SN̈AP: SyNdetic Assistance Processes

Jesse Hoey
School of Computer Science,
University of Waterloo,
Waterloo, Ontario, N2L 3G1, CANADA

Andrew Monk
Department of Psychology,
University of York,
York, YO10 5DD, UK

Thomas Plötz, Dan Jackson, Cuong Pham, Patrick Olivier
School of Computing Science,
University of Newcastle,
Newcastle upon Tyne, NE1 7RU, UK

Abstract

Activity recognition in intelligent environments could play a key role for supporting people in their activities of daily life. Partially observable Markov decision process (POMDP) models have been used successfully, for example, to assist people with dementia when carrying out small multi-step tasks such as hand washing. POMDP models are a powerful, yet flexible framework for modeling assistance that can deal with uncertainty and utility in a theoretically well-justified manner. Unfortunately, POMDPs usually require a very labor intensive, manual setup procedure. This paper describes a knowledge driven method for automatically generating POMDP activity recognition and context sensitive prompting systems for a complex tasks. We call the resulting POMDP a SN̈AP (SyNdetic Assistance Process). The method starts with a psychologically justified (syndetic) description of the task and the particular environment in which it is to be carried out that can be generated from empirical data. This is then combined with a specification of the available sensors and effectors to build a working prompting system that tracks a person's activities and learns their abilities by using sensor data as evidence in the context of the SN̈AP POMDP. The method is illustrated by building a system that prompts through the task of making a cup of tea in a real-world kitchen.

1 Introduction

Dementia is an important problem with serious effects on the society because of the changing demographic towards an aging population The dependency ratio (the ratio of those typically not in the labor force and potentially needing care to those typically in the labor force and thus able to provide care) is increasing. For example, the old age dependency ratio (the projected number of persons aged 65 and over expressed as a percentage of the projected number of persons aged between 15 and 64) in the European Union is projected to increase from 25.9 in 2010 to 50.4 in 2050 [1]. At the same time the number of people with dementia is increasing. The number of persons with Alzheimer's disease worldwide, for example, is expected to double and will top 100 million by the year 2050 [2]. This means that the burden of care will have to shift from the professional arena (e.g. hospitals and clinics) into the home and community.

Many people with dementia wish to remain living in their own homes as long as possible. However, they generally require some assistance in order to do so. Difficulties performing activities of daily living at home, such as preparing food, washing themselves, or cleaning, may trigger the need for personal assistance or relocation to residential care settings [9]. Moreover, it is associated with diminished quality of life, poor self-esteem, anxiety, and social isolation for the person with dementia and their caregiver [4].

Technology to support people in their need to live independently is currently available in the form of personal and social alarms and environmental adaptations and aids. Looking to the future, we can imagine intelligent, pervasive computing technologies using sensors and effectors that help with more difficult cognitive problems in planning, sequencing and attention. A key problem in the construction of such intelligent technologies is the automatic analysis of people's behaviors from sensory data. Activities need to be recognized and – by incorporating domain specific expert knowledge – reasonable conclusions have to be drawn which ultimately enables the environment to perform appropriate actions through a set of actuators. In the example of assisting people with dementia, the smart environment would prompt whenever the residents get stuck in their activities of daily living.

The technical challenge of developing useful prompts and a sensing and modeling system that allows them to be delivered only at the appropriate time is hard but achievable. Certainly the most sophisticated of these is the COACH system [12]. COACH uses computer vision to monitor the progress of a person with dementia washing their hands and prompts only when necessary. COACH uses a partially observable Markov decision process (POMDP), a temporal probabilistic model that represents a decision making process based on environmental observations. The COACH model is flexible in that it can be applied to other tasks [11]. However, each new task requires substantial re-engineering and re-design to produce a working assistance system, which currently requires massive expert knowledge for generalization and broader applicability to private home scenarios. An automatic generation of such prompting systems would substantially reduce the manual efforts necessary for creating assistance systems, which are tailored to specific situations and tasks, and environments. In general, the use of *a-priori* knowledge in the design of assistance systems is a key unsolved research question. Researchers have looked at specifying and using ontologies [6], information from the Internet [18], logical knowledge bases [5, 14], and programming interfaces for context aware human-computer interaction (HCI) [21].

In this paper we present a knowledge driven method for automatically generating POMDP activity recognition and context sensitive prompting system for a kitchen task, as an example. The developed approach starts with a description of a kitchen task and the kitchen in which it is to be carried out that is relatively easy to generate. Interaction Unit (IU) analysis [20], a psychologically motivated method for transcoding interactions relevant for fulfilling a certain task, is used for obtaining a formalized, i.e., machine interpretable task description. This is then combined with a specification of the available sensors and effectors to build a working model that is capable of analyzing ongoing activities and prompting someone. The method is illustrated by building a system to prompt someone through the task of making a cup of tea in a particular kitchen.

