Planning Technology for Intelligent Cognitive Orthotics

Martha E. Pollack
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48103
pollackm@eecs.umich.edu

Abstract

The aging of the world's population poses a challenge and an opportunity for the design of intelligent technology. This paper focuses on one type of assistive technology, cognitive orthotics, which can help people adapt to cognitive declines and continue satisfactory performance of routine activities, thereby potentially enabling them to remain in their own homes longer. Existing cognitive orthotics mainly provide alarms for prescribed activities at fixed times that are specified in advance. In contrast, we describe Autominder, a system we have designed that uses AI planning and plan management technology to carefully model an individual's daily plans, attend to and reason about the execution of those plans, and make flexible and adaptive decisions about when it is most appropriate to issue reminders. The paper concentrates on one of Autominder's three main components, the Plan Manager; other papers in this volume describe its other components (Colbry, Peintner, & Pollack 2002; McCarthy & Pollack 2002).

Introduction

The world's population is aging. The trend in the United States is typical of many industrialized countries. Figures 1 - 3 present populations pyramids based on U.S. census data from 2000, and projections for 2025 and 2050, respectively (Census 2000). Within a population pyramid, each horizontal bar represents the percentage of U.S. residents in a five-year age cohort: the bottom bar represents people aged 0 to 5; the bar above that represents people aged 5 to 10; and so on, up to the topmost bar, which represents people over the age of 100. The population in each age cohort is further divided into males, to the left of the midline, and females, to the right. Historically, the shape of such graphs is pyramidal, as there are more young people than older people.

As can be seen, in 2000, there is a significant bulge in the 25-40 year old cohorts, representing the post-war baby boom, but the basic shape remains pyramidal, with many more people under the age of 60 than people over 60. But by 2025, the pyramid has flattened out, with an increasing proportion of people over 60, and the trend that continues in the 2050 projection.

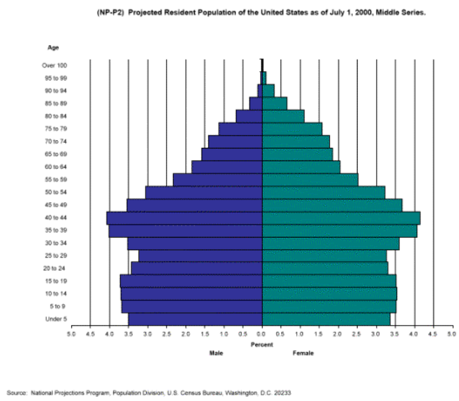

Figure 1: Population Pyramid for the United States in 2000

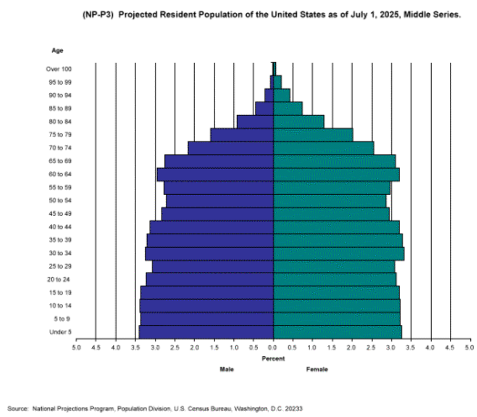

Figure 2: Population Pyramid for the United States in 2025

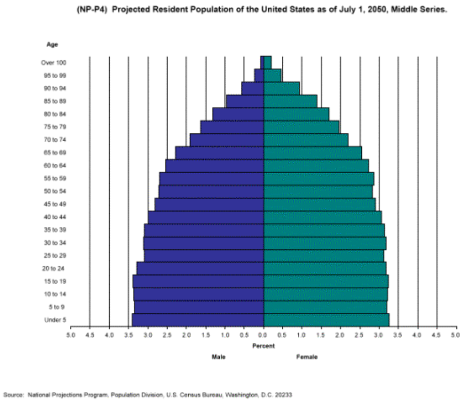

Figure 3: Population Pyramid for the United States in 2050

According to the United Nations Population Division, every region of the world is undergoing a similar demographic transition. In 2000, 606 million people, or approximately 10% of the world's population, were over 60; by 2050, this percentage is expected to double, to 2 billion people, or 21.4% of the population. Even more dramatic will be the increase the percentage of people over 80, often called the "oldest old". Today there are 69 million people in this category, constituting 1.1% of the world's population. Projections show that by 2050 this percentage will nearly quadruple, to 4%: there will be 379 million people over the age of 80. The oldest region of the world today is Europe, with a median age of 37.5; this is projected to rise to 49.5 by 2050 (United Nations 2001).

The aging of the world's population poses a challenge and an opportunity to those of us who design technology. Older adults face a range of challenges: physical, social, emotional, and cognitive. It is important to remember that there is not simply a growing absolute number of older adults, but that older adults will constitute an increasingly large fraction of the population. Thus, while it might be desirable to help older adults meet their challenges by providing them with human assistance, the reality is that there are not and will not be enough younger people to provide all the support and assistance needed. An important question then, is how assistive technology can supplement human caregivers to further enhance the lives of older adults.

