A Conversational Cognitive Aid with Activity Monitoring, Planning and Execution

Richard Levinson, David Halper
Attention Control Systems, Inc.
Mountain View, CA
brainaid.com
{rich, dave}@brainaid.com

Henry Kautz, Craig Harman
Department of Computer Science
University of Rochester
{kautz, charman}@cs.rochester.edu

Abstract

We present recent work to add sensors and activity recognition to a commercially available cognitive aid, enabling context-aware activity monitoring, planning and cueing. The Conversational Assistant for Rehabilitation (CARE) system is a context-aware autonomous agent that interacts with users via spoken conversation, similar to user interaction with a human caregiver. It includes activity models for both the user and the virtual caregiver. CARE's primary activity is talking with the user about their plan, performance and situation. CARE cannot directly execute user activities like bathing or eating, which are viewed as exogenous events by CARE.

We begin with an overview of current executive function support provided by PEAT, the advanced cognitive aid that we are extending. We then discuss how we are extending PEAT with activity monitoring via sensors and activity recognition, a speech interface, and autonomous agent architecture with unified planning and execution.

1 Introduction

Conversation between a patient and their human caregiver is important for building the patient's trust and acceptance of the helper. This may also be true for electronic cognitive assistants. Human caregivers also function independently from the patient. Electronic cognitive aids simply echo the user's schedule back, but human caregivers think independently about when and what to say.

We propose a conversational caregiver's assistant in a mobile phone to help veterans with cognitive impairment. Patients speak with the CARE agent as if they were speaking with a caregiver on a phone. The CARE agent and patient use dialogue to resolve questions and learn preferences, building the user's trust and acceptance of device. We are extending a commercial cognitive aid already being used by the VA, by adding physiological sensors and a conversational interface.

The CARE agent provides context-aware conversational interventions, talking with users during Activities of Daily Living (ADLs), PTSD homework and in-vivo exposure sessions, and collecting experience sampling and biosensor data for therapists to determine if patient should move to the next level of the exposure hierarchy. Anticipated benefits include increased user independence and reduced long-term caregiver costs. Our Medical Advisory Board includes VA neuropsychologists who will provide guidance about clinical and commercial considerations.

1.1 Needs of veterans with cognitive disorders

<u>Executive Functions</u>
Traumatic Brain Injury (TBI) and Post-Traumatic Stress Disorder (PTSD) often impair a person's ability to plan and carry out activities, and to adjust plans for changing situations, which are called executive functions. Many returning veterans with these impairments require cognitive assistance to maintain or regain independence and quality of life. A primary goal of this work is to address the need to support impaired Executive Function, including:

- Impaired capacity for activity monitoring, planning and execution
- Inflexible activity management - difficulty adjusting plans in response to changing goals and situations, including error recovery

<u>Skilled Therapy Support In Home Community</u>
Another need is to provide greater access to skilled cognitive therapy support and methods after veterans leave larger VA centers and return home to their community where there are fewer cognitive rehabilitation specialists. Thus a second objective is to support therapy in home and community. We do this by providing tools for therapists to configure therapeutic goals and methods for each client, thus extending the reach of skilled clinicians outside of their office. The system runs on a phone and can call for help in an emergency. However, the intent is that the system operates autonomously, only checking in with the therapist for periodic "check ups". Usage scenarios include guiding users through ADLS, cognitive therapy including exposure therapy for PTSD, and improving therapy compliance by reminding users when and how to follow their therapy plan (e.g., using a cane when walking outside).

1.2 Limitations of Current Assistive Technology

Existing cognitive aids have several limitations when addressing the problems described in Section 1.1, which have prevented their widespread acceptance by users and caregivers. These limitations include:

- **No Therapy Agent or Therapy Model:** Existing systems do not allow input of therapy goals or plans or for reporting feedback to the therapist. Current systems simply echo the user's schedule back to them like a talking calendar. They do not have the awareness or decision making capability of a human caregiver, who can reason independently about the user's situation and consider different intervention options.
- **Unnatural Interfaces:** The user interface and data entry is complicated for users with cognitive impairment. Interaction using computer displays and buttons is unnatural compared to speaking with a human aid.
- **Limited Awareness:** Pre-scheduled activity reminders may be out of sync and unresponsive to changing situations. Current cognitive aids are only aware of these changes when the user remembers to update the aid.
- **Inability to Learn:** Existing systems have difficulty learning user preferences over time through observation and discussion, the way that human caregivers do.

