People, Sensors, Decisions: Customizable and Adaptive Technologies for Assistance in Healthcare

JESSE HOEY, University of Waterloo
CRAIG BOUTILIER, University of Toronto
PASCAL POUPART, University of Waterloo
PATRICK OLIVIER, Newcastle University
ANDREW MONK, University of York
ALEX MIHAILIDIS, University of Toronto

The ratio of healthcare professionals to care recipients is dropping at an alarming rate, particularly for the older population. It is estimated that the number of persons with Alzheimer's disease, for example, will top 100 million worldwide by the year 2050 [Alzheimer's Disease International 2009]. It will become harder and harder to provide needed health services to this population of older adults. Further, patients are becoming more aware and involved in their own healthcare decisions. This is creating a void in which technology has an increasingly important role to play as a tool to connect providers with recipients. Examples of interactive technologies range from telecare for remote regions to computer games promoting fitness in the home. Currently, such technologies are developed for specific applications and are difficult to modify to suit individual user needs. The future potential economic and social impact of technology in the healthcare field therefore lies in our ability to make intelligent devices that are customizable by healthcare professionals and their clients, that are adaptive to users over time, and that generalize across tasks and environments.

A wide application area for technology in healthcare is for assistance and monitoring in the home. As the population ages, it becomes increasingly dependent on chronic healthcare, such as assistance for tasks of everyday life (washing, cooking, dressing), medication taking, nutrition, and fitness. This article will present a summary of work over the past decade on the development of intelligent systems that provide assistance to persons with cognitive disabilities. These systems are unique in that they are all built using a common framework, a decision-theoretic model for general-purpose assistance in the home. In this article, we will show how this type of general model can be applied to a range of assistance tasks, including prompting for activities of daily living, assistance for art therapists, and stroke rehabilitation. This model is a Partially Observable Markov Decision Process (POMDP) that can be customized by end-users, that can integrate complex sensor information, and that can adapt over time. These three characteristics of the POMDP model will allow for increasing uptake and long-term efficiency and robustness of technology for assistance.

Categories and Subject Descriptors: I.2.1 [**Computing Methodologies**]: Artificial Intelligence

General Terms: Design, Human Factors, Algorithms

The reviewing of this article was managed by associate editor Eric Horvitz.
The authors thank the American Alzheimer's Association, the RCUK Digital Economy Theme, the National Science and Engineering Research Council of Canada, and the Canadian Institutes for Health Research for generous funding of this research. Also the authors thank Toronto Rehabilitation Institute-UHN for their continuing support.
Authors' addresses: J. Hoey (corresponding author), School of Computer Science, University of Waterloo, Waterloo, Canada, N2L3G1; email: jhoey@cs.uwaterloo.ca; C. Boutilier, Department of Computer Science, University of Toronto, Toronto, Canada; P. Poupart, School of Computer Science, University of Waterloo, Waterloo, Canada, N2L3G1; P. Olivier, Department of Computer Science, Newcastle University, Newcastle, UK; A. Monk, Department of Psychology, University of York, York, UK; A. Mihailidis, Department of Occupational Therapy and Occupational Science, University of Toronto, Toronto, Canada.

DOI 10.1145/2395123.2395125 http://doi.acm.org/10.1145/2395123.2395125

Additional Key Words and Phrases: Assistive technology, interactive systems, artificial intelligence, POMDP, prompting, dementia, stroke

ACM Reference Format:
Hoey, J., Boutilier, C., Poupart, P., Olivier, P., Monk, A., and Mihailidis, A. 2012. People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare. ACM Trans. Interact. Intell. Syst. 2, 4, Article 20 (December 2012), 36 pages.
DOI = 10.1145/2395123.2395125 http://doi.acm.org/10.1145/2395123.2395125

1. INTRODUCTION

This article describes a series of significant contributions to the field of intelligent interactive systems, specifically within cognitive Assistive Technology (AT). The contributions arise through summative work that has been completed over the past decade with respect to the application of machine learning to cognitive AT and a framework for using these techniques. In the 1990s, the area of cognitive AT was emerging with several different devices being developed. However, the majority of these devices were not adaptable, flexible, and did not take into account the dynamic needs and contexts of people with disabilities. As such, many of these devices were not successful. As this field continued to mature after the millennium, more developers started to realize that many of the limitations of past systems could be addressed through the use of more advanced techniques such as advanced sensing and machine learning. Researchers from both rehabilitation and computer science started to collaborate more and more on new cognitive AT that applied these and other algorithms in an effort to develop new devices that can automatically determine the needs of a user and adapt accordingly. While there was a significant new body of work in the application of sensing and machine learning to cognitive AT, there was still very limited research being conducted on the specific features and parameters that are required in these new advanced models, and how these models could be easily developed by users who do not have a technical background. This research described in this article provides the results of more than a decade of work to address these issues. Our work goes beyond any single interactive intelligent system to a methodology that addresses the larger context of Intelligent Interactive Systems (IIS) in the domain of assistive technology.

The ratio of healthcare professionals to care recipients is dropping at an alarming rate, particularly for the older population. It is estimated that the number of persons with Alzheimer's disease worldwide, for example, will top 100 million by the year 2050. These cases will prove an insurmountable economic barrier in the provision of care, unless steps are taken to reduce the need for personalized care now. Perhaps even more interesting is the expected increase in the hours of informal care provided for people with dementia. In Canada, for example, this number is expected to triple (from 231 million hours to 756 million hours) by the year 2038 [Alzheimer's Society of Canada 2010]. This implies that the burden of care is shifting from the professional arena (e.g., hospitals and clinics) into the home and community. As more people start taking control of their own healthcare decisions, they will require additional help.

Intelligent Interactive Systems (IIS) can play a key role in healthcare assistance in the home, primarily by connecting providers and recipients and by increasing the range or scope of care provision. A physical therapist can monitor a large number of clients working on their rehabilitation programs at home, without having to be present all the time. A nurse can monitor her patients aging at home, and can provide assistance more readily to those in need at appropriate times. Unfortunately, most current technology for home care is developed for specific applications, and is difficult to modify to suit individual user needs. The economic and social impact of IIS for healthcare lies in three key factors.

—*People / Customizability.* In order for people to have control over intelligent systems, the systems need to be customizable. The concept of *inclusive design* only works for fairly large and uniform segments of the population, but fails for users with diverse and changing needs. Persons with Alzheimer's disease, for example, have needs that change as the disease progresses. Giving such persons (or their carers, family members) control over technology to help them can greatly increase the benefits.

—*Sensors / Generalizability.* In many healthcare situations, sensors must be used to provide valuable information for carers and health professionals. Examples include monitoring patients in hospital or at home, smart-homes to assist persons in independent living, and emergency response systems that detect and respond to falls. In order to ensure uptake of IIS by informal carers and users, sensors must be installed for specific purposes. However, it becomes an economic burden if the installation of sensors needs to be done for each individual and each situation. Instead, we propose that general-purpose sensors can be used, if coupled with appropriate Machine Learning Algorithms (MLAs). These MLAs can learn from observing a person, and can provide virtual sensing capabilities for technological solutions. Learning these virtual sensors enables the IIS to generalize across tasks, users, and environments.

—*Decisions / Adaptivity.* People require assistance with their healthcare needs, but their requirements can change over time. While customization is one element of allowing people to make IIS fit their needs, the system itself also must be able to adapt to people over time as they change. This is particularly true for persons who have limited cognitive abilities (e.g., persons with Alzheimer's disease), who may not be able to take advantage of customization properties directly. Therefore, technological solutions must be able to take decisions about assistance for people, and these decisions must be adaptive to people as they change.

This article will present our work over the last decade in developing interactive intelligent systems for healthcare applications. Central to our work is the development of a unifying model that combines the three important elements described before. The model is based on the Partially Observable Markov Decision Process, or POMDP [Åström 1965; Kaelbling et al. 1998]. The POMDP is a probabilistic, decision-theoretic model of a system interacting with its environment, and is introduced in Section 3. In Section 4, we show how a general-purpose POMDP model of assistance can be formulated, and how this model captures the three key factors given earlier. We then present our work in applying this model to five intelligent interactive systems in Section 5. The first is a system to help persons with dementia during handwashing [Hoey et al. 2010b; Mihailidis et al. 2008]. The second is a device for stroke rehabilitation in the home [Kan et al. 2008, 2011]. The third is a tool for art therapists to use during sessions with persons with Alzheimer's disease [Blunsden et al. 2009; Mihailidis et al. 2010]. The fourth is a system for automatically generating cognitive assistants based on psychological analysis of tasks [Hoey et al. 2011], and the last shows how hierarchical controllers for prompting systems can be formulated using the same method [Hoey and Grzes 2011].

Throughout the article, we refer to *end-users* as our target population that we wish to build systems for (persons with dementia, their caregivers, family members, health professionals, etc.). However, at our current stage of research, we also add engineers and computer scientists to this list, as our methods still require their input. While our eventual goal is to remove dependence on these people, we are not there yet. Some of our systems have involved customizations by real end-users (see, e.g., the art therapy device in Section 5.3), and all have been tested with real end-users (e.g., persons with dementia). Further, we describe methods for learning parts of the model from end-users, including elicitation of preferences in Section 3.2.

2. PREVIOUS WORK

Over the past several years, there has been an increase in the number of new Assistive Technologies (AT) for healthcare that have used machine learning and artificial intelligence techniques [LoPresti et al. 2004; Pollack 2005]. These AT have primarily focused on users with cognitive impairments, helping these users to compensate for their own lack of planning and decision making abilities. These new AT have been labeled Cognitive Assistive Technology (CAT), and aid people with a variety of disabilities, including traumatic brain injury, cerebrovascular accident, learning disabilities, multiple sclerosis, and dementia [LoPresti et al. 2004]. The following is a brief overview of some of the key projects related to the application of probabilistic and decision-theoretic models in cognitive assistive technologies. For a more detailed and expansive overview of the different types of AT that have used artificial intelligence, the reader is referred to LoPresti et al. [2004], Pollack [2005], and Mihailidis et al. [2011].

The COACH (Cognitive Orthosis for Assistive aCtivities in the Home) is one of the longest-running projects in the area of cognitive assistive technologies for older adults with dementia. The primary goal of this project is to develop a system that can monitor and prompt an older adult through a variety of activities of daily living. To date, several iterations of this technology have been developed and tested with this population for the specific activity of handwashing. COACH employs computer vision to track a user and objects of interest as s/he performs an activity of daily living and provides audio and/or visual assistance if, and only if, the user requires it (e.g., s/he performs a step in the handwashing activity out of sequence, gets sidetracked, or is not sure what to do next). A Partially Observable Markov Decision Process (POMDP) is employed to learn characteristics about each individual over time, such as what his/her level of independence is, what type of prompts are most effective with him/her, and the average amount of time it takes him/her to complete each step in handwashing. Not only does this approach enable COACH to customize guidance to each individual's needs, but also allows the system to adapt over time to changes in individuals' responsiveness and capabilities. More details on COACH can be found in Section 5.1.

Autominder is a system to help users remember the tasks that need to be completed. This system differs from COACH, in that it provides reminders of activities at a high level as opposed to providing prompts about how to complete the activity itself. Autominder serves as a cognitive orthotic, providing its users with reminders about their daily activities. Most existing cognitive orthotics mainly issue alarms for prescribed activities at fixed times that are specified in advance. In contrast, Autominder is capable of much more flexible, adaptive behavior. It models its client's daily plans, tracks their execution by reasoning about the client's observable behavior, and makes decisions about whether and when it is most appropriate to issue reminders [Pollack 2006]. Autominder's architecture has three main components, one dedicated to each of these tasks. The Plan Manager stores the client's plan of daily activities in the Client Plan, and is responsible for updating it and identifying and resolving any potential conflicts in it. The Client Modeler uses information about the client's observable activities to track the execution of the plan, storing beliefs about the execution status in the Client Model. A reminder generation component called the Personal Cognitive Orthotic reasons about any disparities between what the client is supposed to do and what she is doing, and makes decisions about whether and when to issue reminders [Pollack 2006]. Autominder has been deployed on three separate platforms: the robotic assistant Pearl, a robotic walker, and on a handheld PDA.

PEAT (the Planning and Execution Assistant and Trainer) is a cognitive orthosis that runs on a cell phone and helps compensate for executive function impairment. Similar to Autominder, PEAT provides assistance by maintaining a schedule of a user's activities and automatically cueing the user when activities need to be started, resumed, or completed [Modayil et al. 2008b]. PEAT uses both planning and activity recognition algorithms to determine what a user is completing, and then to schedule (or reschedule) activities as necessary. Activity recognition on the PEAT is based on an HMM [Modayil et al. 2008b]. Finally, the PEAT system also has the ability to deal with potentially interleaved tasks (i.e., more than one task that may overlap in their executions) using the Interleaved HMM (IHMM) [Modayil et al. 2008a] as a better variant of an HMM for the classification of interleaved activities.

A key aspect of all of the technologies presented in this section (and others) is the ability for these systems to recognize the activities that are being completed. The PROACT system is an example of a technology that uses machine learning for the specific purpose of inferring tasks being completed based on sensor inputs. This system has three components: body-worn RFID sensors, a probabilistic engine that infers activities given observations from these sensors, and a model creator that easily creates probabilistic models of activities [Philipose et al. 2004]. PROACT uses a Dynamic Bayesian Network (DBN) to infer activities that represent daily activities such as making tea, washing, and brushing teeth. Each activity type that PROACT is intended to recognize is modeled as a linear series of steps, where specific objects that are involved in the completion of a step and the probability of seeing each such object are assigned. Recent work in the same direction has investigated how activities can be modeled with a combination of discriminative and generative approaches [Lester et al. 2005], and how common-sense models of everyday activities can be built automatically using data mining techniques [Pentney et al. 2007, 2008], and how human activities can be analyzed through the recognition of object use, rather than the recognition of human behavior [Wu et al. 2007]. This last work uses DBNs as well to model various activities around the home, and a variety of Radio Frequency identification (RFID) tags to bootstrap the learning process.

The Assisted Cognition project [Kautz et al. 2002] was initiated to explore the use of artificial intelligence as a tool to increase the independence and quality of life of Alzheimers patients. One of the first devices developed as part of the project was the Activity Compass [Patterson et al. 2002], a tool designed to help disoriented people find their destination. An extension of the Activity Compass project is Opportunity Knocks [Liao et al. 2007], a system designed to provide directional guidance to a user navigating through a city, and was developed in the Assisted Cognition project.

The system infers a user's activities using GPS sensor readings and a hierarchical Markov model. Movement patterns, based on the GPS localization signals, are translated into a probabilistic model using unsupervised learning. From the model and the user's current location, future destinations and the associated mode of transportation can be predicted. Based on the prediction, the system has the ability to prompt the user if an error in route is detected.