Many types of assistive technology have been developed. Devices ranging from the relatively commonplace, e.g., better hearing aids, to the futuristic, e.g., intelligent wheelchairs (Yanco 1998), can help older individuals meet physical challenges. Older adults can be supported socially and emotionally through technology that helps alleviate the isolation that is often a problem for them. For example, elder-friendly email systems (Burd ND)and projects such as the the Digital Family Portrait (Mynatt *et al.* 2001), or the Dude's Magic box (Rowan & Mynatt ND) facilitate increased interaction between an older person and his or her family members and friends. This paper focuses on technology that can help older adults meet cognitive challenges they may face. Specifically, it describes the use of automated planning technology to develop *cognitive orthotics*.

The next section provides a brief discussion of one type of cognitive decline that may occur with aging–a decay in prospective memory–and discusses the limitations of many existing cognitive orthotic systems. Following that, the paper introduces Autominder, a cognitive orthotic designed and built at the University of Michigan using planning and plan management techniques. A description of Autominder's architecture is followed by a focused discussion of one of its three main components: the plan manager. Only brief descriptions of the other main components are given, because other papers in this proceedings provide more details of them (Colbry, Peintner, & Pollack 2002; McCarthy & Pollack 2002). The paper concludes by discussing other recent work on developing intelligent cognitive orthotics, and then summarizing the current state of Autominder and our plans for continued work.

Cognitive Orthotics

Cogntive functioning frequently changes with age: just as the body ages, so does the mind (Stern & Carstensen 2001). Cognitive changes may be due to normal aging, or may be the result of diseases that occur with greater frequency in older people. One of the most common causes of severe cognitive impairment, Alzheimer's Disease (A.D.), is strongly correlated with age: approximately 10% of people age 65 and older suffer from A.D., while 20% of those aged 70-84, and nearly 50% of those over 85 have A.D. (AoA 2000). However, at least as important are milder forms of cognitive impairment that may be prior to and often distinct from A.D. The Autominder system described in this paper is aimed primarily at people with mild to moderate cognitive impairment.

One effect of age-related cognitive decline may be decreased prospective memory, leading to forgetfulness about routine daily activities, which the disability-research community call Activities of Daily Living (ADLs) and Instrumental Activities of Daily Living (IADLs). ADLs include fundamental tasks such as eating, drinking, bathing, and toileting, while IADLs include tasks such as managing medicines, managing money, light housekeeping, arranging transportation, preparing meals, and so on. Of course, older individuals may have physical difficulties that impede their ability to perform ADLs and IADLs, but the technology described in this paper is aimed people whose primary impairments are cognitive ones, which prevent them from remembering to perform these activities.[1]

When an older adult no longer consistently performs ADLs and IADLs, he or she may not be able to remain at home, but may need to move either to the home of a relative or to a facility-based setting such as an assisted care

[1]The Autominder system is currently deployed on a mobile robot, and in the future it may be possible to piggyback on the robot other functions that are intended to help meet physical challenges. For instance, the robot could serve as a delivery system: fetching medicine, water, eyeglasses, mail, and so on.

home. It is generally accepted that for many people, postponing such a move as long as feasible is desirable, because people frequently report a better quality of life while they remain in their own homes. Additionally, institutionalization has an enormous financial cost, which must be born by the individual, his or her family, and/or the government.

A number of cogntive orthotics have been proposed over the years to help older adults adapt to cognitive declines and continue satisfactory performance of routine activities. Not all cognitive orthotics have been specifically targeted to older individuals; some have instead been aimed at people with cognitive impairments resulting from other causes, e.g., brain damage resulting from stroke or injury. 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; Jonsson & Svensk 1995) and pagers (Hersh & Treadgold 1994). Research has also aimed at improved modeling of clients' activities, notably in the work of Kirsch and Levine (Kirsch *et al.* 1987), and in the PEAT system (Levinson 1997). However, with the exception of PEAT, which is discussed further in the Related Research section of this paper, these systems generally function in a manner similar to alarm clocks: they provide alarms for prescribed activities at fixed times that are specified in advance by a client and/or his or her caregiver. For example, the web page for a typical cognitive orthotic, the "Schedule Assistant," developed and marketed by AbleLink Technologies, describes its capabilities as follows:

> To set up an appointment or reminder in Schedule Assistant, caregivers use a wizard approach to complete the process of recording a message or reminder, selecting a picture prompt to accompany the message if desired, and setting the time and day for it to play. The system is then able to "wake itself up" to play the appointment message at the desired time(AbleLinkTech 2002).

Although significant attention has been given to the critical issues of usability and interface design in existing systems, less emphasis has been paid to the process of carefully modeling the client's plans, attending to and reasoning about their execution, and deciding whether and when it is most appropriate to issue reminders. Such reasoning is the focus of the Autominder system, described in the next section.