1.3 Proposed Solution: The Conversational Assistant for Rehabilitation (CARE)

We propose to address the limitations given in Section 1.2 by extending PEAT to act as a "virtual caregiver." Figure 5 shows this system, called the Conversational Assistant for REhabilitation (CARE). Unlike the current PEAT system, CARE is an independent agent from the user. It is an autonomous "virtual caregiver" that monitors the user's behavior and deliberates about when and what to say to users, intervening only when necessary. CARE's key features include:

- **CARE Agent:** An autonomous agent with a sense-plan-act cycle, based on methods used for NASA's autonomous robots [Levinson 1995b; Levinson 2005, Muscettola et al, 2002; Muscettola et al., 2000; Verma et al, 2005].
- **Therapy Support:** Allow caregivers to specify therapy goals and plans, and provide feedback to caregivers about user performance and therapy compliance.
- **Conversational Interface:** Enables human-like interaction. Designed to be similar to speaking with a human caregiver on the phone. The CARE agent and patient use dialogue to resolve questions and learn user preferences, building the user's trust in the system and leading to increased user acceptance of device. The caregiver uses dialog to resolve questions and learn user preferences.
- **Context Awareness through Sensing:** We will use sensors to try to understand the user's current context, building on our prior research [Modayil, et. al 2008a; Modayil 2008b].
- **Machine Learning:** We will implement machine-learning algorithms that search for patterns in the contextual information (e.g. location, schedule, detected activity) collected by the sensors, again building on our prior work [Liao et al., 2007; Liao et al., 2004].

2 Overview of the PEAT cognitive aid

This section presents an overview of PEAT, the Planning and Execution Assistant and Trainer, which is being extended for this project. PEAT is a NASA Spinoff technology based on autonomous planning and control software developed to provide executive functions for robotic systems [Levinson 1994; Levinson 1997; Levinson1995a; 1995b]. Like humans, autonomous robots must balance planning and reactivity in order to achieve goals in uncertain and changing situations.

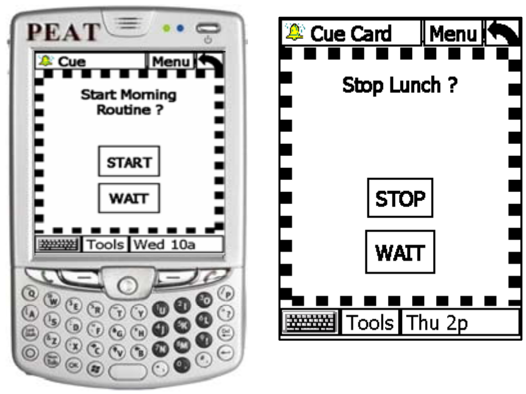


Figure 1: PEAT's cue card shows the current activity and asks users to confirm when starting and stopping activities. PEAT's planner adjusts the schedule based on user self-report of progress.

PEAT's patented software provides executive function assistance for users with cognitive impairment [Levinson 2006; Levinson 2002; Levinson 2000]. The software runs on Windows Mobile PDAs and smart phones. PEAT provides closed-loop activity management help by cueing users through daily activities, monitoring their progress, and replanning in response to changing goals and situations [Levinson 1997].

Figure 1 shows PEAT running on a mobile phone. The system is customized for each user by hiding buttons and access to parts of the system that might confuse them, and also by using personalized activity models, voice recordings, and pictures. PEAT is the only commercial product that provides compensatory support to help with planning activities, monitoring their execution progress, and replanning when changes occur. PEAT is the only system to "close the loop" with integrated monitoring, planning and cueing, to compensate for important executive functions including planning, choice making, sequencing, error detection and error correction.

PEAT includes several features designed to assist users with executive function impairment, including expressive action representations, automatic planning for flexible scheduling, and a wide range of cueing options.

CUE Card: A key feature of PEAT is the unique *Cue Card* (Figures 1 and 2). The cue card is designed to help users with initiation (starting tasks), focus (staying on task), and perseveration (stopping or switching tasks). PEAT's cue card shows how much time until the next task (left) and how much time remains for the current task (right), along with relevant pictures and linked notes.