Smart-home systems [Zhang et al. 2008; Singla et al. 2008; Brdiczka et al. 2009; Abowd et al. 2002; Chen et al. 2005, 2011; Bharucha et al. 2006; Mozer 2005; Intille 2006; Bharucha et al. 2009] attempt to provide assistance to residents across a wide range of tasks using an array of sensors in the environment, algorithms for the analysis of human behavior, and controllers for the provision of the assistance. However, these systems do not take a unifying view of the entire process, from specification through implementation to control of assistance.

There has been much work on modeling human behavior in the computer vision literature [Moeslund et al. 2006], which includes supervised techniques to build models of meeting dynamics [Rybski and Veloso 2004], office activity [Nguyen et al. 2003], and other in-home activities [Hamid et al. 2003].

Partially Observable Markov Decision Processes (POMDPs) provide a rich framework for planning under uncertainty [Åström 1965]. In particular, POMDPs can be used to robustly optimize the course of action of complex systems despite incomplete state information due to poor or noisy sensors, for instance, in mobile robotics [Pollack 2006; Pineau et al. 2003; Montemerlo et al. 2002], in computer vision [Darrell and Pentland 1996], in spoken-dialog systems [Williams and Young 2006; Roy et al. 2003], and in assistive technology [Mihailidis et al. 2008]. As well as the applications described in Section 5, we have applied the model described here to mixed-initiative wheelchair control [Mihailidis et al. 2007], to calendar scheduling, to assistance of cognitively disabled persons in a factory assembly task, to toothbrushing assistance, and to location assistance (works in progress).

3. POMDPS

In this section, we give a brief overview of Partially Observable Markov Decision Processes (POMDPs) since they will be used in subsequent sections to model various task assistance problems. POMDPs provide a principled and flexible framework to model sequential decision-making problems where there is uncertainty in the action effects and noise in the sensor measurements. A discrete-time POMDP consists of the following.

—A finite set S of states. In general, states are sufficient statistics that capture all the necessary information to determine how the system will evolve in the future;
—A finite set A of actions. In general, actions consist of the controls that the system can choose from to influence future states. They represent the set of possible decisions of the system (not the user);
—A stochastic transition model $\Pr : S \times A \to \Delta(S)$. This model defines a distribution $\Pr(s'|s, a)$ that indicates the probability that the system will transition to state s' after executing action a in state s. This distribution satisfies the Markov property, which says that states at a current time step depend only on states at the previous time step, but not earlier time steps;
—A finite observation set O. Observations consist of the bits of information received by the environment at each time step. More precisely, the observations may consist of the measurements produced by some sensors. In general, O and S are different since states are not assumed to be fully observable.
—A stochastic observation model $Pr : A \times S \to \Delta(O)$. This model defines a distribution $\Pr(o'|a, s')$ that indicates the probability of making observation o' after executing action a and arriving in state s'. This distribution quantifies the noise in the sensor measurements and therefore how observations correlate with states;
—A reward function $R(s, a, s')$. This function assigns a reward r (or utility) to transitions from s to s' induced by action a. The reward function provides a general mechanism for quantifying goals in terms of local transitions.

All together, the aforesaid components define a decision process which can be used to model a wide range of problems from inventory management to spoken dialog systems, including assistive tasks as we will see in more detail in the next section. Figure 1(a) shows two consecutive time slices ($t - 1$ and t) of a generic POMDP as an influence diagram. Here circles denote random variables (states and observations), rectangles denote decision variables (actions), and diamonds denote utility variables (rewards).[1]

Given a POMDP, the goal is to select actions that maximize the rewards. If the states were fully observable, the problem could be simplified by searching for a policy

[1]We abuse notation a bit by using the same symbols to denote variables and their domains.

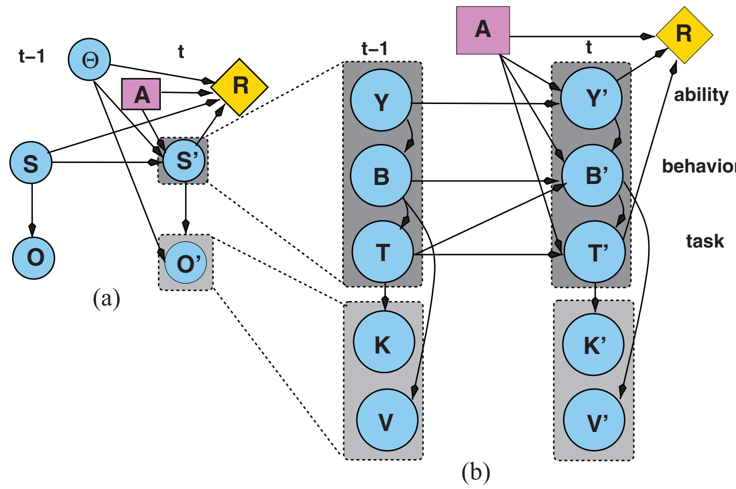

Fig. 1. Two time slices of: (a) a general POMDP and (b) a factored POMDP for modeling interactions with cognitive assistive technology. The state, S, is modified by action A, and produces observation O. We leave off links from actions, A, to observations, O, and the parameters, Θ, in (b) for clarity.

that consists of a mapping from states to actions. However, many problems, including task assistance, have state features that are not directly observable, so it is generally better to consider policies that are mappings from *beliefs* (i.e., state distributions) to actions. At every time step, the system receives an observation that is correlated with the underlying state and therefore it is possible to infer a belief based on the history of past actions and observations. Let $b(s)$ be the belief of the system at the current time step. Then at the next time step, an updated belief $b_{a,o'}$ can be computed after executing action a and making observation o' according to Bayes' theorem.

$$b_{a,o'}(s') \propto \sum_{s} b(s)\,\Pr(s'|s,a)\,\Pr(o'|a,s')$$

Let $\pi : \Delta(S) \rightarrow A$ be a policy that maps beliefs in $\Delta(S)$ to actions in A. A common objective to determine the value $V^{\pi}(b)$ of a policy π starting from belief b is the sum of discounted expected rewards for an infinite horizon.

$$V^{\pi}(b) = \sum_{t=0}^{\infty} \gamma^t E_{\pi}[R(s_t, a_t, s_{t+1})]$$

Here γ denotes the discount factor (between 0 and 1), which can be thought as the weight by which future rewards are scaled down at each time step for a corresponding inflation rate. Since rewards depend on transitions, which are stochastic, the expectation (denoted by E) of each reward is often used in the overall criterion. The best policy π^* is the one that yields the highest value V^* for all beliefs (i.e., $V^*(b) \geq V^{\pi}(b)\ \forall b, \pi$). It can be shown that the optimal value function V^* satisfies Bellman's equation, which can be viewed as a recursive decomposition of V^* in terms of the immediate reward R plus the discounted future rewards (captured by V^* again).

$$V^*(b) = \max_a \sum_{s,s'} b(s)\,\Pr(s'|s,a)\left[R(s,a,s') + \gamma \sum_{o'} \Pr(o'|a,s')V^*(b_{a,o'})\right]$$

3.1. POMDP Solutions

Bellman's equation naturally leads to a dynamic programming technique known as *value iteration* that repeatedly updates an estimated value function by performing the operations in the right-hand side of the equation until convergence to V^*. Alternatively, it is possible to directly optimize a policy by *policy iteration*, or more generally, *policy search*. Instead of precomputing a policy, it is also possible to perform a *forward search* at runtime between each action execution to find the next action to execute. The reader is invited to consult Poupart [2012] for more details on POMDPs and an overview of solution algorithms. Current research has enabled the approximate solution of very large POMDPs [Poupart and Boutilier 2004; Hoey and Poupart 2005; Silver and Veness 2010].

To optimize a policy, we used SymbolicPerseus [Poupart 2005][2]. It implements a factored, point-based value iteration technique based on the Perseus algorithm [Spaan and Vlassis 2005]. There are many other methods for solving POMDPs, including recent successful Monte-Carlo methods [Silver and Veness 2010], but we avoid digressing from the main goal of our work, which is to provide customizable, adaptive, and general-purpose solutions for assistance.

3.2. Learning POMDPs

So far we have described how sequential decision making problems can be formulated as POMDPs for which optimization algorithms can be used to automatically find a good policy. However, how do we obtain a POMDP model in the first place? More precisely, how do we obtain the transition and observation distributions as well as the reward function?

In general, the transition and observation distributions can be specified by hand, learned from data, or a mixture of both. In Section 5.4, we describe a tool (SNAP) that eases the specification of those distributions by hand for task assistance problems. In general, since the transition and observation dynamics of a POMDP are nothing more than an input-output Hidden Markov Model (HMM) [Bengio and Frasconi 1996], it is possible to use learning techniques for HMMs. More specifically, the problem is to infer $\Pr(S'|S, A)$ and $\Pr(O'|A, S')$ based on a sequence of actions, observations, and possibly labeled states. Since the states are not directly observable we may not have their sequence, but in some applications it may be possible to get an expert to label them by hand. In that case, we have a supervised learning problem since all variables are known. The transition and observation distributions that maximize the likelihood of the data can be computed in closed form and correspond to the relative frequency counts of the data. When the sequence of states is not available, unsupervised or semisupervised learning algorithms such as Expectation Maximization [Dempster et al. 1972] can be used to obtain good estimates of the transition and observation distributions. It is also possible to estimate the number of hidden states [Beal et al. 2002] and for factored models, to estimate the number of hidden variables, their domain size, and their dependencies [Doshi-Velez et al. 2011].

Given a specific set of state and observation variables, transition and observation models are, in principle, objectively learnable. By contrast, the reward function reflects the preferences of the end-user for specific states or particular behaviors. Learning reward functions for MDPs from observation is a problem that has been considered in the literature (e.g., Ng and Russell [2000]), but generally requires that one has access to trajectories generated by an *expert*, that is, an actor who is implementing an optimal policy for the MDP in question. In our setting, this is unrealistic, in part because the

[2]www.cs.uwaterloo.ca/~jhoey/research/spudd.

actions available to an intelligent system may be different than those available to end-users (e.g., caregivers), and in part, because expert policies are unlikely to be optimal in the POMDP sense or even cover the range of possibilities required.

As a consequence, the explicit *elicitation* of reward functions is most appropriate in most cognitive assistance domains. Elicitation of utility functions has a long history in decision analysis [Keeney and Raiffa 1976; Salo and Hämäläinen 2001; Abbas 2004] and has attracted considerable attention in AI as well [Chajewska et al. 2000; Boutilier 2002, 2006; Braziunas and Boutilier 2010], with one of the primary aims being the development of interactive systems that elicit utility information effectively by identifying "irrelevant" aspects of a utility function that can be left unspecified. The elicitation of reward functions for MDPs (and POMDPs) has received somewhat less attention, though recent work has considered means for interactive and partial elicitation of both unstructured reward functions [Regan and Boutilier 2009, 2011b] and additive reward functions [Regan and Boutilier 2011a] for fully observable MDPs. The range of queries and feedback one can obtain from a user to construct a reward function is quite rich. It includes: *state* or *attribute queries*, which directly query a user for values of (or bounds on) the reward associated with a specific state or state variable (action costs can be queried in similar fashion); *state/attribute* comparisons, which ask users to state which of two states (or attribute values) is preferred, or to rank a set of states; *trajectory comparison queries*, which ask a user to compare two state-action trajectories or policy demonstrations; and *policy critique*, where a user is shown a policy demonstration, and asked to suggest points at which a different action might have been preferred.

The SNAP method discussed in Section 5.4 also includes tools for the straightforward specification of POMDP reward functions for specific task domains, exploiting the fact that rewards are generally decomposable with respect to specific task goals and subgoals. However, the current form of these tools can only accommodate simple specification that may not be "natural" for end-users. The work referenced before gives us techniques to elicit reward functions in a more "user-friendly" way.

4. MODELING ASSISTANCE

A typical task requiring assistance consists of seven principal elements. We discuss these elements here in the context of the handwashing task for people with dementia (e.g., Alzheimer's disease), who typically require assistance from a human caregiver to wash their hands. The same elements appear in a wide range of assistance tasks. A person with AD loses short-term memory, and therefore has difficulty in *recalling* what they are doing, in *recognizing* necessary objects like the soap, and in *perceiving* affordances (environmental cues for action). An example of this is shown in Figure 2, which shows key frames from about 60 seconds of a video of a user washing their hands, assisted by a human caregiver.

4.1. Seven Elements of Assistance

An assistance task is described by a seven-tuple $\{T, B, Y, A, O, \Theta, R\}$, as follows.

Task, T. This is a characterization of the domain in which the task is taking place in terms of a set of high-level variables. For example, handwashing can be described by task states that describe whether the hands are wet or dry, dirty, soapy, or clean, and whether the water is on or off. In Figure 2, the user's hands are soapy and wet at frame 1745, but become clean and dry by frame 2515. These variables need to be specified for each task, but they characterize the world at a sufficiently high level to make this accessible to end-users.

Behavior, B. The task states are changed by the user's *behavior*, B. Common behaviors during handwashing may be things like rinsing hands or using towel, as in Figure 2.

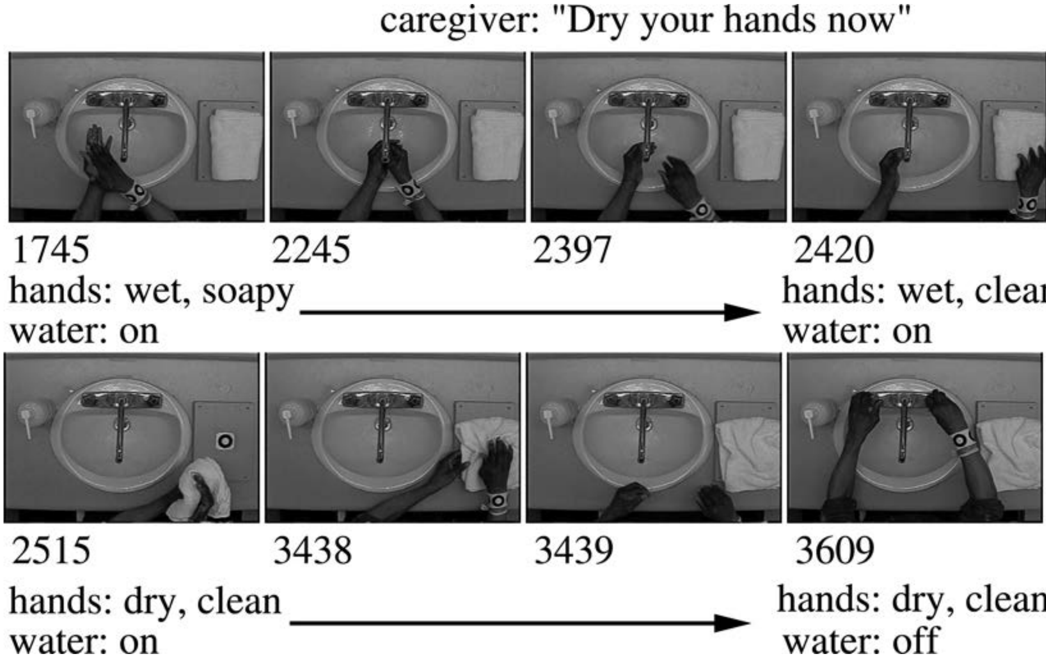

Fig. 2. Example sequence in which a user is prompted to use the towel to dry her hands.