Autominder

The Autominder cognitive orthotic is being developed as part of the Initiative on Personal Robotic Assistants for the Elderly, a multi-university, multi-disciplinary research effort conceived in 1998.[2] The initial focus of the Initiative is to design an autonomous mobile robot that can "live" in the home of an older individual, and provide him or her with reminders about daily plans. To date, two prototype robots have been designed and built by members of the initiative at Carnegie Mellon. The more recent of these robots, named Pearl, is depicted in Figure 4. Pearl is built on a Nomadic Technologies Scout II robot, with a custom-designed and manufactured "head", and includes 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, and stereo camera systems (Baltus *et al.* 2000; Montemerlo *et al.* 2002; Pineau & Thrun 2002). Members of the Initiative also have interests both in other ways in which mobile robots can assist older people (e.g., telemedicine, data collection and surveillance, and physically guiding people through their environments), and in other platforms for the cognitive orthotic system (e.g., wearable devices and aware homes).

Figure 4: Pearl: A Mobile Robot Platform for the Autominder Cognitive Orthotic. Photo courtesy of Carnegie Mellon University.

One of the main software components of Pearl is the cognitive orthotic system Autominder, which is being developed by members of the initiative at the University of Michigan. Our goal is to develop a system that is flexible, adaptive, and responsive–and is thus more effective than a glorified alarm clock. To attain this goal, Autominder must maintain an accurate model of the client's daily plan, monitor its performance, and plan reminders accordingly. Consider, for instance, a forgetful, elderly person with urinary incontinence who is supposed to be reminded to use the toilet every three hours, and whose next reminder is scheduled for

[2] In addition to the University of Michigan, the initiative includes researchers at the University of Pittsburgh and Carnegie Mellon University.

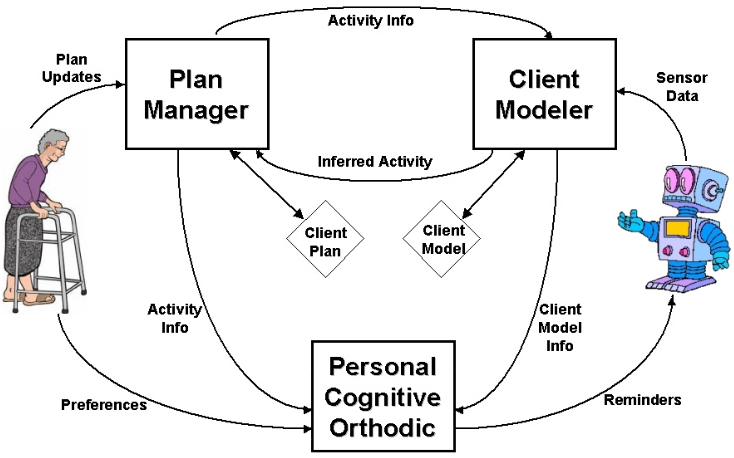

Figure 5: Autominder Architecture

11:00. Suppose that, using its on-board sensors, our robot Pearl observes the person enter the bathroom at 10:40, and conveys this information to Autominder, which concludes that toileting has occurred. In this case, a reminder should not be issued at 11:00, as previously planned. Instead, the client's plan must be adjusted, so that the next scheduled toileting occurs approximately three hours later, i.e., around 13:40. Flexibility is again essential, because a strict three-hour interval may not be optimal. For instance if the client's favorite television program is aired from 13:30 to 14:00, it might be better to issue the reminder at 13:25, 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.")

Autominder's architecture is depicted in Figure 5. As shown, Autominder has three main components: a Plan Manager (PM), which stores the client's plan of daily activities in the *Client Plan*, 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, storing its beliefs about the execution status in the *Client Model*; and a Personal Cognitive Orthotic (PCO), which reasons about any disparities between what the client is supposed to do and what he or she is doing, and makes decisions about when to issue reminders.

Plan Management in Autominder

In Autominder, as in most automated planning systems, we model plans as 4-tuples, $< S, O, L, B >$, where S are steps in the plans, and $O, L,$ and B are temporal ordering constraints, causal links, and binding constraints over those steps.[3] For this application, temporal constraints are very important, and a rich class of such constraints much be supported; specifically, we use the language of disjunctive temporal problems (DTPs) (Oddi & Cesta 2000; Stergiou & Koubarakis 2000; Tsamardinos 2001; Tsamardinos & Pollack 2002) which allows for both quantitative (metric) and qualitative (ordering) constraints, as well as conjunctive and disjunctive combinations of these. We have also recently developed an approach to handling conditional constraints (Tsamardinos, Vidal, & Pollack 2002), but we have not yet implemented these in the Autominder PM.

Formally, each ordering constraint has the form

$$lb_1 \leq X_1 - Y_1 \leq ub_1 \vee \ldots \vee lb_n \leq X_n - Y_n \leq ub_n$$

where the X_i and Y_i refer to the start or end points of steps in the plan, and the lower and upper bounds (lb_i and ub_i) are real numbers. (Without loss of generality, we will assume in this paper that they are integers.) Figure 6 shows how such constraints can be used to express the time at which a step starts or ends, the duration of a step, the amount of time between steps, and so on, as well as expressing ranges and/or disjunctions over such values. Throughout this paper, the start of a step A will be denoted A_S and its end will be denoted A_E. Note that to express a clock-time constraint, e.g., TV watching beginning at 18:00, we use a *temporal reference point* (TR), a distinguished value representing some fixed clock time. In the figure, as well as in the Autominder system itself, the TR corresponds to midnight; the schedule is updated each day.