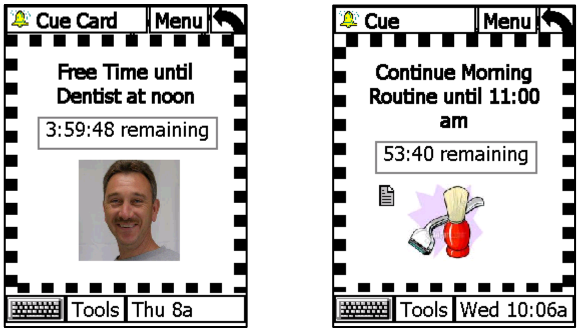


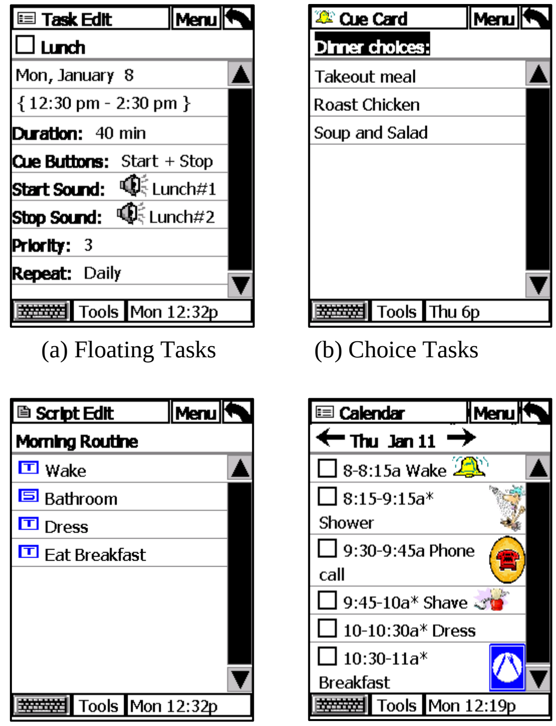
Figure 2: PEAT's cue card shows how much time until the next task (left) and how much time remains for the current task (right), along with relevant pictures and linked notes.

The Cue Card includes the following unique features:

- Information only about the current activity to avoid distractions.
- Highly impaired uses may be restricted so they have access only to the Cue card. The caregiver may set up the schedule for days or weeks in advance but the user my only see the current cue.
- Intrusive and Persistent Cues. System turns power on, and automatically jumps to Cue Card when its time to cue the user to help with initiation. The system keeps cueing until the user responds. The level of intrusiveness is customized for each task and each user.
- Cues include customized pictures and voice recordings
- Stop cues remind users who perseverate (get stuck in repetitive behavior) to move on to another task.
- Cue overrides allow users to delay the cue (like "snooze" button), to start tasks early or late, and to skip or reschedule them. The schedule is automatically adjusted as necessary. These overrides are optional and customized for each user.

PEAT supports a variety of real-world activity representations, in addition to the standard "appointment" which starts at a specific time.

Floating Tasks may start anywhere within time window. Figure 3a shows that Lunch is a 40-minute task that may occur between 12:30pm and 2:30pm with priority 3. It may be delayed within that time window by higher priority tasks. If the duration fills the time window, then the task is a Fixed Task (e.g., appointment) with a start time that cannot slip. All task are put in scheduled in priority order (highest priority tasks are scheduled first), and are rescheduled as necessary based on cue responses and calendar changes.



(a) Floating Tasks (b) Choice Tasks

(c) Scripts (Task Sequences) (d) Morning Script adjusted to fit phone call at 9:30

Figure 3: PEAT activity types include floating tasks, scripts and choices. The Planner reasons about task deadlines, durations, and sequence constraints.

Choice Tasks prompt the user to select from context-dependent task choices. Figure 3b shows a prompt for dinner choices.

Scripts are hierarchical task sequences. Figure 3c shows the generic morning routine with four steps. The "S" in front of the second step indicates that Bathroom is also a script (a "sub-script"). Figure 3d shows a scheduled Morning Routine that has been adjusted to accommodate phone call at 9:30. The phone call was inserted into the Bathroom sub-script, between Shower and Shave. Script steps may include appointments with fixed start times, floating tasks, and choice tasks. Choices may also include scripts, so a dinner choice to may include a script for roasting a chicken.

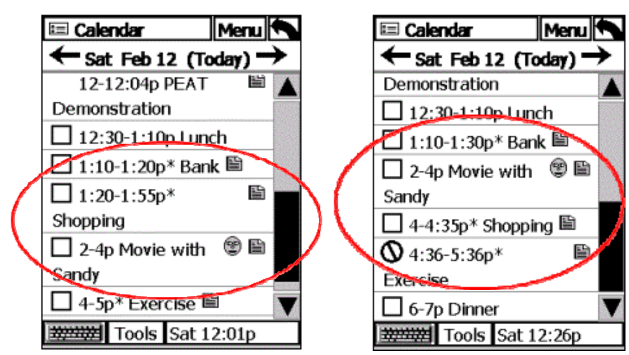


Figure 4: Delay at the bank example of replanning.

