occurred, and the CM notifies the rest of the system.

Over time, the CM should revise its model of the client's expected behavior. As suggested above, it might learn that the client usually remembers on her own to take medicine in the morning, but forgets in the evening–or it might learn that if the client eats breakfast early, she usually eats lunch early in the allowable lunch period also. By default, the CM creates its model based solely on the client plan created by the PM. For example, if the plan states that lunch must be eaten 3-4 hours after breakfast, the CM will encode that information in the probability table of the EatLunch action. Over time, the CM may learn that this relation does not hold when the client eats breakfast before 7am. The CM can then adjust the probability table to encode both the original rule and the learned exception.

2.3 Personalized Cognitive Orthotic

We have described how Autominder stores and updates the client's plan, tracks its execution, and learns the client's typical behavior patterns. We now describe the Personalized Cognitive Orthotic (PCO), the system component that decides what reminders to issue and when. The PCO identifies those 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: it simply involves issuing a reminder at the earliest start time of every activity. However, such a 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: not only does such a plan need to take account of whether a reminder is really necessary, but it must 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) [3, 2]. 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. Note that if the resulting reminders would violate any constraints in the client plan, then it will not be considered further. Rules may also be domain dependent, encoding specific preferences of the client or the caregiver, e.g., finish drinking all water by 5pm if possible.

The PCO eliminates any plan that does not contain reminders for all activities that are mandatory for the safety and well-being of the client, such as doctor's appointments and dietary requirements. Beyond that, the ascribed quality of a reminder schedule will be increased if the reminder times take account of the expected and preferred times of execution; if the schedule includes a single reminder for two or more activities that may overlap temporally and that share preconditions; if potential conflicts among activities have been identified and avoided; if reminders are generally separated in time rather than clustered into a short time period; and if reminders are not included for activities that have already been initiated.

The PCO is also designed to enable the generation of justifications for reminders. Justifications are motivated by the hypothesis that client adherence to plans may be improved when the reasoning behind the existence and timing of a reminder is provided. For example, a reminder of the form "If you take your medicine now, you will not have to do it in the middle of your show," may be more compelling than the simple message "Time for medicine." In generating a justification for a reminder, PCO can make use of the underlying client plan, the preferences of the caregiver and the client, and the particular rewrite rules used in creating the current reminder plan.

3 Autominder on the Robot Platform

A reasonable question to ask is whether a mobile robot is an appropriate platform for a cognitive orthotic; competing alternatives range from hand-held devices, to traditional desktop or laptop computers, to "intelligent houses" with multiple sensors [9]. We see several potential advantages to the use of mobile robots. Handheld devices and desktop/laptop computers have impoverished sensing capabilities and little to no reminding capabilities; moreover, handheld devices may be inappropriate for the targeted class of users, who may have a tendency to misplace them. While intelligent houses can perform sophisticated sensing, they are expensive to build, and elderly people may not want to move from the homes in which they already live. Retrofitting an existing house may also be quite expensive, and once the client moves out, the sensors may no longer be useful. In contrast, an intelligent robotic assistant can "move" to the home of a new client once a previous client is done with it. Additionally, there may be independent reasons to furnish an elderly person with a mobile-robot assistant, for instance if the robot can stimulate social interaction and/or can provide physical assistance (e.g., help in getting out of chair). In that event, it would be cost-effective to piggyback a cognitive orthotic onto the mobile robot. It is worth noting, however, that the Autominder architecture could be readily used with other sorts of platforms.

4 Related Research

The literature on cognitive orthotics is relatively new, the first survey of the cognitive prosthetic field was done by [4]. Cognitive prosthetics and/or orthotics deal with a large number of varying physiological deficiencies, traumatic brain injury, stroke, neurological disease, Alzheimers, etc. Early approaches to organizing activities and providing clues were developed by Kirsch & Levine[10] and Henry & Friedman et al. [7]. The PEAT system[11] however, is the most similar system to Autominder that we are aware of, and the first to use AI techniques. PEAT is a commercial system delivered on a handheld device, which, like Autominder, is designed to provide its user with reminders about her daily activities. PEAT maintains and dynamically updates a calendar of its client's activities. Autominder differs from PEAT in a number of ways: Autominder handles client plans with complex temporal constraints, it attempts to infer its client's actions, it learns the client's typical behavior patterns, and it reasons about the quality of alternative reminder plans. The large literature on workflow systems (e.g., [6]) is also relevant to Autominder, since workflow systems are designed to guarantee that structured tasks are performed by humans in a timely manner. Discussion of some efforts to integrate AI planning technology with workflow tasks is given in [16].