Note also how the disjunctive constraints can be used to express the fact that two steps cannot overlap. We illustrate this further in Figure 7, which shows a DTP network representing the temporal constraints for a very small plan. The nodes in the network represent the start and end points of each step in the plan, plus the temporal reference point, while the arcs represent the nondisjunctive constraints. The one disjunctive constraint is used to enforce the fact that the two steps in the plan cannot overlap. It should be clear from this example that disjunctive constraints also can be used to express alternative temporal means of resolving a conflict in a plan, i.e., we can represent the possibility of promotion *or* demotion in one constraint.

Plan Initialization

The PM in Autominder is initialized in advance of its use with a specification of the client's daily plan, which is constructed by the client's caregiver, possibly in consultation with the client him- or herself. Different daily plans might be constructed, e.g., one for weekdays and one for weekends, with the appropriate plan loaded each morning, but here we will assume that there is just one daily plan.

We currently have a rather minimal GUI for specifying a daily plan.[4] It allows one to select pre-constructed plan fragments for routine activities from a library, and to then input specific temporal constraints on the steps in the selected fragments. Thus, a caregiver might begin construction of a typical daily plan by performing the following steps:

[3] In the current version of Autominder, we work with a propositional representation, and thus omit binding constraints. On the other hand, we have an extended class of links allowed: in addition to traditional causal links, we also have implemented inconditions and (simple) resource constraints.

[4] The same GUI can be used for modifying the plan once execution has begun.

"Toileting should begin between 11:00 and 11:15."	$660 \leq Toileting_S - TR \leq 675$
"Toileting takes between 1 and 3 minutes."	$1 \leq Toileting_E - Toileting_S \leq 3$
"Watching the TV news can begin at 18:00 or 23:00."	$1800 \leq WatchNews_S - TR \leq 1802 \vee$ $2300 \leq WatchNews_S - TR \leq 2302$
"The news takes exactly 30 minutes."	$30 \leq WatchNews_E - WatchNews_S \leq 30$
"Medicine should be taken within 1 hour of finishing breakfast."	$0 \leq TakeMeds_S - EatBreakfast_E \leq 60$
"Toileting and watching the news cannot overlap."	$0 \leq WatchNews_S - Toileting_E \leq \infty \vee$ $0 \leq Toileting_S - WatchNews_E \leq \infty$

Figure 6: Examples of the use of DTP Constraints

Sample Temporal Plan

•Eat breakfast, starting between 7:00 and 8:00; it will take 20 to 30 minutes.

•Bathe, which will take 30 to 40 minutes.

•Complete breakfast and bath by 8:30.

•Breakfast and bathing may not overlap temporally.

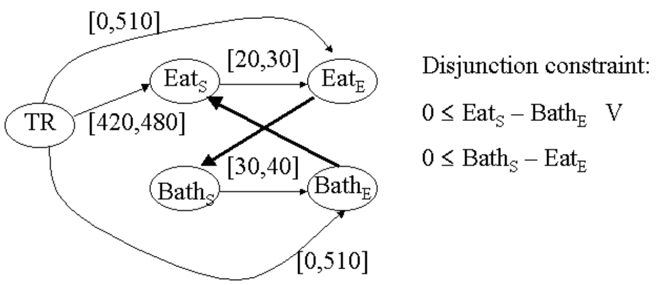

Figure 7: Temporal Network for a Sample Plan. Note the disjunctive constraint that blocks the steps from overlapping.

- Select a pre-constructed plan fragment for breakfast, which includes three steps–going to the kitchen, making breakfast, and eating breakfast–as well as temporal constraints that order these, causal links that capture their dependencies, and some default durations, e.g., that the eating step will take between 20 and 30 minutes.
- Specify that the first step in the breakfast plan must begin by 7:00, and that the last step must be done by 8:30.
- Select a pre-constructed plan fragment for taking medicine, which we will suppose has only one step–take the medicine–with a default duration of 1 minute.
- Specify an interstep constraint to ensure that the medicine taking occurs ate least two hours after finishing breakfast.

As each pre-constructed plan fragment or constraint is added, the PM performs step merging (Tsamardinos, Pollack, & Horty 2000; Yang 1997), that is, it checks to ensure the consistency of the daily plan being constructed and resolves any conflicts. To do this, it uses the same techniques for consistency checking that are used during plan execution; these techniques are described in the next subsection.

Although our current interface is sufficient for development and testing purposes, it seems clear that further work is required to develop more user-friendly interfaces to allow caregivers to specify plans. Little work has been done on this topic, but see (Miksch *et al.* 1998) for one example of the kinds of interfaces that might be developed.