Example: Figure 4 shows an example of how a delay at the bank may cause replanning. On the left, the Bank task is scheduled until 1:20 and Shopping (a floating task) is before Movie (a Fixed task with a start time of 2pm). On the right, leaving Bank at 1:30 instead of 1:20 causes Shopping to be delayed until after Movie, cancelling the low-priority Exercise task.

3 CARE Agent

PEAT provides more executive function support than other commercially available solutions, but is limited by the lack of real-time activity monitoring. It can automatically adjust the schedule for delays and changing situations but relies on the user to self-report those delays or changes.

The Conversational Assistant for REhabilitation (CARE) system functions as an autonomous agent in a mobile phone with integrated sensing, planning and execution (Figure 5).

The agent takes as input a set of domain models which describe both the user and the virtual caregiver behavior. This includes procedural activity models for user scripts such as physical therapy procedures or roasting a chicken. It also includes procedural activity models for the CARE agent such as talking with the user about scheduling an activity or their user interface preferences. These procedures include choice points which identify alternative methods (subroutines) and resources (parameter values), defining a search space of procedure variations. The agent interacts with the user through the phone interface and it receives data from external sensors to identify the user's location, objects they touch, and their pulse and respiration.

The CARE agent functions as a caregiver that is independent from the user. Rather than echoing the user's schedule like a talking calendar, CARE reasons independently about therapeutic plans and goals such as when and what to say to the user in order to increase independence and improve therapy outcomes, and learn user preferences. CARE can help plan the user's activities like dinner but cannot actually execute them. CARE's primary action is talking with the user to ask about their plans and preferences.

The CARE agent's behavior is defined by its goals and actions. CARE Goals include: Increasing user independence, responding to user commands and queries, helping user complete their goals and activities, Learning user preferences. CARE Actions include:

- Planning the user's activities and adjusting the schedule to delays and calendar change, managing schedule conflicts.
- Conversation (integrated speech input and output).
- Activity Prompts
- Asking the user about their in-situ experience and preferences.

3.1 Conversational Interface

We have developed an initial prototype for a speech-based interface to PEAT. This includes Automatic Speech Recognition (ASR) and an optional lip-syncing Avatar. The system supports Mixed Initiative Conversation:

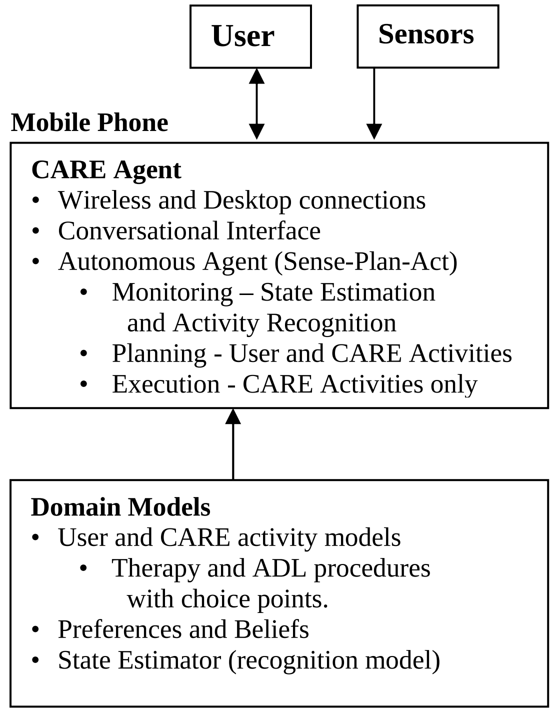


Figure 5: The Conversational Assistant for Rehabilitation (CARE). A caregiver's assistant in a mobile phone.

- User initiates commands like adding tasks or making a phone call.
- Computer initiates conversations to cue the user, remind them about therapy compliance, and/or query how they feel or what they are doing.

We believe a conversational interface enables a more natural and flexible interaction between patient and caregiver. Additionally, PTSD treatment involves explicitly discussing the patient's beliefs in attempts to replace negative beliefs with positive alternatives.

3.2 Activity Monitoring

Figure 6: RFID Bracelet detects RFID tag on box

Sensing: Sensors are used to monitor the user's location, the objects they touch and their physiological state The agent receives sensor data through the phone's wireless sensors which include Bluetooth, WiFi, GPS and phone services. The sensors enable condition-based cues compared to time-based cue which may be out of sync with changing conditions. CARE monitors the following state conditions:

Location: CARE monitors the user's location using GPS outdoor and other methods indoor. Figure 7 shows a pressure mat on the floor which detects when user steps on it.