The user's behavior evolves depending on the ability and the task as well as the system's action, A. The behaviors are the most difficult to manually specify, but can be learned from data.

Ability, Y. Variables describe the cognitive or physical state of the user. This may include the level of dementia and the current responsiveness, or perhaps the level of frustration the user is experiencing with the system, her overall level of health, or the physical ability with respect to the task. The *ability* variables essentially describe internal properties of the user that generalize across tasks, and would be specified by experts for the general system, and then carried over and possibly adapted for each task through learning. The user's expeditious reaction to the prompt in Figure 2, for example, might give us an indication that she is responsive, and is attending to the prompting system. Over a longer time period, the user's overall level of health may change. For example, her medical condition might take a turn for the worse, requiring attention from a professional. Such a change may be noticeable in her responses and behaviors.

Action, A. These are the actions of the system modify the behavior, ability, and the task. These actions could be verbal prompts, calls to human caregivers or other response systems, or physical changes to the environment. During handwashing, these actions are typically verbal prompts or reminders such as the "Dry your hands now" in Figure 2. However, as we will show, actions can include more general dialog iterations, calls to other response systems, or physical control of related systems.

Observations, O. *Task* and *behavior* variables generate observations, K and V, respectively. We use $O = \{K, V\}$ to denote the complete set of observation variables. We are positing two state space factors $\{T, B\}$ that independently give rise to different observation sets. Thus, the observations K are independent of B given T (e.g., a water switch sensor only depends on whether the water is running or not, independently of what the user is doing), and the observations V are independent of T given B (e.g. a gesture recognition system only depends on what the user's hands are doing, not on the result of her actions). These observations are generated from either noninvasive sensors, such as cameras (in which case the observations are video streams), microphones (audio streams), and environmental switches such as thermostats, or from invasive sensors, such as buttons, manual switches, locks, EEGs, etc.

Parameters, Θ. They describe the dynamics and the observation function, and govern how the first five elements interact. In a probabilistic model, these parameters are the conditional probability distributions over some subsets of elements given the rest, and the reward function. For example, we may know that a person who is responsive to audio prompts is likely to do what she is asked to do, if appropriate to the situation. This likelihood can be characterized by a parameter, the setting of which impacts the performance of the model.

Reward, R. Each {state,action,previous-state} combination has some value, given by a reward (or cost) function. The reward function's job is to specify the relative values of each possible outcome. The system will take actions that optimize over the reward function in the long term. A typical such trade-off is between achieving a small reward immediately, versus a large reward further in the future. The handwashing assistant, for example, can incur a small cost in prompting, but this small cost may pay off eventually when the person gets her hands clean.

Our goal is to design a model of the interactions between these elements, to build a method for customization of the model, and to optimize an automated strategy for assistance by maximizing (over the actions) some notion of utility over the possible outcomes. The model must be able to deal with uncertainty in the effects of actions and in sensor measurements, it must be able to tailor to specific individuals and circumstances (adaptivity and customizability), it must be able to trade off various objective criteria (e.g., task completion, caregiver burden, user frustration, and independence), and it must be easy to specify. A Partially Observable Markov Decision Process (POMDP), a decision-theoretic model which has recently received much attention in the AI community, fulfills these constraints (see Section 3).

4.2. POMDP Model of Assistance

To build a POMDP model of assistance, we need to map the elements defined in the last section onto the POMDP model from Section 3. In general, we model the *task* as a consequence of the *behavior* of the user, which is a reaction to the *actions* of the caregiver, conditioned by the *ability* of the user. The *behaviors* and the *task* are not directly observable, but can be inferred from some *observations* from sensors. All these dependencies are conditioned on the *parameters* of the model.

We claim that the *task* will be simple to specify, and can be defined by a nonspecialized person, while the *ability* will require expert knowledge, but will tend to generalize across tasks. On the other hand, the *behaviors* will be much more difficult to specify, but can be learned from data directly [Peters et al. 2009; Hoey et al. 2005], including a model of how they are related to the *observations*. The *rewards* must be specified by end-users, and finding a suitable reward function is a very challenging problem, typically addressed by preference elicitation methods [Keeney and Raiffa 1976; Regan and Boutilier 2011a]. The *actions*, *observations*, *rewards*, and *parameters* of the model can be specified by end-users, so long as we provide appropriate abstractions for them to encode these elements.

Figure 1(b) shows the same model as in Figure 1(a), except that the state, S, has been factored into three sets of variables: *task* (T), *ability* (Y), and *behavior* (B). Here we describe each of these sets, as well as the actions of the system, A, and the observations from which the state of the model is inferred.

Jointly, $S = \{T, B, Y\}$ is known as the state. The transition function can be factored as

$$\begin{aligned} Pr(S'|S, A) &= Pr(T', B', Y'|T, B, Y, A) \\ &= Pr(T'|B', Y', T, B, Y, A)Pr(B'|Y', T, B, Y, A)Pr(Y'|T, B, Y, A) \\ &= Pr(T'|B', T, A)Pr(B'|Y'B, T, A)Pr(Y'|Y, A) \end{aligned}$$

which is a product of three terms, as follows. $Pr(B'|Y', B, A)$ gives the expected behavior of the user given the previous state and the system action. $Pr(Y'|Y, A)$ gives the expected user state given the current behavior, the previous ability and task states, and the system action. $Pr(T'|B', T, A)$ gives the expected task state given the current behavior and the previous task state. Notice that the task state is independent of the ability, Y, given the behavior, B. The idea is that changes in the *task* states are caused by the *behaviors* of the user (and possibly the system actions), independently of the user's ability given the behaviors. The observations $O = \{K, V\}$ are generated by the *task* and *behavior* variables, T and B, respectively, through some observation functions $Pr(K|T)$ and $Pr(V|B)$.

In general, the time scales at which observations occur will be of much shorter duration than those at which task or ability change. Observations of behaviors will typically be frames from some video camera (at 30 Hz), or some segments of an audio stream (at 10 kHz), whereas the task states will only change every few seconds. For example, during handwashing, a typical behavior may be "drying hands", which may take 30 seconds to perform, and result in nearly 1000 video frame observations (e.g., Figure 2), but only cause a single change of the *task* state: the hands become "dry". Thus, the observation functions may introduce some hierarchical structure into the model. On the other hand, observations can be grouped together using some offline, heuristic method that produces a *virtual sensor*. An example is a computer vision algorithm that detects faces [Viola and Jones 2004], variants of which can be found on most modern consumer digital cameras. This algorithm can be seen as a black box that maps video frames to a single bit (face/nonface) that can be treated as an observation by disregarding any noise or confidence information[3] (see Section 5.3).

POMDPs can be used to monitor a person's progress in a task by using Bayes' rule to update a *belief state*, $b(s)$, that gives a probability that the model is in state s. The progression of this belief state through time is what the POMDP attempts to optimize by taking actions that lead to belief states with high reward values.

4.3. System Elements

We have seen how the POMDP can provide adaptive assistance by modeling appropriate elements of the state space and by dynamically monitoring beliefs. This section will present a general-purpose framework for integrating a POMDP into an assistance task, including the customization of the POMDP and system by end-users and the real-time running of the final working product. In this exposition, we point to our work in Section 5 as demonstrations of these techniques.

To allow for customization, we explicitly model the triadic relationship between a *client* (person needing assistance), a carer (*actor*), and a piece of technology with artificial intelligence capabilities (*agent*). This triad is modeled as a two-level system, where the *actor* controls the *agent* to assist the *client* while simultaneously reducing the *actor*'s load, maintaining the *client*'s safety, and promoting the *client*'s independence. The challenge of modeling such a triad is the differing levels of expertise. We wish to give the *actor* the ability to customize the *agent*, but without the need for extensive

[3]Confidence information can be included by adding yet more observations corresponding to the confidences.

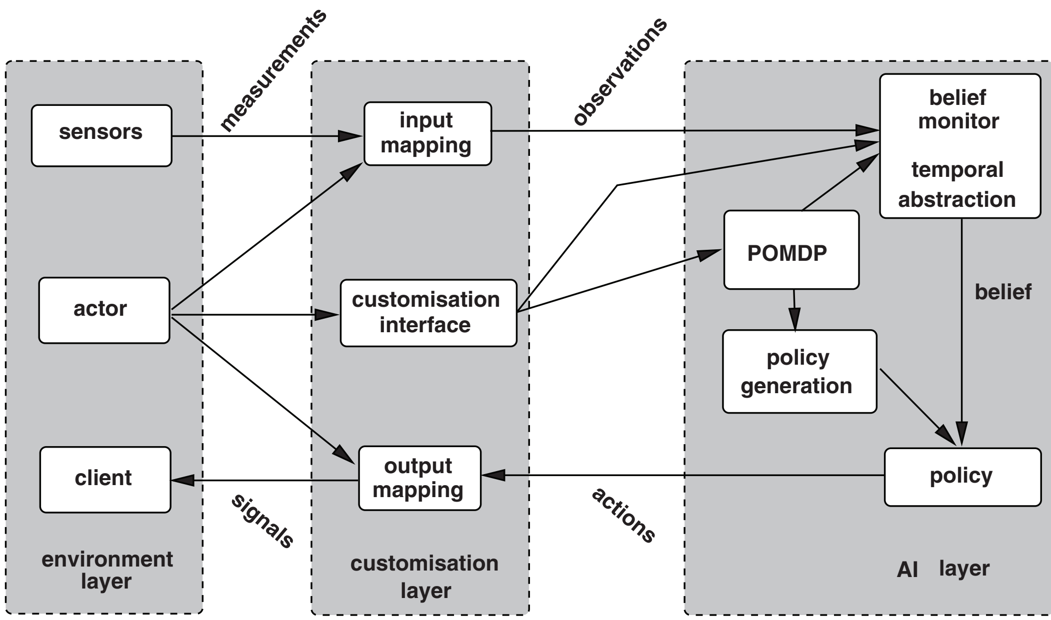

Fig. 3. System framework for assistance showing three layers: environment, customization, and artificial intelligence (agent) layers.

training. The customization should be simple and effective, and the end result should be helpful for the *actor* and adaptive to the *client*.

The framework in Figure 3 is divided into three *layers*. The *environment layer* on the left includes the *client* in need of assistance, an *actor* (e.g., a human assistant), and a set of sensors that measure various elements of the environment. The *actor* and *client* can be one and the same, or two different people or groups. The *customization layer* in the middle includes three elements that can be specified by end-users: a *customization interface* uses a suitable level of abstraction to give access to the POMDP parameters and temporal models, a set of *input mappings* from environment sensor inputs to abstract observations, and a set of *output mappings* from abstract actions to signals in the environment. We will show examples of these customizations in Sections 5.3 and 5.4. The third layer is the *AI layer* on the right, and includes the POMDP model parameters and policy, and a *belief monitor* which maps observations to belief states.

The *AI layer* also includes a method for abstracting over time. This is an important component, as the sensor measurements may be occurring at a very fast rate (e.g., 30 frames per second in video), whereas the POMDP operates on *event-driven* time. Updates in the POMDP need to happen at realistic and natural time frames for the *behaviors*. For example, during handwashing, it takes only a few seconds to turn on the water, but 10–30 seconds to rinse the soap. These timings are dependent on each task and on each user, and so must be specified as part of the customization. Note that some temporal abstraction can be present in the input mappings, but some may depend on the belief state itself, and so these elements are placed in the AI layer in Figure 3. The simplest possible method for temporal abstraction is a function, $\texttt{timeout}(s)$, which gives a timeout (a number of seconds) for each state. This function is consulted by the belief monitor after each new observation is received from the input mappings, by computing a final timeout value, τ, based on the current belief state, $b(s)$. This

Table I. Properties of Each Application, Broken Down into State Variables, Actions, Input/Output Mappings, and Customizability

	COACH	iSTRETCH	ePAD	SNAP
task	plansteps (a-k)	game state	screen state	cup position, cup contents, box condition
behavior	use soap, rinse, taps on/off, dry	time to target, control, compensation	interactive, active, intermittent, inactive	cup to ws, open box, tb to cup, close box
ability	awareness, responsiveness, dementia level	fatigue, range, learning rate	engagement	recognition, recall, affordance
sensors	video camera	time, posture, device rotation	touch screen, video camera	RFID in cup, box-lid switch, RFID in teabag
system actions	use soap, rinse, taps on/off, dry	set range, set resistance, stop	high interactivity, medium interactivity, low interactivity	prompt recognition, prompt recall, prompt affordance
input mapping	computer vision tracking of hands and towel, fixed regions	time limits, control, compensation	interactivity level	virtual sensors
output mapping	audio-visual prompts, specificity levels, caregiver calls	resistance mapping, distance mapping, game state	interactivity level of prompts	audio-visual prompts, direct indications
customisability	none	input/output mappings, parameters	input/output mappings, parameters	input/output mappings, parameters, POMDP structure

Null actions and behaviors are not shown.

timeout can be computed as a weighted average, $\tau = \sum_s b(s) \cdot \mathtt{timeout}(s)$, or as the most likely state's timeout, $\tau = \mathtt{timeout}(\arg\max_s b(s))$. The belief monitor then updates the POMDP whenever: (1) the belief state would change significantly after the update, (2) some observation changes, or (3) the timeout is reached.

5. APPLICATIONS

This section presents five applications of the model we have presented earlier. The first two are focused on the POMDP model and integrated system for two specific assistance scenarios: handwashing for dementia and stroke rehabilitation. These systems, however, include little or no customizability by users, but instead focus on adaptivity and demonstrating the POMDP method for two different tasks. The next two application areas are a device for art therapy and a method for building POMDP prompting systems directly from prior knowledge. These applications include user customization as a central issue. The last application demonstrates how the same model can be used for hierarchical control in assistive systems. Table I details the various elements of the four applications: their state variables (task, behavior and ability), system actions, input/output mappings, and their customization abilities.

