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

Jesse Hoey
School of Computing
University of Dundee
Dundee, Scotland DD1 4HN

Andrew Monk
Department of Psychology
University of York
York. UK. YO1 5DD

Alex Mihailidis
Dept. of Occupational Therapy
and Occupational Science
University of Toronto, Toronto

Abstract

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 health care, such as assistance for tasks of everyday life (washing, cooking, dressing), medication taking, nutrition and fitness. This paper will present a decision theoretic model for general purpose assistance in the home, and will show how this type of general model can be applied to a range of assistance tasks, including prompting for activities of daily living, 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.

1. Introduction

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 personalised 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 (of Canada 2010). What this implies is 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.

Technology can play a key role in healthcare assistance in the home, primarily by connecting providers and recipients. This connection can take the form of telecare, of shared information, data, logs and resources, among others. The key is that technology can increase 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. Therefore, the economic and social impact of technology for healthcare lies in three key factors:

Poeple/Customizability. In order for people to have control over technology, the technology needs to be customisable. 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, sensing technology must be used to provide valuable information for carers and health professionals. Examples include monitoring patients in hospital or at home, smart-home technology to assist persons in independent living, and emergency response systems that detect and respond to falls. In order to ensure uptake of technology by informal carers and users, our models use general purpose sensor data, and must learn from observing a person.

Decisions/Adaptivity. People require assistance with their healthcare needs, but their requirements can change over time. While customisation is one element of allowing people to make technology fit their needs, the technology 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. Therefore, technological solutions must be able to take decisions about assistance for people, and these decisions must be adaptive.

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. 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.

The COACH (Cognitive Orthosis for Assistive aCtivities in the Home), is a system that can monitor and prompt an older adult through a variety of activities of daily living. More details on the COACH can be found in Section 4.1.

Autominder is a system to help users remember the tasks that need to be completed. This system differs from the COACH, in that it provides reminders of activities at a high level as opposed to providing prompts about how to

ICAPS 2010 POMDP Practitioners Workshop, May 12, 2010, Toronto, Canada.

complete the activity itself. It models its client's daily plans, tracks their execution by reasoning about the client's observable behaviour, and makes decisions about whether and when it is most appropriate to issue reminders (Pollack 2006). The Autominder has been deployed on three separate platforms: the robotic assistant Pearl, a robotic walker and on a handheld PDA.

The PROACT system infers 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). Recent work in the same direction has investigated how common sense models of everyday activities can be built automatically using data mining techniques (Pentney, Philipose, and Bilmes 2008).

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. Opportunity Knocks (Liao et al. 2007), a system designed to provide directional guidance to a user navigating through a city, was developed in the Assisted Cognition project.

Partially observable Markov decision processes (POMDPs) provide a rich framework for planning under uncertainty (Åström 1965). For instance, in mobile robotics (Pollack 2006; Montemerlo et al. 2002), in spoken-dialog systems (Williams and Young 2006), and in assistive technology (Mihailidis et al. 2008).

3. 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. A person with AD loses short-term memory, and therefore has difficulty in *recalling* what they are doing, in *recognising* necessary objects like the soap, and in *perceiving* affordances.

3.1 Seven elements of assistance

The seven key elements are as follows.

Task, T: A characterisation 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. These variables need to be specified for each task, but they characterise 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 soap*. 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, their overall level of health, or the physical ability with respect to the task.

Action, A: The actions of 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.

Observations, O: *Task* and *behavior* variables generate observations, K and V, respectively. These observations are generated from either non-invasive 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: describe the dynamics and the observation function, and govern how the first five elements interact.

Reward, R: Each state-action-observation 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 optimise over the reward function in the long term.

Our goal is to design a model of the interactions between these elements, to build a method for customisation of the model, and to optimize an automated strategy for assistance by maximising (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 customisability), 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) fulfills these constraints.