Objects: CARE detects when a user touches objects tagged with Radio Frequency ID (RFID). Figures 6 and 7 show a user wearing an RFID reader bracelet which detects RFID tags on objects within about 5 inches. This allows CARE to detect when user touches objects which have RFID tags on them.

Biosensors: Wearable sensors relay heart and respiration rate, skin temp and movement via Bluetooth to CARE. This is used to monitor the user's physiological state during PTSD exposure therapy.

Activity Recognition: Hidden Markov Models (HMM) are used to infer activities from the sensors [Modayil, et. al 2008a; 2008b]. We classify a user's activities based on observation streams from sensors. This activity classification can be done with a hidden Markov model (HMM) [Rabiner 1989]. In a simple HMM for activity recognition, there is one state per activity, and each activity has some probability of generating the current sensor observations. Formally, given a set of activities *A* and a set of observations *O*, the probability of having performed activity a_t at time *t* after seeing the observation o_t only requires knowing the probability distribution over the activities at the previous time step. This is described by the following equation:

$$P(a_t|o_{1,\ldots,t}) \propto P(o_t|a_t) \sum_{a_{t-1} \in A} P(a_t|a_{t-1}) P(a_{t-1}|o_{1,\ldots,t-1})$$

At each time-step, the best estimate of the current activity is updated in the state vector. The estimate is qualitative, either indicating that the activity classifier is confident that the activity is currently being attempted or indicating that the current activity is ambiguous when the classifier is not confident.The classification is ambiguous when the probability of the most likely activity falls below a threshold.

Although an HMM is commonly used with the Viterbi algorithm to find the most likely sequence of states to explain a sequence of observations, the State Estimator needs to know what the user is *currently* attempting. This classification must be provided online and in real time for timely and effective interventions. Hence, HMM filtering is used instead of the Viterbi algorithm. Since the observation and activity sequences do not have to be stored, the filtering algorithm requires only a constant amount of memory.

Previous research [Philipose et al, 2004] on recognizing activities of daily living with RFID has shown that isolated activities can be recognized with a hidden Markov model with a limited number of states. Difficulties can arise when a user is multitasking between multiple activities in natural environments. Patterson and colleagues [Patterson et al, 2005] collected data from a variety of morning activities

Figure 7: Pressure mat on floor and RFID reader bracelet send wireless data to mobile phone indicating the user is standing in front of the refrigerator and touching the door handle

using RFID tags and readers. They compared multiple activity recognizers and found that an HMM with one state per activity performed well, but increasing model complexity did not improve the recognition performance.

We have developed the Interleaved HMM (IHMM) [Modayil et al., 2008a] as a better variant of an HMM for the classification of interleaved activities. The IHMM augments the state representation in a simple HMM (one state per activity) with a richer HMM state representation that stores the last observation seen in each activity (one state for each observation symbol for each activity). When activities are interleaved, the IHMM can better predict the next observation based on the last observation for the new activity. The IHMM has a very large state space, but an effective approximation reduces the portion of the state space considered at each time step to be comparable to that used for the simpler HMM. The introduction of explicit representations for the interleaving of activities improves the accuracy for both the Viterbi algorithm and the filtering algorithms.

Closely related work to the Interleaved HMM (IHMM) is presented by Duong and colleagues [Doung et al 2005]. They recognize several activities of daily living using a hierarchical hidden semi-Markov model. Their algorithm can perform accurate recognition for some ADLs using observations from cameras that track a user's location.

3.3 Context-aware planning and cueing

Appropriate Cues – The new monitoring system enables condition-based cues that are more in sync with user situation than time-based cues. When monitor detects user starts tasks ahead of schedule (without prompting). Figure 8 shows how tasks are adjusted when the user picks up the cereal a half-hour ahead of schedule. The planner adjusts the schedule to the change, inhibiting inappropriate cues at the originally scheduled time.

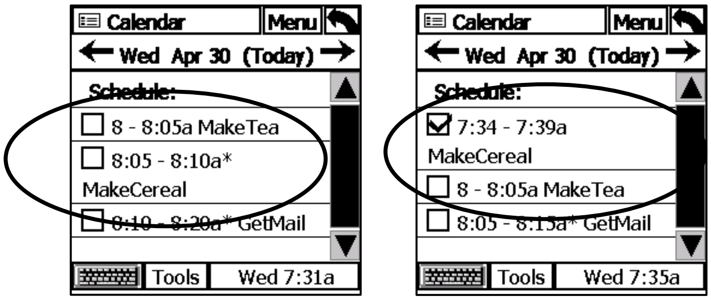


Figure 8: Make Cereal is originally scheduled after Make Tea at 8:05, but the monitor detects user starts activity at 7:30 (shown in Figure 6), then adjust schedule as needed (above).