5.1. Prompting for ADL: COACH

COACH is a real-time intelligent system for assisting persons with dementia during handwashing. The COACH system has been designed and built over the past ten years through a series of four prototypes, each relaxing restrictive assumptions present in its predecessor [Mihailidis et al. 2001; Boger et al. 2005; Hoey et al. 2010b]. There is a

(a) (b)

Fig. 4. (a) COACH handwashing assistant; (b) iSTRETCH haptic robotic device for stroke rehabilitation.

single sensor in the most recent prototype: a video camera. The video is fed to an input mapping that is predefined as a computer vision algorithm that tracks the two hands and the towel, and then outputs the locations of these objects discretized into a small set of regions. The POMDP includes a set of eight plan steps as *task* variables, a set of six simple behaviors, and models the user's responsiveness, awareness, and overall dementia level. The POMDP actions are to do nothing, to call for human assistance, or to give prompts for each plan step at one of three levels of specificity. Specific prompts are more costly than generic prompts, as they decrease feelings of independence in users [Mihailidis et al. 2001]. The output mapping converts the human assistance call to a real call for a caregiver to assist the user, and converts the actions to audio-visual prompts depending on the specificity level. Figure 4(a) shows the system working, with audio-visual prompting. The camera is out of sight on the ceiling above the sink.

The input mappings are given by a computer vision algorithm that tracks hands and towel. We use a mixed-state data-driven Bayesian sequential estimation method using *flocks* of color features [Hoey 2006], which allow objects to be robustly tracked over long periods of time, through large changes in shape and through partial occlusions. The tracker locations are discretized into a fixed set of regions around each element of interest in the sink area (e.g., soap, taps). The timeout function is based on the most likely belief, and is also dependent on the client's estimated level of dementia.

The handwashing task is modeled as a POMDP with 9 state variables, 3 observation variables, and 25 actions. There are 207360 states and 198 observations. The *task* variable is a set of plan steps that break the handwashing task down into eight steps (i.e., the different steps of handwashing that need to be completed, such as turning on the water, using the soap, etc,). The user *behaviors* cause transitions in the plan steps. The user behaviors can be one of six activities: using soap, at water, at tap, at sink, drying, or away. User *ability* is modeled using three factors: dementia level = [low,med,high], giving the user's overall functional ability at handwashing (low dementia level means more demented, lower ability); awareness = [never,no,yes], telling whether the user is aware of what she is doing in the task; and responsiveness = [none,max,med,min], giving what type of prompts the user is responsive to.

The POMDP model estimates a particular user's ability over time. In particular, the model can estimate a user's level of dementia by watching her long- term handwashing behavior over multiple sequences. When a new user starts using the system, the dementia-level variable has some prior distribution set based on the population of users. Over the course of each handwashing sequence, this distribution will shift slightly. This information is then propagated over future episodes of handwashing (this is the only variable whose information is propagated), then the system gets a long-term estimate of the user's dementia level. This information can be invaluable as a monitoring tool for caregivers, who can reassess users when their dementia level is estimated by the system to have changed.

The COACH can provide three levels of prompts to assist the user through the required activity steps. The system has 20 actions, 18 of which comprise prompts for six different plan tasks (water on/off, use soap, wet hands, rinse hands, dry hands) at three levels of specificity (general, moderate, specific). General prompts gently cue the user, while specific prompts are designed to get the user's attention and provide a detailed description of the task. The wording of the prompts was chosen based on work with real caregivers [Wilson et al. 2007], and was fixed for the duration of the experiments we describe here. The other two actions are the "null" action and "call caregiver". The latter action would normally terminate the process, but in our experiments, a caregiver is called to assist with the current step only (in order to preserve data from the study). The COACH does not provide any ability to customize, as it has been developed for a particular task and user group.

The reward function gives a large positive reward for the user completing the task (getting hands washed) and smaller positive rewards for intermediate steps. Prompting actions are costly, with more specific prompts being more costly, due to increased invasiveness leading to decreased independence feelings in the client. Calling for a human caregiver to assist is the most costly action.

The COACH system was tested in an eight–week trial with our target users: six persons with moderate to severe dementia in a long-term care facility in Toronto, Canada. The subjects washed their hands once a day, with assistance given by our automated system, or by a human caregiver, in alternating two-week periods. We measured the number of caregiver interventions that were necessary in each trial, and compared the four weeks with the system to the four weeks with only the human caregiver. Figure 5 shows plots of the number of interactions with a caregiver that were required over the eight week trial. We can see that, for subject 04, the system reduced the number of caregiver interventions to zero, indicating a perfect performance. For subject 05, on the other hand, the effect is less clear. Thus, we

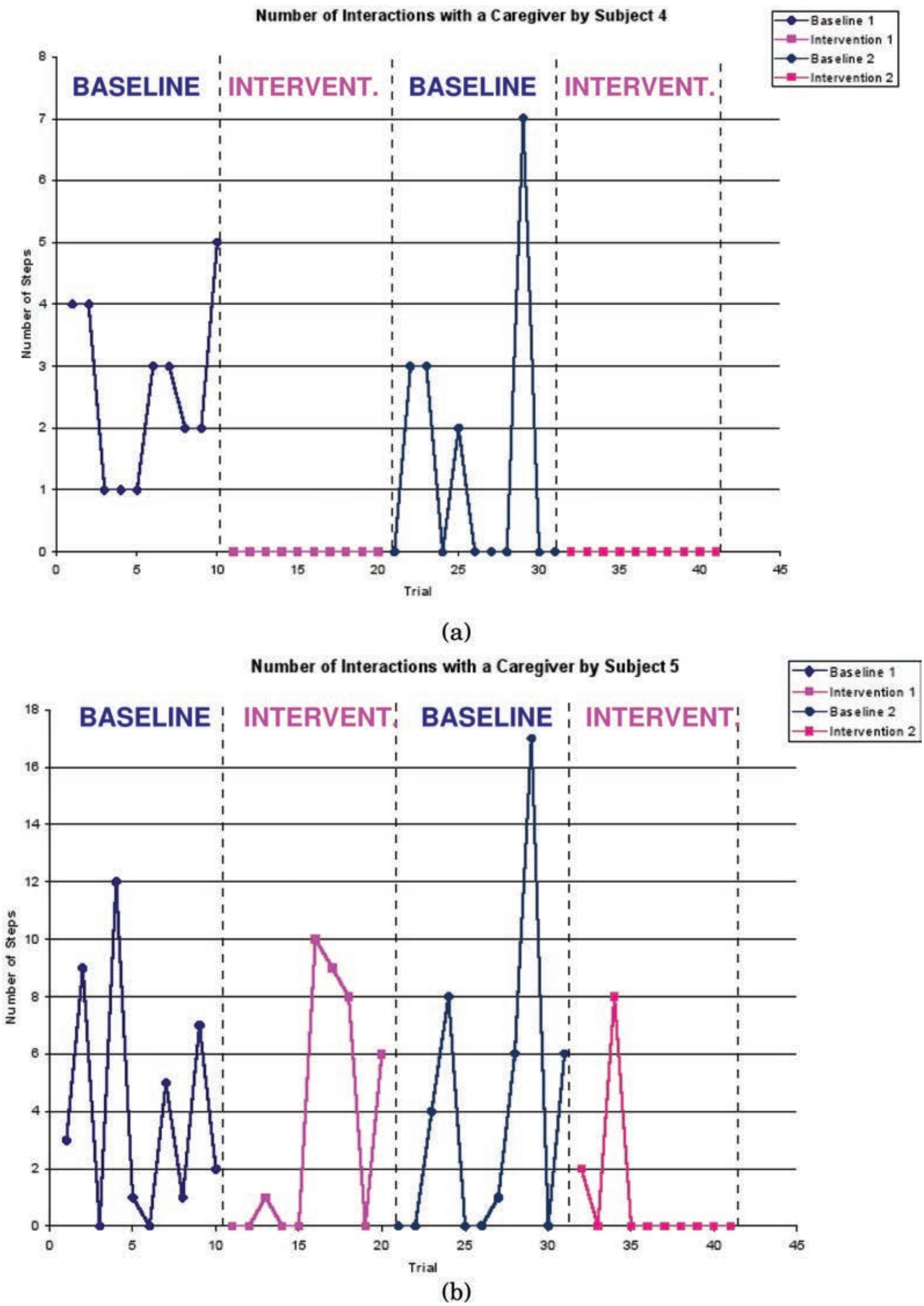

Fig. 5. Number of interactions required with a caregiver for two subjects over eight weeks (41 trials). The trials are grouped into 2-week segments, alternating *baseline* (without the system) and *intervention* (with the system). (a) subject 04; (b) subject 05.

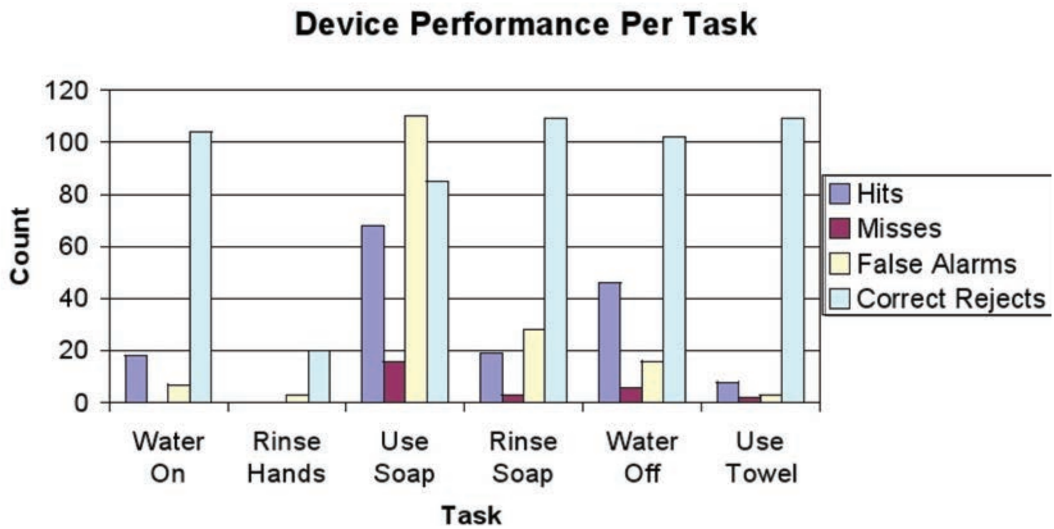

Fig. 6. Performance of the COACH system for handwashing on a per-step basis. *Hits* are the number of correct prompts given at the correct times for the given step, *misses* are the number of missed prompts, *correct rejects* are the number of times the system correctly left the subject to do the step on her own, and *false alarms* are the number of prompts issued when there should have been no prompt.

find the results are user dependent, and this can be due to a number of factors. For example, the method of delivery of the prompts (audio using a male voice) may be more evocative for some subjects than others. On average across all users, the reduction was about 25%.

We also looked in more detail at the ways in which the system failed, and Figure 6 shows the performance of the system for each type of prompt that was issued. We can see that the device has the most trouble with detecting and prompting for soap usage. This can be explained by the large differences in subject's behaviors for this step (e.g., some may use the soap very quickly, others more slowly, some with one hand, some with two, etc.). These differences make it challenging to build a single "use soap" detector, and this leads to the system more often misinterpreting this behavior.

We also found correlations between the system's estimate of the subject's dementia levels and her MiniMental State Exam (MMSE) scores, a well-known clinical measure of cognitive impairment [Hoey et al. 2010b]. A more complete set of our clinical findings (i.e., the effect of the system on users) is reported in Mihailidis et al. [2008], whilst our technical findings are reported in Hoey et al. [2010b].

Learning of behaviors has been investigated in the context of the COACH handwashing system. Given a set of data, we wish to learn automatically what behaviors are present and how these behaviors are related to the task. Behaviors are modeled as patterns of motion and image appearance over short time intervals, using a Dynamic Bayesian Network (DBN) similar to a Hidden Markov Model (HMM). An unsupervised method is used to cluster these behaviors into groups that are simultaneously recognizable (present in the data) and valuable for detecting states in the task and predicting rewards [Hoey and Little 2007]. The method has been applied to a large training set of handwashing data, and we have shown how relevant behaviors can be automatically extracted [Hoey et al. 2005]. We have also investigated reinforcement learning [Poupart et al. 2006; Grzes and Hoey 2011], and supervised learning of behaviors [Peters et al. 2009].

5.2. Stroke Rehabilitation: iSTRETCH

Every year stroke affects 16.3 million people worldwide. It is the second leading cause of death and the leading cause of permanent disability in adults [Truelsen and Bonita 2009]. The worldwide burden of stroke is increasing and expected to rise to 23 million first-ever strokes by 2030 [Strong et al. 2007]. Among those who have experienced a stroke, there are an estimated 64.5 million stroke survivors who live with varying levels of disability and need assistance for Activities of Daily Living (ADL) [Truelsen and Bonita 2009]. The burden of care for stroke survivors is high for the healthcare system and family members or caregivers [Truelsen and Bonita 2009; Stroke 2007]. Research has shown that stroke rehabilitation can reduce the impairments and disabilities that are caused by stroke, and improve motor function, allowing stroke patients to regain their independence and quality of life. It is generally agreed that intensive, repetitive, and goal-directed rehabilitation improves motor function and cortical reorganization in stroke patients with both acute and long-term (chronic) impairment [Fasoli et al. 2004]. However, this recovery process is slow and labor intensive, usually involving extensive interaction between one or more therapists and one patient. One of the main motivations for developing rehabilitation robotic devices is to automate interventions that are normally repetitive and physically demanding. These robots can provide stroke patients with intensive and reproducible movement training in time-unlimited durations, which can alleviate strain on therapists. In addition, these devices can provide therapists with accurate measures on patient performance and function (e.g., range of motion, speed, smoothness) over the course of a therapeutic intervention, and also provide quantitative diagnosis and assessments of motor impairments such as spasticity, tone, and strength [Hidler et al. 2005]. This technology makes it possible for a single therapist to supervise multiple patients simultaneously, which can contribute to the reduction of healthcare costs.

The system we describe in this section models the stroke rehabilitation task as an assistance task to autonomously facilitate upper-limb reaching rehabilitation for moderate-level stroke patients, to tailor the exercise parameters for each individual, and to estimate user fatigue. The system consists of a haptic robotic device coupled to a POMDP model that tracks a user's progress over time, and adjusts the level of difficulty based on the user's current abilities. More details on this system can be found in Kan et al. [2008, 2011], Lam et al. [2008], Huq et al. [2011], and Goetschalckx et al. [2011].