It is worth stressing that the PM is not a traditional plan-generation system. For the kinds of routine activities that we need to represent in our cognitive orthotic, there seems to be little need to perform planning from scratch. Instead, it is sufficient and more efficient to construct generic plan fragments, and allow the PM to merge these fragments, a process that involves adding new constraints, but not new steps or causal links. In future versions of the system, we may extend the PM to do full-fledged planning or replanning when necessary.

Plan Update

The primary role of the PM is to update the client's plan as the day progresses, ensuring its continued consistency. Update occurs in response to four types of events:

1. *The addition of a new activity to the plan.* The daily plan created at initialization provides a starting point for daily activities, but during the course of the day, the client and/or his or her caregivers may want to make additions to the plan: for instance, to attend a bridge game or a newly scheduled doctor's appointment. At this point, plan merging must be performed to ensure that the overall plan remains consistent. Suppose that the client plan initally specifies taking medicine sometime between 14:00 and 15:00, and that the client then adds a bridge game outside the apartment, to begin at 14:30. The PM must update the plan so that the medicine-taking step precedes the client leaving for the bridge game. (We assume that the medicine must be taken at home.) If, in addition, the medicine-taking must occur at least two hours after each meal, the added restriction on when the medicine will be

taken may also further restrict the time at which lunch should be eaten.

2. *The modification or deletion of an activity in the plan.* This is similar to the previous case: the bridge game might be cancelled, or the doctor's office may change the time of the appointment.[5] The types of required changes are like those needed when an activity is added. Note that the PM will add or tighten constraints if needed, but will not "roll back" (i.e., weaken) any constraints. Continuing the example above, if the bridge game were cancelled, the constraint that the medicine be taken between 14:00 and 14:30 would remain in the plan. More sophisticated plan retraction is an area of future research.

3. *The execution of an activity in the plan.* The PM interacts with another component of Autominder, the Client Modeler (CM). The CM is tasked with monitoring plan execution. It receives reports of the robot's sensor readings, for instance when the client moves from one room to another, and uses that to infer the probability that particular steps in the client plan have been executed; it can also issue questions to the client for confirmation about whether a step has been executed. When the CM believes with probability exceeding some threshold that a given step has begun or ended, it passes this information on to the PM. The PM can then update the client plan accordingly. Suppose again that medicine-taking is supposed to occur at least two hours after the completion of each meal. Upon learning that breakfast has been completed at 7:45, the PM can establish an earliest start time of 9:45 for taking the medicine.

4. *The passage of a time boundary in the plan.* Just as the execution of a plan step may necessitate plan update, so may the non-execution of a plan step. As a very simple example, suppose that the client wants to watch the news on television each day, either from 18:00-18:30 or from 23:00-23:30 p.m. At 18:00 (or a few minutes after), if the client has not begun watching the news, then the PM should update the plan to ensure that the 23:00-23:30 slot is reserved for that purpose. (To keep the example simple, assume that the client always wants to watch from the very beginning of the show.)

To perform plan update in each of these cases, the PM formulates and solves a disjunctive temporal problem (DTP). A DTP is a constraint-satisfaction problem $< V, C >$ where the constrained variables V represent time points–in this case, points corresponding to the start and end of steps–and the constraints C are DTP-constraints, as defined earlier (i.e., disjunctions over differences between time points). The domains for the constrained variables are integers, which in Autominder represent the distance in minutes of the time points from the temporal reference point. For example, a time of 480 might be assigned to the time point that represents the beginning of breakfast; this would correspond to 8:00 (480 minutes after the temporal reference point of midnight). In fact, we do not need to assign exact times to most time points; instead we find solutions that correspond to maximum allowable time intervals.

[5] Currently, we allow arbitrary changes to be made to the plan. In subsequent versions of the system, we will need to implement security mechanisms that, for instance, allow the user to make changes to social engagements but not the medicine-taking actions.

Update-Plan-for-Addition($existing$**,**$newfrag$**)**
    E = Convert-to-DTP($existing$)
    N = Convert-to-DTP($newfrag$)
    C = Identify-conflicts($existing \cup newfrag$)
    R = $\emptyset$
    For each member c of C
        R = R $\cup$ a DTP-constraint representing
        the alternative temporal resolutions of c
    P = E $\cup$ N $\cup$ R
    P' = Solve-DTP(P)
    Return(Convert-to-Plan-Representation(P'))

Figure 8: Algorithm for Update after a Plan Addition

To see how this works, consider first the case of updating the plan in response to a plan addition. Psuedo-code for this case is given in Figure 8. The PM begins with the contents of the Client Plan, $existing$, and a plan fragment representing the new activities to be added to the plan, $newfrag$. Both $existing$ and $newfrag$ are encoded as $< S, O, L, B >$ 4-tuples, and so the first step is to convert them to disjunctive temporal problems, E and N, respectively. This is a trivial process that is linear in the number of steps: it involves simply extracting all the temporal constraints and encoding them in a format that our DTP solving engine can handle. Note that there is information lost in the DTP encoding: specifically, the DTP does not encode causal links. Thus, it is crucial that a temporal constraint be explicitly included for each causal link. Additionally, it is necessary to identify all the threats in the union of $existing$ and $newfrag$, a process that is quadratic in the total number of steps. For each identified threat, the PM then constructs a DTP constraint that represents the alternative methods of resolution; call the set of such threat-resolution constraints R. Finally, a plan P that consists of the union of E, N and R is passed to a DTP-solver, which checks for consistency, and returns P augmented by a set of additional constraints that ensure consistency. In particular, if there are any threats in the plan, a resolution will be selected for each one. The last step in the process is to convert the new set of DTP constraints back to a plan tuple.