The long-term goal of this approach is to allow end-users, such as health professionals, to specify and develop their own context sensitive prompting systems for needs as they arise. In the fullness of time, a proper evaluation would involve implementing the idea in a clinical or commercial service and then evaluating that service in terms of the value to a specific population of people that it is intended to serve.

2 Overview of the method

2.1 Partially observable Markov decision process models

A POMDP is a probabilistic temporal model of a system interacting with its environment [3], and is described by (1) a finite set of state variables, the cross product of which gives the state space, S; (2) a set of observation variables, O (the outputs of some sensors); (3) a set of system actions, A; (4) a reward function, $R(s, a, s')$, giving the relative utility of transiting from state s to s' under action a; (5) a stochastic transition model $Pr : S \times A \rightarrow \Delta S$ (a mapping from states and actions to distributions over states), with $Pr(s'|s, a)$ denoting the probability of moving from state s to s' when action a is taken; and (6) a stochastic observation model with $Pr(o|s)$ denoting the probability of making observation o while the system is in state s. Figure 1(a) shows a POMDP as a Dynamic Bayesian network (DBN) with actions and rewards, where arrows are interpretable as causal links between variables.

A POMDP for assistance of persons with dementia breaks the state space down into three key factors as shown in Figure 1(b): states describing elements of the functional task in the real world, T, e.g. whether the water has been boiled or not (the "task factor model"), states capturing the user's cognitive capacities, Y, e.g., to remember what they are supposed to do next (the "ability factor model"), and states capturing an inferred history of what the user has actually done since the last update, B, e.g. fill the kettle (the "behavior factor model"). We use the word "behavior" here to

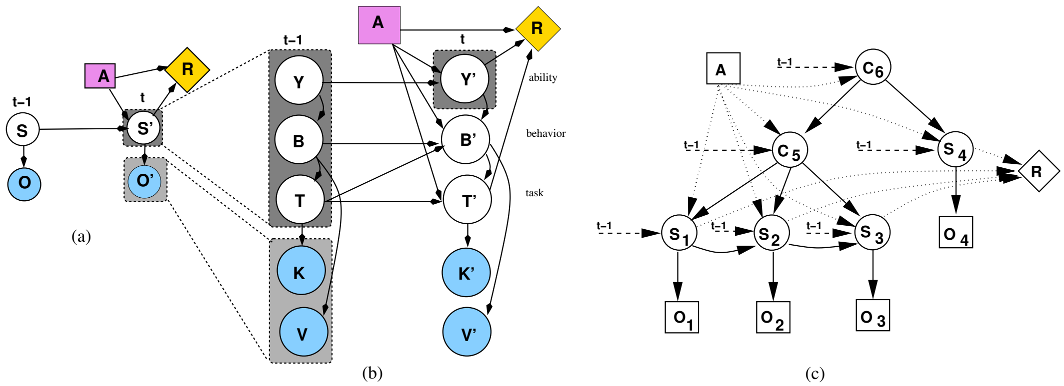

Figure 1: Two time slices of (a) a general POMDP; (b) a factored POMDP for modeling interactions with cognitive assistive technology; (c) Example hierarchical model for task assistance. The dashed lines are connections from the previous time step. Dotted and dashed lines are used only for clarity.

describe actions of the user to distinguish them from actions of the system (i.e., prompts to the user). A preliminary version of this model was explored and used in different contexts in [11, 16, 12]. As we will see, these three factors relate to the psychological analysis, and keeping them separate will ease the translation between the two.

The observations O and O' are states of the sensors at times $t-1$ and t. It is convenient to divide O into the states of sensors relevant to the task environment (K) and states of sensors relevant to user behavior (V). Whereas the first type of sensors monitors, for example, movements of objects in the environment, the latter corresponds to virtual sensors that, for example, monitor the activities the acting person pursues. The system's actions are various prompts or memory aids the system can use to help a user remember things in the task. For persons with dementia, simple audio prompts are often very effective, and are used in the COACH system along with video demonstrations [16, 12]. Finally, the observations are any information from sensors in the environment that can give evidence to the system about the user's behaviors or the task.

2.2 Generating a psychologically justified model for each new task and environment

A key problem is the initial specification of the model for a particular task. The COACH system has been specified manually by technology designers after consultation with end users (persons with dementia and carers) over a number of years. This process is extremely time consuming and difficult to generalize across tasks. In this paper, we describe a method to automatically generate a POMDP-based prompting system using a syndetic modeling technique to capture the circumstances of a particular user, task and environment. We call the resulting prompting system a SN̈AP (SyNdetic Assistance Process).