The robotic device, as detailed in Lam et al. [2008] and shown in Figure 4(b), was built by Quanser Inc., a Toronto-based robotics company. It features a nonrestraining platform for better usability and freedom of movement, and has two active and two passive degrees of freedom, which allow the reaching exercise to be performed in 3D space. The robotic device also incorporates haptic technology, which provides feedback through sense of touch. Encoders in the end-effector of the robotic device provide data to indicate hand position and shoulder abduction/internal rotation (i.e., compensation) during the exercise. Unobtrusive trunk sensors provide data to indicate trunk rotation compensation. The trunk sensors are comprised of three photoresistors taped to the back of a chair, each in one of three locations: the lower back, lower-left scapula, and lower-right scapula. The detection of light during the exercise indicates trunk rotation, as it means a gap is present between the chair and user. Finally, the virtual environment provides the user with visual feedback on hand position and target location during the exercise. The reaching exercise is represented in the form of a 2D bull's-eye game. The goal of the game is for the user to move the robot's end-effector, which corresponds to the cross-tracker in the virtual environment, to the bull's-eye target. The rectangular box is the virtual (haptic) boundary, which keeps the cross-tracker within those walls during the exercise.

The input mappings convert the time it takes the user to reach the target into three ranges: did not reach, slow, or normal; and whether the user demonstrates sufficient control and does not compensate. These settings are done by the physical therapist to allow for the abilities of a particular user. The *task* state is only related to the virtual game, and encodes things such as whether the user has completed a level. At the present time, we do not include any of these elements, but these could be added in the future. The *behaviors* model the time it takes the user to reach the target, the amount of control she exhibits while reaching (the amount of "wiggle" in the device as measured by the end-effector encoders), and whether they compensate or not (measured by the trunk photo-resistor sensors). The user's *abilities* are modeled as a product of three factors: her range at each resistance level, her level of fatigue, and her learning rate (some users rehabilitate faster than others). There are 10 possible actions the system can take. These are comprised of nine actions of which each is a different combination of setting a target distance $d \in \{d1, d2, d3\}$, and resistance level $r \in \{none, min, max\}$, and one action to stop the exercise when the user is fatigued.

The dynamics of all variables were specified manually using simple parametric functions of the user's fatigue and the difference between the system's setting of resistance and distance and the user's range (the amount by which the user is *stretched*). For example, if the user is not fatigued and the system sets a target at the user's range, then the user might have a 90% chance of reaching the target. However, if the target is set further, or if the user is fatigued, then this chance might decrease to 50%. The functions have simple parameters that can be specified by therapists, giving them customization of the POMDP model. The customization interface in this case gives the therapists access to only these simple parameters, and they set the amounts of *stretch* at which they expect a user to be able to reach the target, and how quickly a user will get fatigued. More details can be found in Kan et al. [2008, 2011].

The output can also be customized as mappings from levels of resistance and distance to actual resistance and distances on the haptic device. The idea is that, during weekly visits to a therapist, these mappings are reset based on the monitoring data from the previous week. The POMDP then starts from its initial state, and again guides the user through to the maximum resistance and distance levels, but the starting and ending points are physically different based on the output mappings.

The reward function was constructed to motivate the system to guide the user to exercise at maximum target distance and resistance level, while performing the task with maximum control and without compensation. Thus, the system was given a large reward for getting the user to reach the furthest target distance (d=d3) at maximum resistance (r=max). Smaller rewards were given when targets were set at or above the user's current range (i.e., when stretch >= 0), and when the user was performing well. However, no reward was given when the user was fatigued, failed to reach the target, had no control, or showed signs of compensation during the exercise.

The system has been tested in a pilot study with a single patient and one therapist. The patient was right-side hemiparetic, had a stroke onset of 227 days (7 months and 14 days) before enrollment, scored 4 on the arm section of the Chedoke-McMaster Stroke Assessment (CMSA) Scale [Gowland et al. 1993], was able to move to some degree but still had impaired movements as determined by her therapist, and could understand and respond to simple instructions. In each session, the therapist reviewed each POMDP decision and either agreed or disagreed with it (in which case the therapist made the decision). Each session lasted for approximately one hour and was completed three times a week for two weeks. The therapist agreed with both the target distance and resistance-level decisions made by the POMDP 94% and 90% of the time, respectively, but only 43% of the time for the stop decision. The POMDP wanted to stop the exercise to let the user take a break more often than the therapist wanted, but this could be changed by, for example, setting the cost for the stop action to be higher. The therapist was also asked to evaluate the POMDP decisions by rating the following statements on a Likert scale (1–4) with 4 being the most agreement: (a) the decisions made during the exercise were appropriate and (b) the patient was given an appropriate amount of time to complete each exercise before the next decision was made. The responses were 2.8 ± 0.4 for question (a) and 3.2 ± 0.4 for question (b). More details can be found in Kan et al. [2011]. We have also experimented with more complex, continuous models of how a person's abilities change with the task difficulty [Goetschalckx et al. 2011].

5.3. Art Therapy: ePAD

This section considers assisting persons with dementia and their therapists during art therapy sessions. There is increasing evidence that leisurely activities decrease dementia risk [Karp et al. 2006], and that cognitive activities can slow down the progression of Alzheimer's disease [Geda et al. 2009]. Engagement with visual artworks is also known to have benefits for the promotion of quality of life in older adults [Rusted et al. 2006]. However, many older adults have difficulty motivating themselves to engage in a creative activity for a reasonable period of time. These difficulties are compounded when the older adult has a progressive illness, such as Alzheimer's disease.

There has been some research published on the use of technology in art therapy contexts. For example, investigations on the use of video [Hartwich and Brandecker 1997] and digital photography [Wolf 2007] as new tools in psychotherapy have yielded very positive results. With respect to computer-based art therapy (i.e., using technology as part of the art therapy process), only a limited number of technologies have been developed. Examples include a multitouch surface that is used to simulate a sand-tray with objects in 3D above a 2D sand surface for interactive play therapy sessions [Hancock et al. 2010]. However, these applications are not practical for older adults with dementia because of required high levels of cognition and fine motor skills. Collie and Cubranic [1999] developed a computer-based art therapy program for use in tele-health [Collie and Cubranic 1999], but their system required an arts therapist to be physically present, and was not developed or tested with older adults with dementia. The majority of past computer-based applications have focused on art therapy assessment, such as for judging the main color in a drawing [Kim 2008], and estimating level of dementia based on the elements in a structured Mandala drawing [Kim et al. 2009]. Touchscreen devices for visual artworks have recently been developed [Raffle et al. 2007; Muise and Yim 2008], however, not for persons with dementia, and not as a tool for arts therapists specifically. To date there has been no work on computer-based systems that can prompt and monitor a user's participation in creative arts akin to the one we have developed. In addition, it should be noted that software tools that have been developed as "general" art technologies have not been included in this summary as they were not developed specifically as therapy tools.

Art therapy is a triadic relationship in which client, therapist, and artwork engage with each other. The primary goal of art therapy with older adults is to increase quality of life through promotion of autonomy [Harlan 1990], independence, and creative activities. Engaging in the art making process also provides an outlet for a person's emotional state. Through artwork, a person can communicate, for example, feelings of isolation and loneliness [Burns 2009].

The tool we have created is a creative arts touchscreen interface that presents a user with activities like painting, drawing, or collage (shown in Figure 7). The tool was developed with a user-centered design methodology in collaboration with a group of creative arts therapists. The tool is customizable by therapists, allowing them to design and build personalized therapeutic/goal-oriented creative activities for each client. The customized application attempts to maintain a client's *engagement*.

Fig. 7. ePAD art therapy tool.

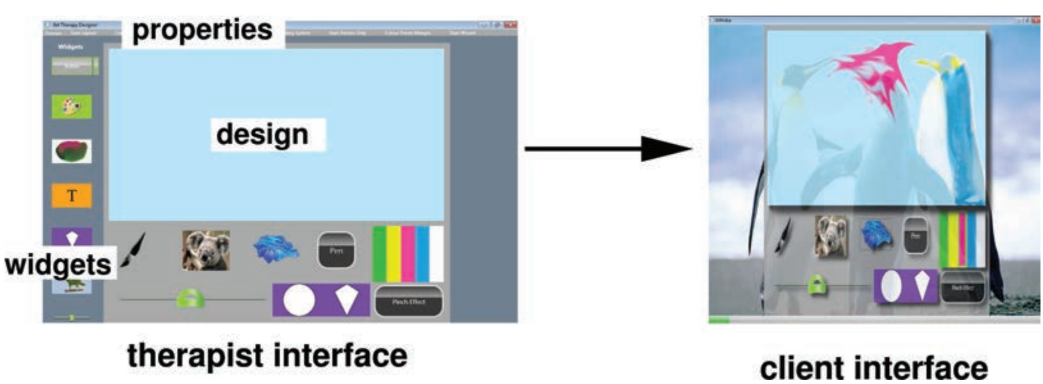

Fig. 8. Art therapist interface and client application.

The therapist uses the designer tool, which is a screen split into three sections: *widgets*, *design*, and *properties* (see Figure 8). The therapist can drag any widget onto the design area, configure it to her needs with a mouse (e.g., resize with a click-and-drag), and set properties and actions for the system. Once done, the therapist exports the newly created application for use by the client (Figure 8 on the right). When using the newly created application, a user does some visual art-work on a touchscreen. A therapist is present, and has a (possibly mobile) interface, but can leave the client for short periods of time (e.g., to interact with another client in a group session). The outputs of the device are a video stream from a Web-camera watching the user's face, and the interface actions (finger movements on the screen). These are passed to a therapist-defined set of input mappings, that map each behavior of a user on the screen to a category of *activity* $\in$ {*interactive,active,intermittent,inactive*}, defined as the amount of *engagement* a behavior requires. A second input mapping uses a computer vision algorithm to detect if a client is looking at the screen, a strong indicator of engagement. Images are taken from a standard Web-camera, and a Haar-like feature recognition method is used to detect a single face in the scene [Viola and Jones 2004].

There are two types of *behaviors* we expect to see in clients of this application.

(1) The *interface_behavior*$\in${*interactive,active,intermittent,inactive*} is whether the user is actively doing something on the interface, and is inferred from observations of her finger interactions (*activity* observation). The observation model $P(\mathbf{activity}|\mathbf{interface\_behavior})$ defines the measurement noise.
(2) The *gesture* variable is a set of gestures that indicate that a user is engaged with the device, and are inferred from the video stream. For example, the *gestures* could be gaze directions ({*looking,not looking*}), indicating if a person is looking at the screen. The *gestures* and the observation model $P(\mathbf{video}|\mathbf{gesture})$ are activity dependent, although for many applications the gaze direction will be sufficient.

The user ability model is given by two factors.

(1) The user's *engagement*$\in${*yes,confused,no*} is the key element of this model, as maintaining engagement is the primary purpose of the device. A user can be engaged (*yes*), or disengaged partially (*confused*) or completely (*no*). The dynamics of engagement, and the effects of the engagement on a user's behaviors, gestures, and task are user and activity dependent.
(2) The user's responsiveness to the system's actions *respond* is factored into a set of variables: $\{\mathbf{respond\_a_1}, \ldots, \mathbf{respond\_a_M}\}$, giving responsiveness to each of the system's action interactivity levels $a_1, \ldots, a_M$, respectively.

The actions the system can take are to do nothing (a_0), or to perform some intervention ($a_1, \ldots, a_M$), where intervention a_i is a level of *interactivity*, defined as the amount of involvement it requires from a user. The generic action is then returned to the application as an action for the system to take at that level of interactivity, using a set of therapist-defined output mappings. The actions range from adding shapes or images to the canvas, to animating buttons, or playing audio files. Action customizations are, for example, the sizes and locations of shapes that will be added by the system. The therapist can change the interactivity level $\in$ {*low, high, stimulate*} for each action. An example of a configuration would be to set the interactivity level of add circles to low, and play audio prompt to stimulate. The trade-off is that a very interactive prompt may get a disengaged user involved, but may be a disruptive action for an already engaged user, causing her to disengage.

The dynamics of the POMDP hinges on the user's *engagement*, which changes dynamically over time as a function of the system's actions, and her previous behaviors. For example, if the user is disengaged, but is looking at the screen, and the system does something of interest, then the user may become engaged with some probability. On the other hand, if a user is already engaged, and the system gives a prompt or changes the interface, the user may become confused. The user's responsiveness comes into play when the system takes an action. If she is responsive to the interactivity level of the action, the effect of the action is to increase her engagement.

The reward function is based solely on the user's engagement, with $+10$, -1, -2 if the user is engaged, confused, or not engaged, respectively. System actions are costly (-0.5), but only if the user is engaged. This models the effect of an action reducing feelings of independence in a user if she is already engaged. The POMDP controller (policy) thus computed takes actions that lead to behaviors defined as more *involved* by the therapist. Actually, the POMDP is optimizing over the user's *engagement*, but this is reflected in the amount of *involvement* of the behaviors. However, the POMDP must trade-off *involvement* and *engagement* against *interactivity*.

The system has recently been through a six-week trial with six art therapist-client dyads. The therapists built applications specifically for their clients, and then engaged with them in a series of six one-hour sessions. The feedback from therapists and clients alike was very positive, and we found many avenues for future work. Here we give a brief overview of the trial results with some salient points.

The goal of these trials is to find out if (A - Engagement) the device can be used by therapists to engage their clients in art; and (B - Automation) if the device could be used to automatically maintain user engagement for short periods of time during a session, thereby freeing up therapist time for other things. Throughout, the Likert scale is 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree.

A - Engagement. A primary question is whether the device can be used to engage clients in visual artwork. Therapists generally (5/6) felt that the device was an effective tool which they could use to engage their clients in a creative process, but some commented on the need for direction as to using the device. ATs were generally (5/6 for at least one session) fairly satisfied with the way the device could be used to support them in engaging clients and these ATs could envision using the device in an actual session. However, ATs (median of 4/5 on Likert for client engagement and 3/5 on Likert for their own satisfaction) remained uncertain as to whether they/ their clients were satisfied with the device, and felt that it might distract their other clients if they were to use this in a group art therapy session.

B - Automation and Prompting. Overall, the prompts were not very well explored by the therapists, probably because there were too many new things for each therapist/client dyad to deal with. This indicates that our study methodology could have used an additional step of usage of the device without system prompts. However, choosing and configuring system actions and additional settings was reported by ATs as being straightforward, but the rating system (level of interactivity) and the function of each prompt was described as being confusing and/or counter-intuitive by (2/6) ATs. The ATs also called for more flexibility in defining prompts.

Overall, we received a great deal of positive feedback from therapists, who were excited about the prospect of using the device. Clients were also very positive about the device. Most of them were able to use the device without difficulty, seemed to enjoy themselves, and, for the most part, seemed to be engaged in the task. The study also highlighted some of the key shortcomings of the current prototype. Our goal of creating a tool that would autonomously engage clients in art work (for short periods) has not yet been achieved. In hindsight, we believe the therapists were overwhelmed with the novelty of the device, and that more design work needs to go into the system action usage. While the customizability of ePAD was present functionally, our trials were not long enough to fully explore this aspect. The trials also gave us a "to-do" list of elements that can be added/removed from the current prototype. More details on this system can be found in Blunsden et al. [2009] and Mihailidis et al. [2010], and simulated examples of this system in use can be found in Hoey et al. [2010a]. Details on the six-week trial described before can be found in a forthcoming paper [Leuty et al. 2011].