DTP solving, which is NP-complete, is the only computationally expensive step in the process. In Autominder, we use the Epilitis DTP-solver (Tsamardinos 2001; Tsamardinos & Pollack 2002). Epilitis integrates a number of efficiency heuristics, and has been demonstrated to solve benchmark problems two orders of magnitude faster than the previous state-of-the art solvers. For our current Autominder scenarios, which typically involve about 30 actions, Epilitis nearly always produces solutions in less than one second, a time that is well within the bounds we require.

Like prior DTP solvers (Oddi & Cesta 2000; Stergiou & Koubarakis 2000; Armando, Castellini, & Giunchiglia

Update-Plan-for-Modification($existing, mods$)

$plan$ = Make the modifications in $mods$ to $existing$
(i.e., remove and/or replace constraints)
M = Convert-to-DTP($plan$)
C = Identify-conflicts($plan$)
R = $\emptyset$
For each member c of C
R = R $\cup$ a DTP-constraint representing
the alternative temporal resolutions of c
P = M $\cup$ R
P' = Solve-DTP(P)
Return(Convert-to-Plan-Representation(P'))

Figure 9: Algorithm for Update after a Plan Modification

1999), Epilitis does not attempt to solve the DTP directly by searching for an assignment of integers to the time points. Instead, it solves a meta-CSP problem: it attempts to find one disjunct from each disjunctive constraint such that the set of all selected disjuncts forms a consistent Simple Temporal Problem (STP) (Dechter, Meiri, & Pearl 1991). An STP is like a DTP, except that the constraints must be atomic inequalities; no disjunctions are allowed. The details are beyond the scope of the current paper (but see (Tsamardinos 2001; Tsamardinos & Pollack 2002)). The important point here is that by using this approach, Epilitis can return an entire STP, which provides interval rather than exact constraints on the time points in the plan. Consider again our example of the plan that involves taking medicine between 14:00 and 15:00, which is amended with a plan to leave for a bridge game at 14:30. Epilitis will return a DTP that constrains the the medicine to be taken sometime between 14:00 and 14:30; it does not have to assign a specific time (e.g., 14:10) to that action.

The other three cases of plan update are similar. In response to a plan modification, the PM again begins with the current contents of the Client Plan, $existing$, but this time, instead of a second plan to merge in, it has a set of constraints from $existing$ that are to be removed or changed. Thus, it makes the specified modifications to $existing$ and then converts it to a DTP, identifies conflicts, and performs DTP solving as before. The pseudo-code for this is shown in Figure 9.

The psuedo-code for the other two cases of plan update is not shown, as they are similar to the previous ones. In the third case of update, a step S has begun or finished execution. In response, the PM shrinks the temporal constraint(s) associated with the start end, and/or duration of S to a unit interval. For instance, if we know that breakfast began at time 480, then the constraint associated with it becomes $480 \leq EatBreakfast_S - TR \leq 480$. As long as execution has occurred within the legal bounds, there is no need to identify conflicts; instead, the resulting plan with the reduced constraints is passed directly to the DTP solver so that the new tighter constraints can be propagated.

In the fourth case, a time boundary has passed without a step having begun or ended. At this point, the PM must remove the now invalidated disjunct from a constraint, and then attempt to solve the DTP anew. In our TV news example, the plan would include a constraint
$1800 \leq WatchNews_S - TR \leq 1802 \vee$
$2300 \leq WatchNews_S - TR \leq 2302$
i.e., that watching the news must start either right about 18:00, or else about 23:00. If this step has not begun by shortly after 18:00, the first disjunct is no longer viable. Thus, the PM must remove it from the representation of the plan, and attempt to resolve the DTP, using the remaining disjunct. In the current example, there is an alternative disjunction to try. Sometimes, though, when an invalidated disjunct is removed, there may not remain any alternatives; in that case an execution failure has occurred. As with other cases of execution failure, e.g., missed deadlines, Autominder would record this fact, making it available to the caregiver if appropriate.

The discussion of passed time boundaries brings to light one point that was passed over earlier. In general, there may be multiple solutions to a DTP, i.e., multiple consistent STPs that can be extracted from the DTP. In the current version of Autominder, the PM arbitrarily selects one of these (the first one it finds). If subsequent execution is not consistent with the STP selected, then the DTP will attempt to find an alternative consistent solution. A more principled approach would select solutions in an order that provides the greatest execution flexibility. For example, the solution that involves watching the 18:00 news leaves open the possibility of instead watching the news at 23:00. If the first solution found instead involved watching the later news show, then after an execution failure there would be no way to recover, as it would be too late to watch the 18:00 news. Unfortunately selecting DTP solutions to maximize flexibility is a difficult problem (Tsamardinos, Pollack, & Ganchev 2001).