Figure 9: CARE detects that user steps outside without the cane and generates a contingent cue.

Contingent Cues – Contingent Cues are not pre-scheduled but are triggered by monitored conditions. We've extended PEAT activities to specify monitored conditions that must be true when task starts (**preconditons**) or ends (**success conditions**). This enables contingent cues which remind the user when preconditions are not satisfied. For example, the task *GetMail* requires users to touch their cane as a precondition for getting the mail. Figure 9 shows the contingent cue that is generated if the system detects the user stepping outside without the cane.

Autonomous Agent Architecture: Our initial prototype for the CARE system uses the PEAT planning and cueing system. However, as we develop more of the CARE agent model and behavior, and real-time monitoring, we have more need for an full-fledged autonomous agent architecture with a sense, plan, act cycle.

Our CARE agent's autonomous system model is based on two systems developed at NASA Ames Research Center for autonomous robots. The PROPEL system shown in Figure 10 [Levinson 1995b; 2005] and IDEA [Muscettola et. al 2002; 2000] both provide methods for unified planning and execution in autonomous systems. The system also incorporates contingent execution methods based on the PLEXIL system, also developed at NASA [Verma et al., 2005].

The agent's unified planning and execution provides critical self-management and executive functions for the CARE agent itself, which is different from the need to compensate for the user's impaired executive functions. Like the user and a human caregiver, the CARE agent must have flexible planning and reactivity in order to maintain its own autonomy in changing situations. PROPEL and IDEA provide the starting point for that technological infrastructure.

Unified Planning and Execution – Most autonomous systems with integrated planning and execution components use different action representations for planning and execution. Usually, a formal, declarative, logic-based activity model is used for the planner but a procedural programming-language (like C or Java) is used for the execution model, as with the 3T system [Bonasso et al., 1997].

In contrast with this hybrid approach, Levinson proposes that tight integration between planning and execution requires *unification* of the planning and execution components, including unification of their activity models as shown in Figure 10 [Levinson, 1995b, Levinson 2005]. A key motivation for unifying the activity models used by planning and execution is to eliminate planner blind spots, so the planner can "see into" and reason about details of the execution system's model and state, and vice versa. With different action representations for planning and execution, the planner has only an abstract model of execution failure, which limits its ability to plan error recoveries. Other benefits of unified models include avoiding redundant model development efforts, and reducing the problem of keeping the two models consistent with each other.

To address these issues of unified planning and execution, Levinson developed the Procedure Planning and Execution Language (PROPEL) [Levinson 1995b; 2005], and co-developed the IDEA system [Muscettola et al. 2002, Muscettola et al. 2000]. IDEA is similar to Propel because they both use the same action representation for both planning and execution. However, IDEA uses a declarative representation for both function, while Propel uses a procedural representation for both functions. These autonomous agent methods have never fully been integrated into PEAT, but the addition of sensors and monitoring, has led to us incorporating more Propel and IDEA methods. As we flesh out the CARE agent model and integrate real-time activity monitoring, there is more need for an autonomous agent with its own executive functions.

PROPEL: Figure 10 shows PROPEL's unified Planning and Execution architecture for autonomous agents. The Agent is defined by a model which specifies its goals and behavior. For this application, the CARE Agent's goals include increasing user independence and therapy compliance. The CARE Agent's behavior is specified by a library of therapy and conversation procedures. PROPEL's procedural representation is particularly appropriate for the CARE application because conversation is inherently procedural, with loops and conditionals and state variables. CARE agent effectors include the phone's audio speakers and the graphical display.

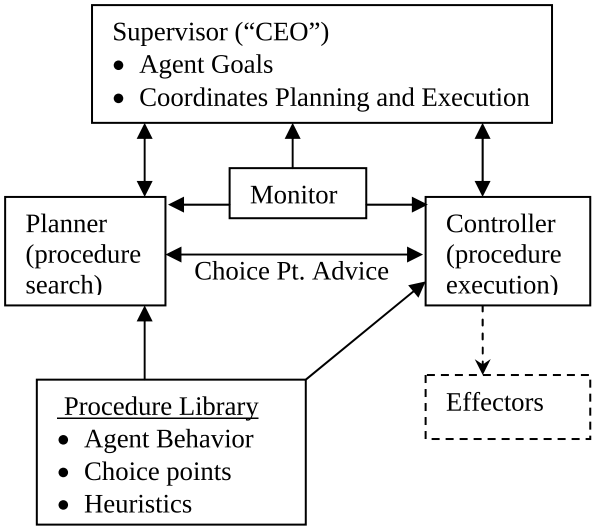


Figure 10: Propel's Autonomous Agent Architecture with unified Planning and Execution [Levinson 1995b; 2005]

Procedure Library (Activity Model): The Agent behaviors are defined by the Procedure library, and are consumed by the Planner and controller modules. A single Procedure Library (activity model) is shared by both the planning and execution components. Using the same procedure library for planning and execution enables the planner and controller to reason about procedural state of each other.