3.2 POMDPs

A discrete-time POMDP consists of: a finite set S of states; a finite set A of actions; a stochastic transition model $\Pr : S \times A \rightarrow \Delta(S)$, with $\Pr(t|s,a)$ denoting the probability of moving from state s to t when action a is taken; a finite observation set O; a stochastic observation model with $\Pr(o|s)$ denoting the probability of making observation o while the system is in state s; and a reward assigning reward $R(s,a,t)$ to state transition s to t induced by action a. Figure 1(a) shows the POMDP as a Bayesian network. Since the system state is not known with certainty, a *policy*, π, maps either *belief states* (i.e., distributions over S) or action-observation *histories* into choices of actions. A policy is computed to maximize some aggregate measure of utility over time. One such measure is the expected discounted sum of rewards, $\sum_t \gamma^t r_t$, where r_t is the reward obtained at time t, and $\gamma \in [0,1]$ is a *discount factor* that makes large and distant (in time) rewards equivalent to small and immediate rewards. We will not delve into details of POMDP solution methods, but note that current research has enabled the approximate solution of very large POMDPs, and we refer to (Lovejoy 1991) for an overview of POMDP concepts and algorithms.

3.3 POMDP model of Assistance

The general POMDP we present models 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 *behaviours* 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 non-specialised person, while the *ability* will require expert knowledge, but will tend to generalise across tasks. On the other hand, the *behaviors* will be much more difficult to specify, but can be learned from data directly (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. 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.

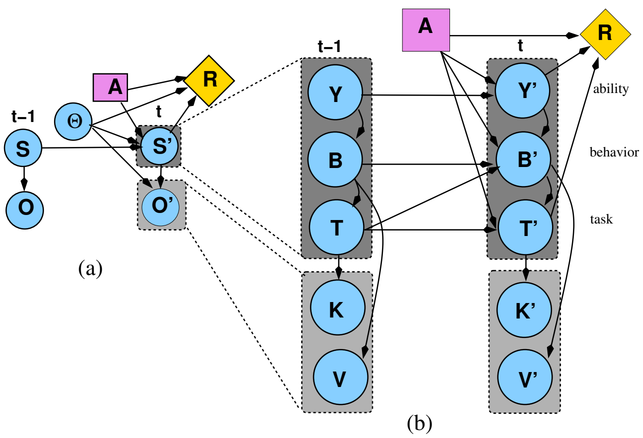


Figure 1: Two time slices of (a) a general POMDP and (b) a factored POMDP for modeling interactions with cognitive assistive technology. The parameters are denoted Θ, and are not shown in (b) for clarity.

Figure 1(b) shows the same model, 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

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

Notice that the task state is independent of the ability, Y, given the behaviour, 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 $Pr(V|B)$.

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 optimise by taking actions that lead to belief states with high reward values.

To compute an approximate policy, we used SymbolicPerseus (Poupart 2005)[1]. It implements a factored, structured point-based approximate solution technique based on the Perseus algorithm (Spaan and Vlassis 2005).

3.4 System elements

We now return to the main goal of our work, to provide customisable, adaptive and general purpose solutions for assistance. To allow for customisation, 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 *actor* controls *agent* to assist *client* while simultaneously reducing *actor* load, maintaining *client* safety and promoting *client* independence. The challenge of modeling such a triad is the differing levels of expertise. We wish to give *actor* the ability to customise *agent*, but without the need for extensive training. The customisation should be simple and effective, and the end result should be helpful for *actor* and adaptive to *client*.

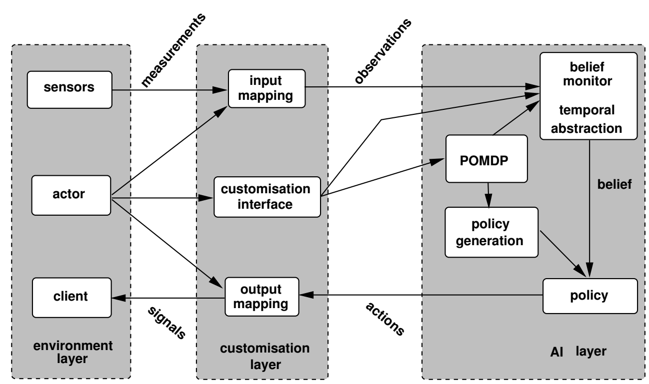


Figure 2: System framework for assistance showing three layers: Environment, customisation, and artificial intelligence (agent) layers.