5.4. Syndetic Assistance Processes: SNAP

Our final application area is focused on the customization of the POMDP prompting system. Our long-term goal is to design mechanisms for end-users (clients, caregivers, family members) to specify situated prompting systems for a specific need and task, and to be able to use the powerful decision making (AI) and computer vision and sensing tools that we are developing. Eventually, an end-user will, when confronted with a

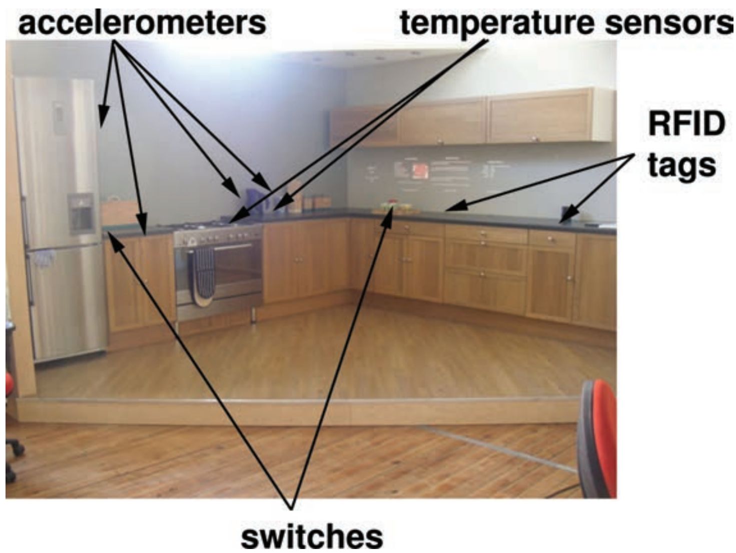

Fig. 9. Ambient kitchen, showing example sensors.

new problem needing a solution, be able to design and customize a situated prompting system with a POMDP back-end such as described previously.

The key to such a development is the formalization of a model of assistance (as before), and of a method for translating a real-world task and need into such a model. We begin by combining human factors research for task analysis [Wherton and Monk 2009], ubiquitous sensing of an environment [Olivier et al. 2009] (a kitchen), and the POMDP assistance model described earlier. In this section, we will give an overview of the task analysis and ubiquitous sensing, and then show how this can be translated into the POMDP assistance framework described before. We will use an example in this case of a person needing assistance in making a cup of tea. We are carrying out experiments in the ambient kitchen, a fully sensored and operational kitchen at the University of Newcastle, shown in Figure 9. More details can be found in Olivier et al. [2009].

The task analysis technique, as described in Wherton and Monk [2009], breaks a particular task down into a set of goals, states, abilities, and behaviors. The technique involves an experimenter video-taping a person being assisted during the task, and then transcribing and analyzing the video using a syndetic modeling technique. The end result is an Interaction Unit (IU) analysis that uncovers the states and goals of the task, the user's cognitive abilities, and the user's actions. A simplified example for the first step in tea making (getting out the cup and putting in a teabag) is shown in Table II. The rows in the table show a sequence of steps, with the user's current goals, the current state of the environment, the abilities that are necessary to complete the necessary step, and the behavior that is called for. The abilities are broken down into ability to *recall* what they are doing, to *recognize* necessary objects like the kettle, and to *perceive* affordances of the environment. This model of cognitive abilities is defined a priori by experts in the psychology of dementia, but generalizes across tasks, as mentioned in Section 4.1. While this model is specific to dementia, a range of similar models could also be built in a the same way (with different ability sets) for any task in which the user is cognitively overloaded.

The IU analysis shown in Table II can be converted to a POMDP model as follows. First, the *task* variables correspond to the entries in the state column in Table II. For

Table II. IU Analysis of the First Step in Tea Making

IU	Goals	Task States	Abilities	Behaviors
1	Final	cup empty on tray, box closed	Rn cup on tray, Rl step	No Action
2	Final, cup TB	cup empty on tray, box closed	Af cup on tray WS	Move cup tray→WS
3	Final, cup TB	cup empty on WS, box closed	Rn box, Rl box contains TB, Af box closed	Alter box to open
4	Final, cup TB	cup empty on WS, box open	Af TB in box cup	Move TB box→cup
5	Final	cup tb on WS, box open	Af box open	Alter box to closed
	Final	cup tb on WS, box closed		

Rn=recognition, Rl=Recall, Af=Affordance, tb=teabag, ws=work surface.

example, in the first step of tea making, these include the box condition (open, closed) and the cup contents (empty or with teabag). The user *behaviors* are described by one variable with values for each behavior in Table II. For the first IU group in tea making, these include opening/closing the box, moving the teabag to the cup, and doing nothing or something unrelated (these last two behaviors are always present). The user abilities are related to known Cognitive Abilities (CAs). These are the ability of the user to recall (Rl), recognize (Rn), and remember affordances (Af). For the first IU group, these include the ability to recognize the tea box ("Rn box" on line 3 of Table II) and the ability to perceive the affordance of moving the teabag to the cup ("Af TB in box cup" on line 4 of in Table II). In the associated POMDP, the abilities are related to the behaviors in the same way as in the framework given earlier; each behavior is dependent on a list of relevant abilities. That is, in order for the client to open the box, he/she must be able to recognize the box ("Rn Box"), recall that the box contains teabags ("Rl box contains TB", and see the affordance of opening the box ("Af box closed" as afforded by the box being closed).

The IU analysis shown in Table II is for a single goal the user can hold (in this case getting the teabag in the cup on the worksurface). For each goal, there is one overall ability to recall the step ("Rl Step" on line 1 of in Table II), and this ability serves only to push the goal for that user onto her list of known goals. We will pursue this further in Section 5.5 when we discuss hierarchical control mechanisms that can deal with multiple goals.

The system actions are the things the system can do to help the user. We define one system action for each necessary ability in the task. The actions correspond to a prompt or signal that will help the user with this particular ability, if missing. The system actions are specified by a ubiquitous sensing expert, based on known properties of the actuators in the environment. For example, to help with recognition of the cup, a light could be shone on it, or an audio prompt could be delivered. A range of prompt *specificities* can also be added for the same ability. A more specific prompt is usually considered more expensive (so will be used less often), but more effective [Mihailidis and Fernie 2002; Hoey et al. 2010b].

The observations are specified by a ubiquitous sensing expert, and are related to each state (task) variables or user behavior. For example, in the ambient kitchen, there are sensors in the countertops to detect if a cup is placed on them, and sensors in the teabags to detect if they are placed in the cup. The sensor noise is measured independently (as a miss/false positive rate for each state/sensor combination). The POMDP is able to fuse information from different sensors in a natural way by folding these sensor noise measures into the observation function. The POMDP can also handle missing sensor data, and sensors can be added or deleted easily from the model. In fact, the POMDP can also be easily modified so it must explicitly query sensors (rather than being forced to query all sensors at each time step). The POMDP can implement a decision-theoretic trade-off in this case, choosing the most in expensive sensor to use that will give it sufficient information to accomplish its goals [Koltunova et al. 2011].

The dynamics and initial state are produced directly from the IU analysis (Table II), by looking across each row and associating state transitions between rows. We take this to be deterministic, as any uncertainty will be introduced by the user's abilities (so we assume a perfectly able user is able to always successfully complete each step). The most important aspect of the dynamics is the effect of a system action on each ability. Each action has an effect on its associated cognitive ability, given by the probability that the cognitive ability associated with the action will change to "yes" if it is "no" if the system takes this action. For example, the "prompt recognition cup" action (e.g., a light shone on the cup) makes it more likely that the user can recognize the cup if she can't already. These probabilities would be specified either by an expert in dementia, or by a caregiver or therapist, or can be learned from data (see Section 3.2). Ideally, a therapist or caregiver would specify some distribution or range of probabilities, which could then be refined by the learning system.

The reward function specifies the final goal state (the final task states the last row in Table II), along with any intermediate goal states. In the tea making example, the system gets a big reward if the teabag is in the cup at the end and the box is closed, and smaller rewards if the teabag is in the cup but the box is open, and if the cup is empty by the box is open. These rewards are specified as a relative weighting for each subgoal. The actions are costly because we want to let a user do it herself if she can. These costs may be in physical values (dollars and cents) or in costs in terms of impact on a user. For example, an audio prompt for a step may have a low monetary cost, but a high cost in terms of making the user feel less independent. On the other hand, a laser beam indicating an object may have a high monetary cost, but a low user-impact cost. The precise specification of the reward function is very challenging, but a rough approximation can be elicited from prior knowledge or discussions with carers and psychologists (see Section 3.2).

We have developed software for the specification of situated prompting systems using the SNAP method, and have tested the method in the ambient kitchen using tea making and an actor [Hoey et al. 2011]. We have also generated POMDPs using SNAP for wayfinding, toothbrushing, handwashing, scheduling reminders (all for persons with dementia), and for a factory assembly task (for person's with Down's syndrome). The analyses for these last four tasks were performed by different biomedical engineers and research assistants, all with limited experience with POMDPs or planning in AI. As an example of the power of our method, the factory task was coded in about six hours by one biomedical engineer. The factory task contained 6 different IU analyses and associated POMDPs, each with about 5000 states, 24 observations, and 6 actions. This can be compared to a manual coding of the system for handwashing (a smaller task), that took over 6 months of work resulting in the system described in Boger et al. [2006].

In recent work, we have developed a relational database to encode the SNAP analysis for automated translation into a POMDP [Grzes et al. 2011]. The structure of this database forms a relational schemata of a Probabilistic Relational Model (PRM) [Getoor and Taskar 2007], and any particular instance is a relational skeleton that can be "grounded" using specific probabilities and rewards into a POMDP model for a specific task. This method allows for reuse of the schemata and skeleton in different tasks. Our future plans include testing on other tasks, advancing the specification towards end-users, learning parameters and reward functions, and doing lifted planning and inference.

5.5. Hierarchical Control

In order to act independently in more complex environments with multiple tasks, a person must at maintain a *goal stack* that provides her with motivation and direction for the task [Duke et al. 1998]. For example, in the kitchen, a goal stack might be

make breakfast→*make tea*→*get teabag*→*open teabox*.

Therefore, the person must have the ability to *recall* each of these goals in order, and may need to have some further abilities for recognizing objects, or for perceiving *affordances* of the environment [Ryu and Monk 2009]. The cognitive modeling approach of Wherton and Monk [2009], as explored in Section 5.4, uses a stack to model a person's goals. A person pushes a sequence of goals onto the stack, and then pops them off when they are achieved. Persons with dementia have difficulty pushing subgoals onto this stack, and often lose track of which subgoal they are completing (leading to a subgoal being popped from the stack prematurely). The pushing of a subgoal onto this stack is a user behavior, but a fundamentally different one from other physical acts (e.g., scooping some sugar), or psychological acts (such as recognizing the teacup), both of which only serve to immediately further the following behavior on the part of the user.

If we attempt to model an entire smart-home system as a single POMDP using the SNAP method described in the last section, we will soon run into computational problems. A simple solution to this problem is to distribute the model and approach each task independently. As in hierarchical approaches for robotics [Theocharous et al. 2004] and reinforcement learning [Sutton et al. 1999; Dietterich 2000], each subtask can be solved independently, and the solutions controlled sequentially. This approach does not work for a situated prompting system, however, for three reasons. (1) The client may not be acting rationally, and so may switch tasks suddenly, even to a dependent task that is not possible before the original task is complete, and may leave the original task in a critical state (e.g., the stove left on) [Geib 2002]; (2) the control in the assistance domain are suggestions to a client, and may not always be effective; (3) the system must be as *passive* as possible, letting the client accomplish tasks on her own if she can do so in order to maintain feelings of independence and control. These three considerations combine to make the subtasks nonindependent, and render approaches to assistance in which the system shares the action space with the client less effective (e.g., Fern et al. [2007]). In this case, it is not optimal for the system to "take over" and do things for the client if she forgets. Instead, the system must remind the person as passively as possible what to do.

For these reasons, and our goal of building truly distributed assistance systems that can be easily added or removed from a smart-home, we seek a distributed control mechanism that will unify many POMDP-based controllers for assistance. Pineau et al. [2001] propose an approach that handles problems with a decomposable action and state space. The method we use is similar, but our problem structure does not allow for easy decomposition of the state space, since high-level states (goal recall abilities for high-level goals) have effects that are felt throughout the hierarchy. We impose a hierarchy on the controller, but allow for sharing of some state and value information between levels, resulting in an approximate solution. With sufficient computational power, we can relax this imposed hierarchy, allowing other methods (such as Pineau et al. [2001]) to be used directly. Distributed control in our case can also be formulated as a resource allocation problem [Hoey and Grzes 2011]. Weakly coupled resource allocation tasks (in which subtasks are independent given resources) can be successfully tackled with an approximate, greedy algorithm for solving MDPs [Meuleau et al. 1998; Dolgov and Durfee 2006; Guestrin and Gordon 2002]. Distributed control can also be modeled as a Dec-POMDP [Seuken and Zilberstein 2007; Boutilier 1999], which poses computational challenges beyond what we are capable of handling at this time. Specific classes of DEC-POMDPs have been developed [Varakantham et al. 2009], but rely on having "highly" autonomous agents that interact only in certain locales. Our method, on the other hand, has very dependent agents (they share resources), but imposes a structure that can be exploited for policy construction using a greedy approach. Further, we are explicitly leveraging detailed a priori domain knowledge to construct specific solution methods that work for the large domains we are working on.

Therefore, we have designed a higher-level POMDP controller whose job is only to deal with a particular level of user subgoals [Hoey and Grzes 2011]. Once a user has pushed a particular subgoal onto her stack, this high-level controller passes off control to a lower-level subgoal controller that guides the user through the individual steps related to that subgoal. If the person loses track of what she is doing in that subgoal, the lower-level controller will attempt to prompt her to get her back on track. If she does not respond to such prompts, the higher-level controller will start to believe that the person has lost the subgoal from her stack (the belief in her ability to recall the subgoal will decrease), and will step in and issue an appropriate reminder of that subgoal, vetoing any actions recommended by the lower-level controllers.