Other Autominder Components

In addition to the PM, Autominder has two other principal components. The Client Modeler (CM) was mentioned above in the discussion on updating the plan in response to plan execution. As noted there, the job of the CM is to monitor the execution of the plan, attempting to infer its status from information obtained from the robot sensors and requesting confirmation from the client when appropriate. To build the CM, we have been adapting Bayesian inference mechanisms to handle the temporal demands of this application; details can be found in (Colbry, Peintner, & Pollack 2002).

The remaining component of Autominder is the Personal Cognitive Orthotic (PCO), which is responsible for making the decision about what reminders to issue and when. To do this, the PCO reasons about the client plan and the client model, identifying any evolving discrepancies between them. It turns out to be relatively easy to generate a legal reminder plan–such a plan simply includes a reminder for every planned activity at the earliest possible time of its execution. However, a reminder plan constructed this way is likely to be a rather poor one when judged by the criteria we use in Autominder, namely:

1. ensuring that the client is aware of activities he or she is expected to perform,
2. increasing the likelihood that the client will perform at least the essential activities (such as taking medicine),
3. avoiding annoying the client, and
4. avoiding making the client overly reliant on the system.

While the simple approach would lead to a reminder plan that satisfies the first criteria, it is unlikely to satisfy the third or fourth, and this in turn may have a negative impact on the second criteria. Consequently, we employ the local search techniques of the Planning by Rewriting algorithm (Ambite & Knoblock 2001) to iteratively search for an improved reminder plan; for details of our approach, see (McCarthy & Pollack 2002)

Related Research

Several existing cognitive orthotics systems were mentioned earlier in this paper. The most notable of these from a plan-based perspective is PEAT (Levinson 1997). This was the first, and to the best of our knowledge, the only marketed cognitive orthotic system that relies on automated planning technology. PEAT, which is marketed primarily to patients with traumatic brain injury, is deployed on a handheld device, and provides visible and audible clues about plan execution. Like Autominder, PEAT maintains a detailed model of the client's plan and tracks its execution, propagating temporal constraints when the client inputs information specifying that an action has been performed. Also, upon the addition of a new action, PEAT simulates the plan to uncover any conflicts, using the PROPEL planning and execution system (Levinson 1995) for this purpose. However, PEAT uses a less expressive planning language than Autominder; it does not attempt to infer the plan execution status; and it does not perform principled reasoning about what reminders to issue when, instead automatically providing a reminder for each planned activity.

Within the past year or two, several new projects aimed at designing intelligent cognitive orthotics have begun to emerge. The MAPS project at the University of Colorado is focusing on the HCI issues involved in building a handheld cognitive orthotic (Carmien 2002). The Independent LifeStyle Assistant Project (ILSA) at Honeywell is another recent related effort, which has some aims that overlap with our own (Miller & Riley 2001). Yet another, even newer project is the Assisted Cognition Project at the University of Washington (Kautz *et al.* 2002). While Autominder is being targeted mainly at people with milder forms of cognitive impairment, the Washington project aims at developing a cognitive orthotic system–an *adaptive prompter*–for people with Alzheimer's disease. The system will use ubiquitous sensors to monitor the performance of routine tasks, and provide prompts when a client gets "stuck". For instance, a sensor in the bathroom might notice that a person with A.D. has picked up a toothbrush but then stopped; in response, the adaptive prompter would provide guidance to the person about putting toothpaste on the brush and using it to brush his or her teeth. As can be seen, the adaptive prompter is targeted at people with more severe cognitive decline than what we imagine for a typical Autominder client.

Conclusions

The Autominder system as described in this paper has been fully implemented in Java and Lisp on Wintel platforms; we are also working on a Web-based interface for plan initialization and update. The most recent version system has been tested in the laboratory; an earlier version was integrated with the robot software and included in a preliminary field test conducted at the Longwood Retirement Community in Oakmont, PA in June, 2001. The goals of that test were, first, to ensure that the robot control software and the cognitive orthotic would work together, and second, to get an initial sense of the acceptability of such a system to older individuals. On both accounts, the test was successful. Admittedly, the older adults who enrolled in the studies were volunteers, and people likely to be intimidated or put off by this type of technology would not have volunteered. However, the people who did participate were uniformly excited about the system, as were the staff at Longwood, who made a number of suggestions to us about how this type of technology could also be used to assist them in their caregiving tasks. 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. These field tests will be more directly focused on the performance of the cognitive orthotics software.