The Action Representation is procedural, with loops, conditionals, variables, and subroutines. Procedures also include *choice points* which specify options for choosing different subroutines and parameter values (Figure 11). Choice points identify steps in the procedure where different methods and resources may be selected. Choice points are nondeterministic subroutine calls or assignment statements which define a search space of procedure variations.

Procedures for the CARE application include cueing and conversations. For example, consider the CueUser task shown in Figure 11. This shows pseudocode for a proposed example rather than our existing implementation. However, implementation is well defined by prior Propel systems [Levinson 1995b; Levinson 2005]

The CueUser procedure (Figure 11) illustrates the use of choice points in the action representation. Choice points are shown in **Bold** text. The CARE agent may choose the maximum number of cue repetitions by choosing an integer between 1 and 10 (*maxCues, line 2*). The agent also chooses the cue message content (*content, line 5*), the *mode* (line 6) is the presentation modality (audio, visual, verbal, graphical), and the amount of time to wait before reprompting the user (*timeout, line 7*). This simple procedure with choice points produces a rich search space of cue variations

Search heuristics H1, H2, H3 and H4 are domain-specific "*LessThanOrEqual*" predicates used for sorting choices. A choice is "*lessThan*" the other if it is "better" than the other in a given domain context. Sorting criteria

```
 1  DefineTask CueUser (event) {
 2  int maxCues = ChooseInteger(1,10, H1);
 3   n = 0; done = false;
 4   While ((not done) and (n  <  maxCues)) {
 5      content = ChooseCueContent(n, event, H2);
 6      mode = ChooseCueMode(n, event, H3);
 7      timeout = ChooseTimeout(n, event, H4);
 8      GenerateCue(cntent, mode);
 9      response = GetResponse(timeout);
10      if (response is not "timeout")
11         then done = true;
12       else  n = n+1;
13    }// end while
14  } // end Task
```

Figure 11: CueUser is a hypothetical Propel procedure for cueing the user with choice points defining a search space of cue procedure variations.

include user preferences, planner advice, and execution results. The heuristics are designed to maximize achievement of the CARE agent goals which may be customized for each user. The heuristics may include planner advice about choices which may maximize goal achievement

Unified Planning and Control: The Planner and Controller both interpret the procedures from the library. The Planner explores a search space of procedure variations defined by the choice points and uses backtracking, while the Controller selects a single trajectory through the choice space in real-time without backtracking. The Planner and Controller both use "Choice point advice" heuristics to make choice point selections. Choice Point Heuristics include advice rules. The planner and controller exchange choice point advice rules which identify the choices each has taken and results of those choices (search success or failure/backtracking).

Planner searches procedure variations to maximize goal achievement. The planner takes procedures as input, monitor state updates, and choice point advice which is exchanged with the controller. The Supervisor sends start and stop messages to control the planner.

Backtracking in the planner space may be caused by hard domain constraint violations such as planning to visit a store after hours or trying to make a sandwich without bread. Soft constraints include preferring lunch before noon, and involves searching for the "best" scoring path as determined by a heuristic function as with A*. For CARE, the "best" choices are those that maximize user independence, therapy compliance, and quality of life. Simulation stubs" are used to simulate effector actions during planning so that the agent doesn't talk to itself out loud while planning conversations.

The Controller executes procedures in real-time. The controller takes as input procedures, monitor state updates, and choice point advice which is exchanged with the planner. The Supervisor sends start and stop messages to the controller.

The Controller does not require planner input because choices are made heuristically. During execution, heuristics are used to make real time selections at choice points. A default choice is made without deliberation if no planner choice is available. However, heuristic choices may be improved after the planner has evaluated the execution system's default choices in current context.

Execution Failures may be caused by hard failures like missing the bus or by soft constraints such generating inappropriate cues and therapy options which are out of sync with the user's current situation and annoy the user.

The **Supervisor** acts as Chief Executive Officer (CEO) for the whole system by coordinating the Planner and Controller. The Supervisor sends *start* and *stop* messages to the Planner and Controller, and it receives Success and Failure messages from the Planner and Controller.