The framework in Figure 2 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. The *customization layer* in the middle includes three elements that can be specified by end users: a *customisation 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 4.3 and 4.4. The third layer is the *AI layer* on the right, and includes the POMDP model parameters, Θ, 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

[1]see www.cs.uwaterloo.ca/∼ppoupart/software

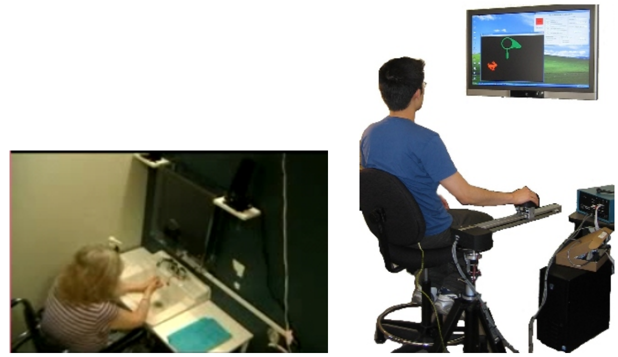
frames per second in video), whereas the POMDP operates on *event-driven* time. Updates in the POMDP need to happen at realistic and natural time frames for the *behaviours*. For example, during handwashing, it takes only a few seconds to turn on the water, but 10-30 seconds to rinse the soap. The simplest 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 timeout can be computed as a weighted average, $\tau = \sum_s b(s) \cdot \texttt{timeout}(s)$, or as the most likely state's timeout, $\tau = \texttt{timeout}(\arg\max_s b(s))$. 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.

4. Applications

This section presents five applications of the model we have presented above. 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 customisability 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 customisation as a central issue. The last application demonstrates how the same model can be used for hierarchical control in assistive systems. Table 1 details the various elements of the four applications: their state variables go (task, behaviour and ability), system actions, input/output mappings, and their customisation abilities.

4.1 Prompting for ADL: COACH

COACH is a real-time 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, Fernie, and Barnebel 2001; Boger et al. 2005; Hoey et al. 2010b). There is a single sensor in the most recent prototype: a video camera. The video is fed to an input mapping that is pre-defined as a computer vision algorithm that tracks the two hands and the towel, and then outputs the locations of these objects discretised into a small set of regions. The POMDP includes a set of eight plansteps as *task* variables, a set of six simple behaviours, 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 planstep at one of three levels of specificity. Specific prompts are more costly than generic prompts, as they decrease feelings of independence in users (Mihailidis, Fernie, and Barnebel 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 3(a) shows the system working, with audio-visual prompting. The camera is out of sight on the ceiling above the sink.



(a) (b)

Figure 3: (a) COACH handwashing assistant (b) iSTRETCH haptic robotic device.

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 et al. 2010b), 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 discretised into a fixed set of regions around each element of interest in the sink area. 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 modelled as a POMDP with 9 state variables, 3 observation variables, and 25 actions. There are 207360 states and 198 observations (Hoey et al. 2010b). The *task* variable is a set of plansteps 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 *behaviours* cause transitions in the plansteps. The user behaviors can be one of six activities: using soap, at water, at tap, at sink, drying, or away. User *ability* is modelled 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 they are doing in the task; and responsiveness = [none,max,med,min], giving what type of prompts the user is responsive to (Hoey et al. 2010b).

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) (Hoey et al. 2010b). 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 prior experience, and was fixed. The other two actions are the 'null' action and 'call caregiver.' The latter action ends the process and is presumed to result in successful task completion. The COACH does not provide any ability to customise, as it has been developed for a particular task and user group.

The reward function gives a large positive reward for the

	COACH	iSTRETCH	ePAD	SNAP
task	plansteps (a-k)	game state	screen state	cup position, cup contents, box condition
behaviour	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
customis-ability	none	input/output mappings, parameters	input/output mappings, parameters	input/output mappings, parameters, POMDP structure

Table 1: Table showing properties of each application, broken down into state variables, actions, input/output mappings, and customisability. Null actions and behaviors are not shown.

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. Our clinical findings (i.e. the effect of the system on users) are reported in (Mihailidis et al. 2008), whilst our technical findings, with more technical details are reported in (Hoey et al. 2010b).

Learning of behaviours has been investigated in the context of the COACH handwashing system. Given a set of data, we wish to learn automatically what behaviours are present and how these behaviours are related to the task. Behaviours 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 behaviours into groups that are simultaneously recognisable (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 behaviours can be automatically extracted (Hoey et al. 2005). We have also more recently investigated supervised learning of behaviours (Peters, Wachsmuth, and Hoey 2009).