The controllers can be nested, encoding tasks that require assistance into a tree-shaped hierarchy, such that each path from the root to a leaf describes a particular set of abilities required to complete the task at the leaf. Each node in the tree consists of a stand-alone controller, encoded as a POMDP, that prompts the client for a subset of abilities, but relies on its ancestors to prompt for any abilities higher in the tree. The children of a node can be seen as *macro*behaviors that require the abilities controlled by the parent. An initial, offline phase computes policies for each node in the tree. Subsequently, during interactions with a client online, the control algorithm processes each new observation in a two-pass distributed fashion. A bottom-up traversal of the tree allows each node to compute its expected value given each possible constraint placed by its parent and each value reported by its children, and to report this expected value to the parent. Subsequently, a top-down traversal reveals the best action to take at each level.

Such a hierarchical controller is an instance of the assistance POMDP described in Section 4.2, which we refer to as a *composite controller*. The systems we have described previously are *atomic controllers* in the sense that they receive observations directly from the environment, perform actions directly in the environment (e.g., audio prompts), and are self-contained prompting systems for individual subtasks. The composite controllers, on the other hand, receive observations from other controllers (either atomic or composite), and rely on the subgoals for the majority of actions. Each composite controller has a set of N subcontrollers, denoted $C_1, C_2, \ldots, C_N$, and has the following structure.

—The *observations* for the composite controller include K, whether each subgoal is completed or not according to its belief state, and V, the index of the subgoal in which user activity or an exogenous event (indicated by a sensor change without a user action) has been observed, or 0 if no activity has been observed and there was a timeout.
—The *behavior* variable, B, gives the subgoal the user is currently attempting, as reported by the subcontrollers. The behavior can be *none* meaning that a timeout was observed (no specific user actions were taken).
—The *ability* variables are one for each *recall* ability of each subgoal, and condition what a user is expected to do next.
—The *task* variables include variables denoting whether that subgoal has been completed yet. Subgoals complete at some finite rate if the user behavior corresponds

correctly to the current control subgoal and all prerequisites are fulfilled. Additionally, a *control* variable gives the current subgoal that should be being attempted by the user. Note that this may be different from the behavior (if the user is doing the wrong thing).

—The *reward* function for this controller corresponds to the ordering of goals to complete the task.

—*Subgoal dependencies* (on other subgoals) are encoded in the composite controller as behavior relevance functions: they give the situations in which each behavior is relevant, and will be zero for behaviors (subgoals) that depend on previous subgoals according to the dependencies

A composite controller runs in parallel (event-driven) time with the current sub-goal in control, and makes two contributions.

(1) It has a veto power over the current control subgoal: if it decides to take an action, the subcontroller is forced to “do nothing”. These actions will be taken in situations when the composite controller does not believe the subcontroller will be able to complete.
(2) It decides on the subgoal in control by consulting its own *control* variable and choosing the most likely control subgoal.

These two contributions allow the composite controller to have some degree of control over the achievement of value. It can decide to take action to steer a user towards a control goal that she is not currently attempting. It computes a value for each action (including its own prompting actions and all allocations to its children) based on its own Q-function, and a weighted sum of the Q-functions of its children given the allocation and its estimates of whether the children will complete if allocated control. More details of this control mechanism can be found in Hoey and Grzes [2011].

6. CONCLUSIONS

This article has described the current state of our efforts over the past decade towards developing general-purpose assistance technologies for persons with cognitive and physical disabilities. The systems we have built are based on a general-purpose framework for the specification, customization, and use of decision-theoretic prompting systems. The method is based on the partially observable Markov decision process, and combines elements allowing for user customization, system adaptivity to users, and general-purpose sensing abilities. This article has given a detailed presentation of the method, and five case studies of applications of the method to situated prompting systems.

Our future work is working in three directions. First, we seek to bring our current technologies to market by increasing robustness and quality guarantees. For example, we are currently developing a version of the handwashing system that will be inexpensive, easy to install, and can be evaluated with a large number of users in their own homes. Second, we are developing novel algorithms and systems that will take our current prototypes to the next level, allowing for more complete end-user customization, adaptation to users through learning, and generalizability through novel computer vision methods. A limitation of our current work is the relatively small population sizes that we have tested our systems with. We have seen significant variability in the responses of clients to our systems, and we are currently researching methods for increasing the ability of end-users to customize our systems to suit their invididual needs. A further area of investigation is in system to detect and repair problem areas within a task (e.g., rinsing soap in the COACH). Finally, through a user-centered design process, we are seeking to develop and explore new areas for deployment of our methods and systems. The usage of multitouch display tables for physical and cognitive rehabilitation is a key area in this regard. We are also exploring mobile applications for persons with dementia, specifically to assist in wayfinding for persons who wander [Hoey et al. 2012].

Overall, we see the major impacts of this work being brought to fruition by building more formal methods for user and developer inclusion in the design and development process of assistive technologies for smart-homes. We are working towards the "do-it-yourself" smart-home solution, where end-users can easily custom-build solutions by downloading and installing simple plug-and-play components. These solutions will require a solid theoretical grounding to assure robustness and repeatability, and to ensure that the "user" gets as much attention as the "system".

ACKNOWLEDGMENTS

We thank the many people who have been involved over the years: Tom Bartindale, Axel von Bertoldi, Justin Bimbrahw, Scott Blundsen, Jen Boger, Marcelle Candido, Brent Carmichael, Tammy Craig, Steve Czarnuch, Pantelis Elinas, Kate Fenton, Dave Geisbrecht, Robby Goetschalckx, Marek Grześ, Rajibul Huq, Dan Jackson, Patricia Kan, Ryan Keyfitz, Shehroz Khan, Kelvin Lam, Paul Lam, Valerie Leuty, JP Lobos, Christian Peters, Cuong Pham, Thomas Plötz, Dmitry Pyryeskin, Jasper Snoek, Babak Taati, Rosalie Wang, Szymon Wartak, Rozanne Wilson, and Kris Zutis. Finally, we thank all the participants who have given so generously of their time to helping evaluate our research.

REFERENCES

ÅSTRÖM, K. J. 1965. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Appl. 10*, 174–205.

ABBAS, A. 2004. Entropy methods for adaptive utility elicitation. *IEEE Trans. Syst. Sci. Cybernetics 34,* 2, 169–178.

ABOWD, G., BOBICK, A., ESSA, I., MYNATT, E., AND ROGERS, W. 2002. The aware home: Developing technologies for successful aging. In *Proceedings of the AAAI Workshop and Automation as a Care Giver*. Association for the Advancement of Artificial Intelligence (AAAI) Press.

ALZHEIMER'S DISEASE INTERNATIONAL. 2009. World alzheimer's report. http://www.alz.co.uk/research/worldreport/.

ALZHEIMER'S SOCIETY OF CANADA. 2010. Rising tide: The impact of dementia on canadian society. http://www.alzheimer.ca/english/rising_tide/rising_tide_report.htm.

BEAL, M. J., GHAHRAMANI, Z., AND RASMUSSEN, C. E. 2002. Infinite hidden Markov model. In *Advances in Neural Information Processing Systems*.

BENGIO, Y. AND FRASCONI, P. 1996. Input/output HMMs for sequence processing. *IEEE Trans. Neural Netw. 7,* 5, 1231–1249.

BHARUCHA, A., ATKESON, C., AND CHEN, D. 2006. Caremedia: Automated video and sensor analysis for geriatric care. *J. Amer. Med. Directors Assoc. 7,* B9.

BHARUCHA, A. J., ANAND, V., FORLIZZI, J., DEW, M. A., REYNOLDS, C. F., STEVENS, S., AND WACTLAR, H. 2009. Intelligent assistive technology applications to dementia care: Current capabilities, limitations, and future challenges. *Amer. J. Geriatr. Psychiat. 17,* 2, 88–104.

BLUNSDEN, S., RICHARDS, B., BARTINDALE, T., JACKSON, D., OLIVIER, P., BOGER, J. N., MIHAILIDIS, A., AND HOEY, J. 2009. Design and prototype of a device to engage cognitively disabled older adults in visual artwork. In *Proceedings of the ACM 2nd International Conference on PErvasive Technologies Related to Assistive Environments*.

BOGER, J., HOEY, J., POUPART, P., BOUTILIER, C., FERNIE, G., AND MIHAILIDIS, A. 2006. A planning system based on Markov decision processes to guide people with dementia through activities of daily living. *IEEE Trans. Inf. Technol. Biomed. 10,* 2, 323–333.

BOGER, J., POUPART, P., HOEY, J., BOUTILIER, C., FERNIE, G., AND MIHAILIDIS, A. 2005. A decision-theoretic approach to task assistance for persons with dementia. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1293–1299.

BOUTILIER, C. 1999. Sequential optimality and coordination in multiagent systems. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI)*. 478–485.

BOUTILIER, C. 2002. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02)*. 239–246.

BOUTILIER, C., PATRASCU, R., POUPART, P., AND SCHUURMANS, D. 2006. Constraint-Based optimization and utility elicitation using the minimax decision criterion. *Artif. Intell. 170,* 8–9, 686–713.

BRAZIUNAS, D. AND BOUTILIER, C. 2010. Assessing regret-based preference elicitation with the UTPREF recommendation system. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC'10)*. 219–228.

BRDICZKA, O., CROWLEY, J. L., AND REIGNIER, P. 2009. Learning situation models in a smart home. *IEEE Trans. Syst. Man Cybernet. B: Cybernet. 39,* 1.

BURNS, J. 2009. An interpretive description of the patterns and practice of arts therapists working with older people who have dementia in the uk. Ph.D. thesis, Queen Margaret University, Edinburgh, Scotland.

CHAJEWSKA, U., KOLLER, D., AND PARR, R. 2000. Making rational decisions using adaptive utility elicitation. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*. 363–369.

CHEN, D., YANG, J., AND WACTLAR, H. 2005. A study of detecting social interaction with sensors in a nursing home environment. In *Computer Vision in Human-Computer Interaction (ICCV Workshop on HCI)*, N. Sebe, M. Lew, and T. Huang, Eds. Lecture Notes in Computer Science, vol. 3766. Springer, 199–210.

CHEN, L., NUGENT, C. D., BISWAS, J., AND HOEY, J., EDS. 2011. *Activity Recognition in Pervasive Intelligent Environments*. Atlantis/Springer.

COLLIE, K. AND CUBRANIC, D. 1999. An art therapy solution to a telehealth problem. *Art Therapy: The J. Amer. Art Therapy Assoc. 16,* 4, 186–193.

DARRELL, T. AND PENTLAND, A. P. 1996. Active gesture recognition using partially observable Markov decision processes. In *Proceedings of the 13th Iternational Conference on Pattern Recognition (ICPR)*.

DEMPSTER, A., LAIRD, L., AND RUBIN, D. 1972. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B 39,* 1, 1498–1527.

DIETTERICH, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res. 13*, 227–303.

DOLGOV, D. A. AND DURFEE, E. H. 2006. Resource allocation among agents with mdp-induced preferences. *J. Artif. Intell. Res. 27*, 505–549.

DOSHI-VELEZ, F., WINGATE, D., TENENBAUM, J., AND ROY, N. 2011. Infinite dynamic Bayesian network. In *Proceedings of the International Conference on Machine Learning*.

DUKE, D., BARNARD, P., DUCE, D., AND MAY, J. 1998. Syndetic modelling. *Hum. Comput. Interact. 13,* 4, 337.

FASOLI, S., KREBS, H., AND HOGAN, N. 2004. Robotic technology and stroke rehabilitation: Translating research into practice. *Topics Stroke Rehab. 11,* 4, 11–19.

FERN, A., NATARAJAN, S., JUDAH, K., AND TADEPALLI, P. 2007. A decision-theoretic model of assistance. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI)*.

GEDA, Y. E., ROBERTS, R., KNOPMAN, D., PANKRATZ, V. S., CHRISTIANSON, T., BOEVE, B., IVNIK, R., TANGALOS, E. G., ROCCA, W., AND PETERSEN, R. 2009. Cognitive activities are associated with decreased risk of mild cognitive impairment: The mayo clinic population-based study of aging. In *Abstracts of the 61st Annual Meeting of the American Academy of Neurology*.

GEIB, C. W. 2002. Problems with intent recognition for elder care. In *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Automation as Caregiver*.

GETOOR, L. AND TASKAR, B., EDS. 2007. *Statistical Relational Learning*. MIT Press.

GOETSCHALCKX, R., POUPART, P., AND HOEY, J. 2011. Continuous correlated beta processes. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*.

GOWLAND, C., STRATFORD, P., WARD, M., MORELAND, J., TORRESIN, W., AND HULLENAAR, S. V. 1993. Measuring physical impairment and disability with the chedoke-mcmaster stroke assessment. *Stroke 24,* 1, 58–63.

GRZES, M. AND HOEY, J. 2011. Efficient planning in rmax. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

GRZES, M., HOEY, J., KHAN, S., MIHAILIDIS, A., CZARNUCH, S., JACKSON, D., AND MONK, A. 2011. Relational approach to knowledge engineering for pomdp-based assistance systems with encoding of a psychological model. In *Proceedings of the ICAPS Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*.

GUESTRIN, C. AND GORDON, G. 2002. Distributed planning in hierarchical factored mdps. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*. 197–206.

HAMID, R., HUANG, Y., AND ESSA, I. 2003. Argmode - Activity recognition using graphical models. In *Proceedings International Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Detection and Recognition of Events in Video*.

HANCOCK, M., CATE, T. T., CARPENDALE, S., AND ISENBERG, T. 2010. Supporting sandtray therapy on an interactive tabletop. In *Proceedings ACM SIGCHI Conference on Human Factors in Computing Systems*.

HARLAN, J. 1990. Beyond the patient to the person aspects of autonomous functioning in individuals with mild to moderate dementia. *Amer. J. Art Therapy 28,* 4, 99–106.

HARTWICH, P. AND BRANDECKER, R. 1997. Computer-Based art therapy with inpatients: Acute and chronic schizophrencis and boderline cases. *Arts Psychother. 24,* 4, 367–373.

HIDLER, J., NICHOLS, D., PELLICCIO, M., AND BRADY, K. 2005. Advances in the understanding and treatment of stroke impairment using robotic devices. *Topics Stroke Rehab. 12,* 2, 22–35.

HOEY, J. 2006. Tracking using flocks of features, with application to assisted handwashing. In *Proceedings of the British Machine Vision Conference*. M. J. Chantler, E. Trucco, and R. B. Fisher, Eds., vol. 1. 367–376.

HOEY, J. AND GRZES, M. 2011. Distributed control of situated assistance in large domains with many tasks. In *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS)*.

HOEY, J., LEUTY, V., ZUTIS, K., AND MIHAILIDIS, A. 2010a. A tool to promote prolonged engagement in art therapy: Design and development from arts therapist requirements. In *Proceedings of 12th International ACM SIGACCESS Conference on Computers and Accessibility*.