The methodology is an interaction between two designers: a trained human factors annotator, and a ubiquitous sensing technician. First, the human factors annotator uses the syndetic Interaction Unit (IU) modeling technique described below, and applies it to videos showing a person attempting to do the task with the help of a human assistant. The end-result is an IU analysis that uncovers the various elements of the task, the user's cognitive abilities, and the user's behaviors. Second, the ubiquitous sensing technician proposes a set of sensors and actuators that can be retrofitted to the user's environment for the particular task, and provides a specification of the sensors that consists of three elements: (1) a name for each sensor and the values it can take on (e.g. on/off); (2) a mapping from sensors to the states and behaviors as given by the human factors annotator showing the evidentiary relationships, and (3) measurements of each sensor's reliability at detecting the states/behaviors it is related to in the mapping.

The IU analysis and the sensor specification are then fed into a software that automatically generates a SN̈AP POMDP for this particular user/task/environment combination. The SN̈AP POMDP is solved to yield a policy of action, and a generic software controller then runs the SN̈AP POMDP, taking input from the sensors through the ubiquitous sensing technician's interface. In the following, we first describe the IU analysis, followed by the sensor modeling, and the conversion to a SyNdetic Assistance Process.

3 Specifying the task and environment: Interaction Unit Analysis

The starting point for the automatic generation of a POMDP prompting system is a psychologically justified description of the task and the particular environment in which it is to be carried out. To illustrate the method we are using we thus needed a real example of someone with dementia carrying out a real task that has not previously been modeled using a POMDP. Serendipitously we had access to videotapes of the same woman (JF) making a cup of tea on two occasions in her own kitchen. These are part of a collection that was the basis of an analysis of the problems that people with dementia have with kitchen tasks [22]. JF has dementia of the Alzheimer's type and lives with her husband who now does all the cooking. They store tea and coffee making items on a tray on the counter as they believe this helps her when making hot drinks for herself. She can do other tasks alone (e.g., dressing and cleaning). They lived in the same house before the onset of dementia when she used to do all the kitchen tasks.

The POMDP prompting system was built into the Ambient Kitchen, a high fidelity prototyping environment for pervasive technologies [17] at Newcastle University (figure 2). The videos were used to select appliances and utensils similar to those used by JF.

Figure 2: Set-up of the test environment used for the SNAP evaluation. Left: Overview of the Ambient Kitchen in Culture Lab at Newcastle University. Right: Close-up of the main work surface for the tea preparation task.

Task analysis has a long history in Human Factors [13] and Occupational Therapy [8] where tasks are referred to as "activities of daily living" (ADL). In both cases the emphasis is on describing the actions taken by a user and the intentions (goals and sub-goals) that give rise to those actions. There has been less emphasis on how actions are driven by the current state or changes in the environment. Syndetic modeling [7] remedies this omission by describing the conjunction of cognitive and environmental precursors for each action. Modeling both cognitive and environmental mechanisms at the level of individual actions turns out to be much more efficient than building separate cognitive and environmental models [20].

3.1 IU analysis

The task analysis technique [22], breaks a particular task down into a set of goals, states, abilities and behaviours, and defines a hierarchy of tasks that can be mapped to a POMDP, a policy for which will be a situated prompting system for a particular task. The technique involves an experimenter video-taping a person being assisted during the task, and then transcribing and analysing the video. The end-result is an Interaction Unit (IU) analysis that uncovers the states and goals of the task, the client's cognitive abilities, and the client's actions. A simplified example for the first step in tea-making (getting out the cup and putting in a tea-bag) is shown in Table 1. The rows in the table show a sequence of steps, with the client's current goals, the current state of the environment, the abilities that are necessary to complete the necessary step, and the behaviour that is called for. The abilities are broken down into ability to *recall* what they are doing, to *recognise* necessary objects like the kettle, and to *perceive* affordances of the environment.

The IU analysis shown in Table 1 can be converted to a POMDP model by factoring the state space as shown in Figure 1(b). The method is described in detail in [10], here we give a brief overview. The *task* variables are a characterisation of the domain in terms of a set of high-level variables, and correspond to the entries in the state column in Table 1. For example, in the first step of tea making, these include the box condition (open, closed) and the cup contents (empty or with teabag). The task states are changed by the client's *behavior*, B, a single variable with values for each behaviour in Table 1. For the first IU group in tea making, these include opening/closing the box, moving the

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

teabag to the cup, and doing nothing or something unrelated (these last two behaviours are always present). The client's *abilities* are their cognitive state, and model the ability of the client to recall (Rl), recognise (Rn) and remember affordances (Af). For the first IU group, these include the ability to recognise the tea box and the ability to perceive the affordance of moving the teabag to the cup.