5 Conclusions

We have described the architecture of Autominder, an agent that provides plan-management assistance to an elderly client. We have shown how we combine a range of AI technologies to provide cognitive orthotic capabilities, and we have argued that incorporation of real-time client data is integral to the effectiveness, autonomy, and user-friendliness of the system. In addition, we have suggested some reasons for using mobile robots as a platform for Autominder.

Prototype versions of Autominder have been implemented, integrated onto a mobile robot (Pearl), and field-tested with elderly people on an abbreviated set of activities. In the current version of the system, the CM does not yet learn client behavior over time, and the PCO does not yet handle preferences. All other mechanisms described above have been implements. In the near future we will be conducting two types of evaluations. First, we will perform more extensive field tests to determine whether the set of activities we current model are adequate for actual monitoring of elderly clients. Second, we will conduct systematic experiments in which we simulate many different executions for given client plans, and generate alternative reminder plans by varying the heuristic evaluation functions. These reminder plans will then be assessed by professional healthcare workers.

References

[1] E. Giunchiglia A. Armando, C. Castellini. Sat-based procedures for temporal reasoning. In *5th European Conference on Planning*, 1999.

[2] José Luis Ambite. *Planning by Rewriting*. PhD thesis, University of Southern California, Los Angeles, CA, 1998.

[3] José Luis Ambite and Craig Knoblock. Planning by rewriting: Efficiently generating high- quality plans. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, 1997.

[4] Elliot Cole. Cognitive prosthetics: an overview to a method of treatment. *NeuroRehabilitation*, 12:39–51, 1999.

[5] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[6] Dimitrios Georgakopoulos, Mark Hornick, and Amit Sheth. An overview of workflow management: From process modeling to workflow autonomation infrastructure. *Distributed and Parallel Databases*, 3:119–153, 1995.

[7] Kimberly Henry, Mark Friedman, Shirley Szekeres, and Debra Stemmler. Clinical evaluation of prototype portable electronic memory aid. In *Proceedings of the RESNA 12th Annual Conference*, pages 254–255, June 1989.

[8] John F. Horty and Martha E. Pollack. Evaluating new options in the context of existing plans. *Artificial Intelligence*, 127(2):199–220, 2001.

[9] C. D. Kidd, R. J. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the Second International Workshop on Cooperative Buildings*, 1999.

[10] Ned Kirsch, Simon P. Levine, Maureen Fallon-Krueger, and Lincoln A Jaros. The microcomputer as an 'orthotic' device for patients with cognitive deficits. *Journal of Head Trauma Rehabilitation*, 2(4):77–86, 1987.

[11] Robert Levinson. Peat – the planning and execution assistant and trainer. *Journal of Head Trauma Rehabilitation*, 1997.

[12] Nursebot: Robotic assistants for the elderly. Avail at `http://www.cs.cmu.edu/~nursebot`.

[13] Angelo Oddi and Amedeo Cesta. Incremental forward checking for the disjunctive temporal problem. In *European Conference on Artificial Intelligence*, 2000.

[14] National Institute on Aging and United States Bureau of the Census. Aging in the united state: Past, present, and future. Avail at http://www.census.gov/ipc/prod/97agewc.pdf.

[15] Martha E. Pollack and John F. Horty. There's more to life than making plans: Plan management in dynamic environments. *AI Magazine*, 20(4):71–84, 1999.

[16] Martha E. Pollack and Ioannis Tsamardinos, and John F. Horty. Adjustable Autonomy for a Plan Management Agent. *1999 AAAI Spring Symposium on Adjustable Autonomy*, Stanford, CA, March, 1999.

[17] A. M. Rivlin. Caring for the disabled elderly: Who will pay?, 1988.

[18] Kostas Stergiou and Manolis Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. In *15th National Conference on Artificial Intelligence*, 1998.

[19] Ioannis Tsamardinos. *Constraint-Based Temporal Reasoning Algorithms, with Applications to Planning*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, 2001.

[20] Ioannis Tsamardinos, Martha E. Pollack, and John F. Horty. 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*, 2000.

[21] Qiang Yang. *Intelligent Planning: A Decomposition and Abstraction Based Approach*. Springer, New York, 1997.