HOEY, J. AND LITTLE, J. J. 2007. Value-Directed human behavior analysis from video using partially observable Markov decision processes. *IEEE Trans. Pattern Anal. Mach. Intell. 29,* 7, 1118–1132.

HOEY, J., PLÖTZ, T., JACKSON, D., MONK, A., PHAM, C., AND OLIVIER, P. 2011. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervas. Mobile Computing*. (Special Issue on Knowledge-Driven Activity Recognition in Intelligent Environments) *7,* 3, 299–318.

HOEY, J. AND POUPART, P. 2005. Solving POMDPs with continuous or large discrete observation spaces. In *Proceedings International Joint Conference on Artificial Intelligence*. 1332–1338.

HOEY, J., POUPART, P., BOUTILIER, C., AND MIHAILIDIS, A. 2005. Semi-Supervised learning of a POMDP model of patient-caregiver interactions. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Modeling Others from Observations*. 101–110.

HOEY, J., POUPART, P., VON BERTOLDI, A., CRAIG, T., BOUTILIER, C., AND MIHAILIDIS, A. 2010b. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Comput. Vis. Image Understand. 114,* 5, 503–519.

HOEY, J., YANG, X., QUINTANA, E., AND FAVELA, J. 2012. LaCasa:Location and context aware safety assistant. In *Proceedings of 6th International Conference on Pervasive Computing Technologies for Healthcare*.

HUQ, R., KAN, P., GOETSCHALCKX, R., HEBERT, D., HOEY, J., AND MIHAILIDIS, A. 2011. A decision-theoretic approach in the design of an adaptive upper-limb stroke rehabilitation robot. In *Proceedings of International Conference on Rehabilitation Robotics*.

INTILLE, S. S. 2006. The goal: Smart people, not smart homes. In *Proceedings of the International Conference on Smart Homes and Health Telematics*. IOS Press.

KAELBLING, L. P., LITTMAN, M. L., AND CASSANDRA, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell. 101*, 99–134.

KAN, P., HOEY, J., GOETSCHALCKX, R., AND MIHAILIDIS, A. 2011. The development of an adaptive upper-limb stroke rehabilitation robotic system. *J. NeuroEngin. Rehab. 8,* 33.

KAN, P., HOEY, J., AND MIHAILIDIS, A. 2008. Stroke rehabilitation using haptics and a POMDP controller. In *Proceedings of the AAAI Fall Symposium on Caring Machines: AI in Eldercare*.

KARP, A., PAILLARD-BORG, S., WANG, H.-X., SILVERSTEIN, M., WINBLAD, B., AND FRATIGLIONI, L. 2006. Mental, physical and social components in leisure activities equally contribute to decrease dementia risk. *Dement. Geriatr. Cogn. Disord. 21,* 2, 65–73.

KAUTZ, H., ARNSTEIN, L., BORRIELLO, G., ETZIONI, O., , AND FOX, D. 2002. An overview of the assisted cognition project. In *Proceedings Association of the AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*. 60–65.

KEENEY, R. L. AND RAIFFA, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Wiley, New York.

KIM, S. 2008. Computer judgment of main color in a drawing for art psychotherapy assessment. *Arts Psychother. 35*, 140–150.

KIM, S., BETTS, D., KIM, H., AND KANG, H. 2009. Statistical models to estimate level of psychological disorder based on a computer rating system: An application to dementia using structured mandala drawings. *Arts Psychother. 36,* 214–221.

KOLTUNOVA, V., HOEY, J., AND GRZEŚ, M. 2011. Goal-Oriented sensor selection for intelligent phones: (Gossip). In *Proceedings of the International Workshop on Situation Activity and Goal Awareness*. (*SAGAware '11*). ACM, New York, 83–88.

LAM, P., HÉBERT, D., BOGER, J., LACHERAY, H., GARDNER, D., APKARIAN, J., AND MIHAILIDIS, A. 2008. A haptic-robotic platform for upper-limb reaching stroke therapy: Preliminary design and evaluation results. *J. NeuroEngin. Rehab. 5,* 15, 1–13.

LESTER, J., CHOUDHURY, T., KERN, N., BORRIELLO, G., AND HANNAFORD, B. 2005. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the International Joint Conferences on Artificial Intelligence*. 766–772.

LEUTY, V., BOGER, J., YOUNG, L., HOEY, J., AND MIHAILIDIS, A. 2011. Engaging older adults with dementia in creative occupations using artificially intelligent assistive technology. *Assist.Technol.* DOI 10.1080/10400435.2012.715113.

LIAO, L., PATTERSON, D., FOX, D., AND KAUTZ, H. 2007. Learning and inferring transporation routines. *Artif. Intell. 171,* 5-6, 311–331.

LOPRESTI, E. F., MIHAILIDIS, A., AND KIRSCH, N. 2004. Assistive technology for cognitive rehabilitation: State of the art. *Neuropsychol. Rehab. 14,* 1/2, 5–39.

MEULEAU, N., HAUSKRECHT, M., KIM, K.-E., PESHKIN, L., KAELBLING, L. P., DEAN, T., AND BOUTILIER, C. 1998. Solving very large weakly coupled Markov decision processes. In *Proceedings Association for the Advancement of Artificial Intelligence (AAAI)*. 165–172.

MIHAILIDIS, A., BLUNSDEN, S., BOGER, J., RICHARDS, B., ZUTIS, K., YOUNG, L., AND HOEY, J. 2010. Towards the development of a technology for art therapy and dementia: Definition of needs and design constraints. *Arts Psychother. 37,* 4.

MIHAILIDIS, A., BOGER, J., CANDIDO, M., AND HOEY, J. 2008. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC Geriatr. 8,* 28.

MIHAILIDIS, A., BOGER, J. N., HOEY, J., AND JIANCARO, T. 2011. *Zero Effort Technologies: Considerations, Challenges, and Use in Health, Wellness, and Rehabilitation.* Morgan Claypool.

MIHAILIDIS, A., ELINAS, P., BOGER, J., AND HOEY, J. 2007. An intelligent powered wheelchair to enable mobility of cognitively impaired older adults: an anticollision system. *IEEE Trans. Neural Syst. Rehab. Engin. 15,* 1, 136–143.

MIHAILIDIS, A. AND FERNIE, G. R. 2002. Context-Aware assistive devices for older adults with dementia. *Gerontechnol. 2*, 173–189.

MIHAILIDIS, A., FERNIE, G. R., AND BARNEBEL, J. C. 2001. The use of artificial intelligence in the design of an intelligent cognitive orthosis for people with dementia. *Assist. Technol. 13*, 23–39.

MODAYIL, J., BAI, T., AND KAUTZ, H. 2008a. Improving the recognition of interleaved activities. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp08)*.

MODAYIL, J., LEVINSON, R., HARMAN, C., HALPER, D., AND KAUTZ, H. 2008b. Integrating sensing and cueing for more effective activity reminders. In *Proceedings of the AAAI Fall Symposium on AI in Eldercare: New Solutions to Old Problems*.

MOESLUND, T. B., HILTON, A., AND KRÜGER, V. 2006. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Understand. 2*-3, 90–126.

MONTEMERLO, M., PINEAU, J., ROY, N., THRUN, S., AND VERMA, V. 2002. Experiences with a mobile robotic guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI).

MOZER, M. C. 2005. Lessons from an adaptive house. In *Smart Environments: Technologies, Protocols, and Applications*, D. Cook and R. Das, Eds., J. Wiley and Sons, Hoboken, NJ, 273–294.

MUISE, K. AND YIM, J. 2008. A touching harmony: Midas in artistic practice. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*. ACM, 213–214.

NG, A. AND RUSSELL, S. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*. 663–670.

NGUYEN, N. T., BUI, H. H., VENKATESH, S., AND WEST, G. 2003. Recognising and monitoring high-level behaviors in complex spatial environments. In *Proceedings International Conference on Computer Vision and Pattern Recognition (CVPR)*.

OLIVIER, P., MONK, A., XU, G., AND HOEY, J. 2009. Ambient kitchen: Designing situated services using a high fidelity prototyping environment. In *Proceedings of the ACM 2nd International Conference on PErvasive Technologies Related to Assistive Environments*.

PATTERSON, D., ETZIONI, O., AND KAUTZ, H. 2002. The activity compass. In *Proceedings of the Ist International Workshop on Ubiquitous Computing for Cognitive Aids*.

PENTNEY, W., PHILIPOSE, M., AND BILMES, J. 2008. Structure learning on large scale common sense statistical models of human state. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*.

PENTNEY, W., PHILIPOSE, M., BILMES, J. A., AND KAUTZ, H. A. 2007. Learning large scale common sense models of everyday life. In *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*. 465–470.

PETERS, C., WACHSMUTH, S., AND HOEY, J. 2009. Learning to recognise behaviors of persons with dementia using multiple cues in an hmm-based approach. In *Proceedings of the ACM 2nd International Conference on PErvasive Technologies Related to Assistive Environments*.

PHILIPOSE, M., FISHKIN, K., PATTERSON, D., FOX, D., KAUTZ, H., AND HAHNEL, D. 2004. Inferring activities from interactions with objects. *Pervas. Comput. 10–17*.

PINEAU, J., MONTEMERLO, M., POLLACK, M., ROY, N., AND THRUN, S. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics Auton. Syst. 42,* 3–4.

PINEAU, J., ROY, N., AND THRUN, S. 2001. A hierarchical approach to pomdp planning and execution. In *Proceedings of the ICML Workshop on Hierarchy and Memory in Reinforcement Learning*.

POLLACK, M. E. 2005. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Mag. 26,* 2, 9–24.

POLLACK, M. E. 2006. Autominder: A case study of assistive technology for elders with cognitive impairment. *Generat. J. Amer. Soc. Aging 30,* 2, 67–69.

POUPART, P. 2005. Exploiting structure to efficiently solve large scale partially observable markov decision processes. Ph.D. thesis, University of Toronto, Toronto, Canada.

POUPART, P. 2012. An introduction to fully and partially observable Markov decision processes. In *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, E. Sucar, E. Morales, and J. Hoey, Eds., IGI Global, 33–62.

POUPART, P. AND BOUTILIER, C. 2004. VDCBPI: An approximate scalable algorithm for large scale POMDPs. In *Proceedings of the Conference on Advances in Neural Information Processing Systems 17 (NIPS)*. 1081–1088.

POUPART, P., VLASSIS, N., HOEY, J., AND REGAN, K. 2006. An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*.

RAFFLE, H., VAUCELLE, C., WANG, R., AND ISHII, H. 2007. Jabberstamp: Embedding sound and voice in traditional drawings. In *Proceedings of the Conference on Interaction Design and Children (IDC)*. ACM Press.

REGAN, K. AND BOUTILIER, C. 2009. Regret-Based reward elicitation for Markov decision processes. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*. 454–451.

REGAN, K. AND BOUTILIER, C. 2011a. Eliciting additive reward functions for Markov decision processes. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*. 2159–2164.

REGAN, K. AND BOUTILIER, C. 2011b. Robust online optimization of reward-uncertain MDPs. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*. 2165–2171.

ROY, N., GORDON, G., AND THRUN, S. 2003. Planning under uncertainty for reliable health care robotics. In *Proceedings of the International Conference on Field and Service Robotics*.

RUSTED, J., SHEPPARD, L., AND WALLER, D. 2006. A multi-centre randomized control group trial on the use of art therapy for older people with dementia. *Group Anal. 39,* 4, 517–536.

RYBSKI, P. E. AND VELOSO, M. M. 2004. Using sparse visual data to model human activities in meetings. In *Proceedings IJCAI Workshop on Modeling Other Agents from Observations (MOO 04)*.

RYU, H. AND MONK, A. F. 2009. Interaction unit analysis: A new interaction design framework. *Hum.-Comput. Interact. 24,* 4, 367–407.

SALO, A. AND HÄMÄLÄINEN, R. P. 2001. Preference ratios in multiattribute evaluation (PRIME)–Elicitation and decision procedures under incomplete information. *IEEE Trans. Syst. Man Cybernetics 31,* 6, 533–545.

SEUKEN, S. AND ZILBERSTEIN, S. 2007. Improved memory-bounded dynamic programming for decentralized pomdps. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

SILVER, D. AND VENESS, J. 2010. Monte-Carlo planning in large pomdps. In *Conference on Advances in Neural Information Processing Systems*.

SINGLA, G., COOK, D. J., AND SCHMITTER-EDGECOMBE, M. 2008. Incorporating temporal reasoning into activity recognition for smart home residents. In *Proceedings of the AAAI Workshop on Spatial and Temporal Reasoning*. 53–61.

SPAAN, M. T. J. AND VLASSIS, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *J. Artif. Intell. Res. 24*, 195–220.

STROKE. 2007. Consensus panel on the stroke rehabilitation system "time is function". Heart and Stroke Foundation of Ontario.

STRONG, K., MATHERS, C., AND BONITA, R. 2007. Preventing stroke: Saving lives around the world. *Lancet Neurol. 6,* 2, 182–187.

SUTTON, R., PRECUP, D., AND SINGH, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell. 112*, 181–211.

Theocharous, G., Murphy, K., and Kaelbling, L. 2004. Representing hierarchical POMDPs as DBNs for multi-scale robot localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA 04)*.

Truelsen, T. and Bonita, R. 2009. The worldwide burden of stroke: Current status and future projections. In *Handbook of Clinical Neurology: Stroke Part I: Basic and Epidemiological Aspects*, M. Fisher, Ed., 3rd Series, vol. 92. Elsevier, Amsterdam, 327–336.

Varakantham, P., young Kwak, J., Taylor, M., Marecki, J., Scerri, P., and Tambe, M. 2009. Exploiting coordination locales in distributed pomdps via social model shaping. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Viola, P. A. and Jones, M. J. 2004. Robust real-time face detection. *Int. J. Comput. Vis. 57,* 2, 137–154.

Wherton, J. P. and Monk, A. F. 2009. Problems people with dementia have with kitchen tasks: the challenge for pervasive computing. *Interact. Comput. 22,* 4, 253–266.

Williams, J. D. and Young, S. 2006. Partially observable Markov decision processes for spoken dialog systems. *Comput. Speech Lang. 21,* 2, 393–422.

Wilson, R., Rochon, E., Mihailidis, A., Leonard, C., Lim, M., and Cole, A. 2007. Examining effective communication strategies used by formal caregivers when interacting with alzheimer's disease residents during an activity of daily living (adl). *Brain Lang. 103,* 1-2, 199–200.

Wolf, R. 2007. Advances in phototherapy training. *Arts Psychother. 34*, 123–133.

Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J. M. 2007. A scalable approach to activity recognition based on object use. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Zhang, S., McClean, S., Scotney, B., and Nugent, C. 2008. Learning under uncertainty in smart home environments. In *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference*.

Received May 2011; revised December 2011; accepted March 2012