4.2 Stroke Rehabilitation: iSTRETCH

Stroke is the leading cause of physical disability and third leading cause of death in most countries around the world. The consequences of stroke are devastating with approximately 75% of stroke sufferers being left with a permanent disability. 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, Krebs, and Hogan 2004). However, this recovery process is slow and labor-intensive, usually involving extensive interaction between one or more therapists and one patient. Rehabilitation robots can partially automate these interventions, provide intensive and reproducible movement training, and be used for assessment by therapists.

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, Hoey, and Mihailidis 2008; Lam et al. 2008).

The robotic device, as detailed in (Lam et al. 2008) and shown in Figure 3(b), was built by Quanser Inc., a Toronto-based robotics company. It features a non-restraining 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 virtual environment provides the user with visual feedback on hand position and target location.

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. The *task* state is only related to the virtual game, and encodes things such as whether the user has completed a level. The *behaviours* model the time it takes the user to reach the target, the amount of control they exhibit 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 sensors). The user's *abilities* are modeled as a product of three factors: their range at each resistance level, their level of fatigue, and their 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. 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 customisation of the POMDP model. More details can be found in (Kan, Hoey, and Mihailidis 2008).

The output can also be customised 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.

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, 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 enrolment, 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 their 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 e.g. setting the cost for the stop action to be higher.

4.3 Art Therapy: ePAD

Engagement with visual artworks is also known to have benefits for the promotion of quality of life in older adults (Rusted, Sheppard, and Waller 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.

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

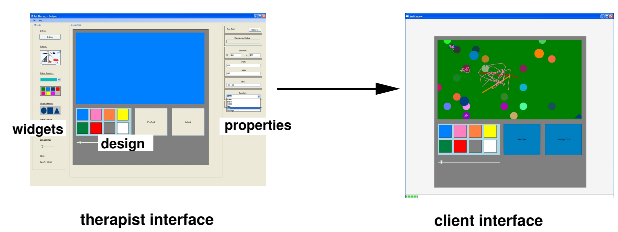


Figure 4: Art Therapist interface and client application.

The therapist uses the designer tool, which is a screen split into three sections: *widgets*, *design*, and *properties* (see Figure 4). The therapist can drag any widget onto the design area, configure it to their needs with a mouse , and set properties and actions for the system. Once done, the therapist exports the newly created application for use by the client (Figure 4 on the right). When using the newly created application, a user does some visual art-work on a touch screen. 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 behaviour of a user on the screen to a category of *involvement* $\in$ {interactive,active,intermittent,inactive}, defined as the amount of *engagement* a behaviour 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 *behaviours* we expect to see in clients of this application. The **behaviour**$\in\{$*interactive,active,intermittent,inactive*$\}$ is whether the user is actively doing something on the interface, and is inferred from observations of their finger interactions. 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 user ability model is given by two factors: 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 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 ($a_1, \ldots, a_M$), given as 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. The therapist can change the *interactivity level* $\in\{$*low, high, stimulate*$\}$ for each action. 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 them 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 their previous behaviours. The user's responsiveness comes into play when the system takes an action. If they are responsive to the interactivity level of the action, the effect of the action is to increase their 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. More details on this system can be found in (Blunsden et al. 2009), and simulated examples of this system in use can be found in (Hoey et al. 2010c).

4.4 Syndetic Assistance Processes: SNAP

Our final application area is focused almost entirely on the customisation 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 new problem needing a solution, be able to design and customise a situated prompting system with a POMDP back-end such as described above.

The key to such a development is the formalisation of a model of assistance (as above), 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 above. We will use an example in this case of a person needing assistance in making a cup of tea.

The task analysis technique, as described in (Wherton and Monk 2009), breaks a particular task down into a set of goals, states, abilities and behaviours. The technique involves an experimenter video-taping a person being assisted during the task, and then transcribing and analysing 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. For example, one element of the state in tea-making is that the cup contains a teabag, a goal of the client. The client's action is to place the teabag in the cup, but this requires three key client *abilities*: to *recall* that they are making tea, to *recognise* the box of tea, and to *perceive the affordances* of the teabag for making tea, as shown in Table 2. This model of cognitive abilities is defined *a priori* by experts in the psychology of dementia, but generalises across tasks, as mentioned in Section 3.3.