The Supervisor may implement various "executive" strategies such as planning for 10 minutes before executing the procedure, or executing the procedure without any planning and then planning only if a failure occurs.

Acknowledgements. This work is funded in part by the following: DARPA contract W31P4Q-08-C-0170, ARMY contract W81XWH-08-C-0740.

References

[Bonasso et al., 1997] Bonasso, R, Firby R., Gat, E., Kortenkamp, D., Miller, D.,and Slack, M. Experiences with an Architecture for Intelligent, Reactive Agents, in Journal of Experimental and Theoretical Artificial Intelligence, January, 1997.

[Duong et al, 2005] Duong, T.; Bui, H.; Phung, D.; and Venkatesh, S. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In Proc. 2005 IEEE Computer Vision and Pattern Recognition 2005.

[Fisch et al., 2007] Fisch, J., Dowds, M., Levinson, R., Bushnick, T. 2007. Current Evidence-Base for Efficacy of Electronic Aids for Memory and Organization. American Congress on Rehabilitation Medicine, Washington DC. 2007.

[Liao et al., 2007] Liao, L., Fox, D., Kautz, H.. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields International Journal of Robotics Research, 2007.

[Liao et al., 2004] Liao, L., Fox, D., Kautz, H. Learning and Inferring Transportation Routines. Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, CA, 2004.

[Levinson, 2006] Levinson, R., 2006. Automatic Planning and Cueing System and Method. U.S. Patent # 7,027,996.

[Levinson, 2005] Levinson, R., Unified Planning and Execution for Autonomous Software Repair. ICAPS-05 Workshop on Plan Execution, Monterey, CA. 2005.

[Levinson, 2002] Levinson, R., 2002. An Automatic Planning and Cueing System and Method. Patent # 6,381,580.

[Levinson, 2000] Levinson, R., 2000. Intelligent Planning and Calendaring System with Cueing Feature and Floating Tasks. U.S. Patent # 6,047,260.

[Levinson, 1997] Levinson, R., The Planning and Execution Assistant and Trainer. Journal of Head Trauma Rehabilitation, Aspen Press. 1997.

[Levinson, 1995a] Levinson, R. A Computer Model of Prefrontal Cortex Function. Annals of the New York Academy of Sciences:Structure and Function of Prefrontal Cortex Vol. 769. 1995.

[Levinson, 1995b] Levinson, R. A General Programming Language for Unified Planning and Control. Artificial Intelligence, Vol. 76. Special Issue: Planning and Scheduling. 1995.

[Levinson, 1994] Levinson, R. Human Frontal Lobes and AI Planning Systems. Proc. of AIPS-94. AAAI Press. 1994.

[Modayil et al., 2008a] Modayil, J,, Bai, T., and Kautz, H. Tenth International Conference on Ubiquitous Computing (UBICOMP08), Seoul, South Korea, 2008

[Modayil et al., 2008b] Modayil, J., Levinson, R., Harman, C., Halper, D., Kautz, H. "Integrating Sensing and Cueing for More Effective Activity Reminders" AI in Eldercare AAAI Fall Symposium on AI in Eldercare, Washington, DC. 2008.

[Muscettola et al., 2002] Muscettola, N., G. A. Dorais, C. Fry, R. Levinson, and C. Plaunt, "IDEA: Planning at the core of autonomous reactive agents," in Proc. of the 3rd International NASA Workshop on Planning and Scheduling for Space, 2002

[Muscettola et al., 2000] Muscettola, N., Dorais G.A., Fry,C., Levinson, R., Plaunt, C. A Unified Approach to Model-Based Planning and Execution. Proc. of the 6th Int'l Conf. on Intelligent Autonomous Systems. 2000.

[Patterson et al., 2005] Patterson, D., Fox, D., Kautz, H., Philipose, M. Fine-grained activity recognition by aggregating abstract object usage. In Proc of IEEE Int'l Symp. on Wearable Computers, Osaka, Japan, 2005.

[Philipose et al., 2004] Philipose, M.; Fishkin, K.; Perkowitz, M.; Patterson, D.; Hahnel, D.; Fox, D.; and Kautz, H. Inferring activities from interactions with objects. IEEE Pervasive Computing: Mobile and Ubiquitous Systems, 2004.

[Rabiner, 1989] Rabiner, L.. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2):257–286. 1989.

[Verma et al., 2005] Verma, V., Jonsson, A., Simmons, R., Estlin, T., Levinson, R. Survey of Command Execution Systems for NASA Spacecraft and Robots. ICAPS-05 Workshop on Plan Execution, Monterey, CA 2005.