We have a number of plans for the continued development of Autominder, some of which were already mentioned in this paper. We have planned extensions to the individual reasoning modules, for example, adding the ability to handle conditional constraints to the PM; supplementing the PM with full-fledged planning capabilities to support replanning; enabling the CM to learn the patterns of client activity over time, in order to better interpret observed behavior; and developing techniques for providing better justifications for reminders issued by the PCO. We are also interested in the deployment of the system on alternative hardware platforms. Although there are many advantages to using a robot, including the ability to piggyback on other capabilities, there are clearly also reasons to explore handheld and/or wearable devices and ubiqitous sensors to support cognitive orthotics. Finally, after our experiences with the staff at Longwood, we are interested in exploring the use of systems like ours within the facility-based setting. In that context, the system would coordinate the daily plans not only of a single person, but of multiple people, including both the residents and the staff that takes care of them.

Acknowledgements

Primary financial support for this research comes from an ITR grant from the National Science Foundation (IIS-0085796). Some of the supporting technology was developed with funding from the the Air Force Office of Scientific Research (F49620-01-1-0066) and the Defense Advanced Research Projects Agency and Air Force Research

Office, Air Force Materiel Command, USAF (F30602-00-2-0621). The views and conclusions herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR or the U.S. Government.

I would like to thank all the members of the Nursebot team, and especially my students, Laura Brown, Dirk Colbry, Colleen McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos, who do all the hard work, somehow turning my half-baked ideas into theories and working systems.

References

Able Link Tech. 2002. Ablelink technologies, `http://www.ablelinktech.com`. Quotation from product descriptions subpage.

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

AoA. 2000. Factsheet on Alzheimer's disease. U.S. Government Administration on Aging, `http://www.aoa.gov/factsheets/alz.html`.

Armando, A.; Castellini, C.; and Giunchiglia, E. 1999. SAT-based procedures for temporal reasoning. In *5th European Conference on Planning*.

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*.

Burd, L. N.D. I-mail. `http://www.cs.colorado.edu/~l3d/clever/projects/i_mail/index.html`.

Carmien, S. 2002. MAPS: PDA scaffolding for independence for persons with cognitive impairment. `http://www.cs.colorado.edu/~l3d/clever/projects/maps/`.

Census 2000. Census report 2000: National projections program/National estimates program. Population Division, U.S. Census Bureau, Washington, D.C. 20233.

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*.

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

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

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. *Journal of Medical Systems* 22:95–101.

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

Jonsson, B., and Svensk, A. 1995. Isaac: A personal digitial assistant for the differently abled. In *Proceedings of the 2nd TIDE Congress*, 356–361.

Kautz, H.; Fox, D.; Borriello, G.; and Arnstein, L. 2002. An overview of the assisted cognition project. In *AAAI Workshop on Automation as Caregiver*. To appear.

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.

Levinson, R. 1995. A general programming language for unified planning and control. *Artificial Intelligence* 76.

Levinson, R. 1997. PEAT–the planning and execution assistant and trainer. *Journal of Head Trauma Rehabilitation* 769.

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*.

Miksch, S.; Kosara, R.; Shahar, Y.; and Johnson, P. 1998. Asbruview: Visualization of time-oriented, skeletal plans. In *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems*, 11–18.

Miller, C., and Riley, V. 2001. The independent lifestyle assistant. In *Proceedings of the XVIIth World Congress of the International Association of Gerontology*.

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*. To appear.

Mynatt, E. D.; Rowan, J.; Craighill, S.; and Jacobs, A. 2001. Digital family portraits: Providing peace of mind of extended family members. In *Proceedings of the ACM Conference on Human Factors in Comuting Systems*, 333–340.

Oddi, A., and Cesta, A. 2000. Incremental forward checking for the disjunctive temporal problem. In *European Conference on Artificial Intelligence*.

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

Rowan, J., and Mynatt, E. D. N.D. Dude's magic box and grandma's lap desk. `http://www.cc.gatech.edu/fce/ecl/projects/dude/index.html`.

Stergiou, K., and Koubarakis, M. 2000. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence* 120:81–117.

Stern, P. C., and Carstensen, L. L., eds. 2001. *The Aging Mind: Opportunities in Cognitive Research*. Washington, D.C.: National Academy Press.

Tsamardinos, I., and Pollack, M. E. 2002. Efficient solution techniques for disjunctive temporal problems. Under review; available upon request from the authors.

Tsamardinos, I.; Pollack, M. E.; and Ganchev, P. 2001. Flexible dispatch of disjunctive plans. In *Proceedings of the 6th European Conference on Planning*, 417–422.

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.; Vidal, T.; and Pollack, M. E. 2002. CTP: A new constraint-based formalism for conditional, temporal planning. *Constraints*. To appear.

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

United Nations. 2001. World population prospects: The 2000 revision highlights. United Nations Population Division. `http://www.un.org/esa/population/publications/wpp2000/wpp2000h.pdf`.

Yanco, H. A. 1998. Wheelesley, a robotic wheelchair system. In Mittal, V. O.; Yanco, H. A.; Aronis, J.; and Simpson, R., eds., *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial Intelligence*. Springer-Verlag. 256–268.

Yang, Q. 1997. *Intelligent Planning: A Decomposition and Abstraction Based Approach*. New York: Springer.