The IU analysis can be converted to a POMDP model by mapping the states to *task* variables, the recall, recognition and affordance elements to *ability* variables, and the actions to *behaviours*. In the associated POMDP, the abilities are related to the behaviours in the same way as in the framework above: each behaviour 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 recall the step (recall that they are trying to put the teabag in the cup), recognise the box, and see the affordance of opening the box.

The system actions are the things the system can do to help the user. We define one 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. For example, to help with recognition of the cup, a light could be shone on it, or an audio prompt could be delivered.

The observations are specified by a ubiquitous sensing expert, and are related to each state (task) variables or user behaviour. For example, in the ambient kitchen, there are sensors in the counter-tops 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 dynamics and initial state are produced directly from the IU analysis. 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). Each action improves its associated cognitive ability. For example, the 'prompt recognition cup' action (e.g a light shone on the cup) makes it more likely that the user can recognise the cup if they can't already.

The reward function specifies the 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. The actions are costly because we want to let a user do it themselves if they can.

We have developed software for the specification of situated prompting systems using the SNAP method, and have

IU	Goals	Task States	Abilities	Behaviours
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	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		

Table 2: IU analysis of the first step in tea making. Rn=recognition, Rl=Recall, Af=Affordance, tb=teabag, ws=work surface.

tested the method in a kitchen using tea making (Hoey et al. 2010a). Our future plans include testing on other tasks, and advancing the specification towards end-users.

4.5 Hierarchical Control

Hierarchical control for situated prompting systems poses a number of challenges, primarily due to the interleaved and concurrent nature of sub-goals. The cognitive modeling approach of (Wherton and Monk 2009), as explored in Section 4.4, defines a user's mental state as a *goal stack*, onto which sub-goals must be pushed by a person. Persons with dementia have difficulty pushing sub-goals onto this stack, and often lose track of which sub-goal they are completing (leading to a sub-goal being popped from the stack). The pushing of a sub-goal onto this stack is a user behaviour, but a fundamentally different one from other physical acts (e.g. scooping some sugar), or psychological acts (such as recognising the teacup), both of which only serve to immediately further the following behaviour on the part of the user.

Therefore, we can imagine a higher level POMDP controller whose job is only to deal with a particular level of user sub-goals. Once a user has pushed a particular sub-goal onto their stack, this high-level controller passes off control to a lower-level sub-goal controller that guides the user through the individual steps related to that sub-goal. If the person loses track of what they are doing in that sub-goal, the lower-level controller will attempt to prompt them to get them back on track. If they do not respond to such prompts, the higher-level controller will start to believe that the person has lost the sub-goal from their stack (the belief in their ability to recall the sub-goal will decrease), and will step in and issue an appropriate reminder of that sub-goal, vetoing any actions recommended by the lower-level controllers. Such higher level controllers can be arbitrarily nested, as the sub-goals and recall abilities they deal with can be defined at any level of abstraction. This is a key element of deploying such technology on a larger scale, in which multiple ADL prompting systems act in concert for assistance at home.

Such a hierarchical controller is an instance of the assistance POMDP described in Section 3.3, 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 sub-tasks. The composite controllers, on the other hand, receive observations from other controllers (either atomic or composite), and rely on the sub-goals for the majority of actions. Each composite controller has a set of N sub-controllers, 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 *behaviour* variable, B, gives the sub-goal the user is currently attempting, as reported by the sub-controllers. The behaviour 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 sub-goal, and condition what a user is expected to do next.
- The *task* variables include variables denoting whether that sub-goal has been completed yet. Subgoals complete at some finite rate **if** the user behaviour corresponds correctly to the current control sub-goal and all pre-requisites are fulfilled. Finally, a *control* variable gives the current sub-goal that should be being attempted by the user. Note that this may be different from the behaviour (if the user is doing the wrong thing).
- The reward function for this controller corresponds to the ordering of goals to complete the task.
- Sub-goal pre-requisites are encoded in the composite controller as behaviour relevance functions: they give the situations in which each behaviour is relevant, and will be zero for behaviours and situations that depend on previous subgoals according to the prerequisites.

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

1. It has a veto power over the current control sub-goal: if it decides to take an action, the sub-controller is forced to "do nothing". These actions will be taken in situations when the composite controller does not believe the sub-controller will be able to complete.
2. It decides on the sub-goal in control by consulting it's own *control* variable and choosing the most likely control sub-goal.

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 they are not currently attempting.

5. Conclusions

This paper has described a general purpose framework for the specification, customisation and use of decision-theoretic prompting systems for persons with cognitive and physical disabilities. The method is based on the partially observable Markov decision process, and combines elements allowing for user customisation, system adaptivity to users, and general-purpose sensing abilities. This paper has given a detailed presentation of the method, and four case studies of applications of the method to situated prompting systems. Our future work is to refine these methods to allow for more complete end-user customisation, and to bring these types of models into the homes of persons with cognitive and physical disabilities wishing to stay at home.

Acknowledgements The authors thank the American Alzheimer's Association for generous funding of this research. The authors also wish to thank Patrick Olivier for use of the Ambient kitchen.

References

Åström, K. J. 1965. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications* 10:174–205.

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.; Poupart, P.; Hoey, J.; Boutilier, C.; Fernie, G.; and Mihailidis, A. 2005. A decision-theoretic approach to task assistance for persons with dementia. In *Proc. IEEE International Joint Conference on Artificial Intelligence*, 1293–1299.

Fasoli, S.; Krebs, H.; and Hogan, N. 2004. Robotic technology and stroke rehabilitation: Translating research into practice. *Topics in Stroke Rehabilitation* 11(4):11–19.

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.

Hoey, J., and Little, J. J. 2007. Value-directed human behavior analysis from video using partially observable Markov decision processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(7):1118–1132.

Hoey, J.; Poupart, P.; Boutilier, C.; and Mihailidis, A. 2005. Semi-supervised learning of a POMDP model of patient-caregiver interactions. In *Proc. IJCAI Workshop on Modeling Others from Observations*, 101–110.

Hoey, J.; Plötz, T.; Jackson, D.; Monk, A.; Pham, C.; and Olivier, P. 2010a. Rapid specification and automated generation of prompting systems to assist people with dementia.

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. *Computer Vision and Image Understanding*.

Hoey, J.; Zutis, K.; Leuty, V.; and Mihailidis, A. 2010c. Development of an intelligent customisable tool to promote prolonged engagement of persons with dementia in art therapy. In *To appear.*

Kan, P.; Hoey, J.; and Mihailidis, A. 2008. Stroke rehabilitation using haptics and a POMDP controller. In *AAAI Fall Symposium on Caring Machines: AI in Eldercare*.

Kautz, H.; Arnstein, L.; Borriello, G.; Etzioni, O.; ; and Fox, D. 2002. An overview of the assisted cognition project. In *Proc. AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, 60–65.

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. *Journal of NeuroEngineering and Rehabilitation* 5(15):1–13.

Liao, L.; Patterson, D.; Fox, D.; and Kautz, H. 2007. Learning and inferring transporation routines. *Artificial Intelligence* 171(5-6):311–331.

Lovejoy, W. S. 1991. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research* 28:47–66.

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 Geriatrics* 8(28).

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. *Assistive Technology* 13:23–39.

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*. Edmonton, Canada: AAAI.

of Canada, A. S. 2010. Rising tide: The impact of dementia on canadian society. http://www.alzheimer.ca/english/rising_tide/rising_tide_report.htm. ISBN 978-0-9733522-2-1.

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*.

Pentney, W.; Philipose, M.; and Bilmes, J. 2008. Structure learning on large scale common sense statistical models of human state. In *Proc. AAAI*.

Peters, C.; Wachsmuth, S.; and Hoey, J. 2009. Learning to recognise behaviours 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. *Pervasive Computing* 10-17.

Pollack, M. E. 2006. Autominder: A case study of assistive technology for elders with cognitive impairment. *Generations: The Journal of the American Society on Aging* 30(2):67–69.

Poupart, P. 2005. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D. Dissertation, University of Toronto, Toronto, Canada.

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 Analysis* 39(4):517–536.

Spaan, M. T. J., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research* 24:195–220.

Viola, P. A., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 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. *HCI*.

Williams, J. D., and Young, S. 2006. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language* 21(2):393–422.