The system actions are prompts that can be given to help the client regain a lost ability. We define one system action for each necessary ability in the task. The actions correspond to a prompt or signal that will help the client with this particular ability, if missing. *Task* and *behavior* variables generate observations, O. For example, in a kitchen environment 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) [19, 10].

The dynamics and initial state are produced directly from the IU analysis (Table 1), 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 client's abilities (so we assume a perfectly able client 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 client can recognise the cup if they can't already. The reward function specifies the goal states (in Table 1), and assigns a cost to each prompt, as client independence is paramount.

3.2 Hierarchical control

The IU analysis breaks an ADL like making a cup of tea down into a number of sub-tasks, or sub-goals. For tea making, there are five sub-goals. This decomposition arises naturally according to the major elements of recall noted in the videos from which this IU analysis was made. The five sub-goals are partially ordered, and the partial ordering can be specified as a list of pre-requisites for each sub-goal giving those sub-goals that must be completed prior to the sub-goal in question. Since each sub-goal is implemented as a separate POMDP controller, a mechanism is required to provide hi-level control to switch between sub-goals. The controller we use is very simple, maintaining a current `control` index, and passing all observations made to the `control` sub-goal POMDP. The control is switched when either the `control` POMDP has reached its goal (to within some threshold), or a new sensor measurement (change) is made that does not correspond to the current `control` sub-goal. The newly selected `control` is either the sub-goal that this new sensor measurement is associated with if all pre-requisite sub-goals are complete, or the first pre-requisite sub-goal if not. This allows a user to switch sub-goals during execution, but only to those that respect the partial ordering referred to above.

In fact, the syndetic task analysis contains an explicit reference to client goals being organised in a *stack* structure. We can use this structure and assume that there are two different types of cognitive abilities: *goal-recall* and *behaviour-recall*. The *goal-recall* abilities are those that affect only the mental state of the client and their goal stack. These abilities allow a person to recall a sub-goal that is necessary to complete during the task. For example, if a person is making a coffee, and has put granules in the cup, then they must recall that the next step is to boil the water. This act of recall pushes a new goal onto their goal stack, and has this effect only. The *behaviour-recall* abilities are then required to accomplish the subtask of boiling water (e.g. recognising the kettle), but these call for specific environmental behaviours (e.g. filling the kettle). However, these abilities will not be relevant if they client does not first have the appropriate *goal-recall* ability. The *goal-recall* abilities define a hierarchical breakdown, whilst the *behaviour-recall* abilities define sequential steps within a level in the hierarchy.

Figure 1(c) shows an example for a hierarchical model involving four subtasks (with state spaces S_i $i = 1 \ldots 4$ (possibly containing *behaviour-recall* abilities), with related observation sets (O_i $i = 1 \ldots 4$), and two sets of *goal-recall* abilities C_5 and C_6). This figure is showing the same POMDP model as in Figure 1(b), except we have factored the *goal-recall* abilities C out of Y, and organised these factors graphically in a tree structure for clarity. The tree structure shows that, to perform subtask S_1, a client will need to recall goals C_5 and C_6, and to have abilities for S_1 (in that order). For example, if C_6 is the goal of making a breakfast of tea and toast, then C_5 may be making the tea, which involves getting the teabag out of the box and placing it in the cup (S_1) and then boiling water (S_2) and adding it to the cup (S_3), while S_4 may be making the toast. Note that the tree structure will be specific to each individual and each environment. In the example above, the task of making toast involves no *goal-recall* abilities (other than the recall of the goal of making the toast in the first place). However, some other client may forget that the toast is in the toaster (e.g. an old-fashioned toaster that does not pop up automatically), and require an additional level in the tree for a *goal-recall* ability to get the toast out of the toaster.

The dynamics of the subtasks at the leaves are such that progress toward the goal is only made if the entire path of *goal-recall* abilities leading to the root of the tree are true (the client has these goals on their stack), otherwise, progress will stall (as the client will have forgotten what they are doing). The dynamics can be further complicated by the fact that some subtasks rely on other subtasks to be complete before they can begin (e.g. the arrows $S_1 \rightarrow S_2 \rightarrow S_3$ in Figure 1(c)).

The full model will become intractably large for even a moderate number of subtasks. To handle this complexity, we break the hierarchy into a set of individual controllers, one for each node in the tree, by adding two new variables to each node that are an abstract representation of the state of its parent and its children. The addition of these two variables turns each node in the tree into a POMDP model as shown in Figure 1(b) if we make the association of the child variables with a macro-behaviour/task (indicating which subtask is currently being pursued by the client - behaviour - and which have been completed - task) and of the parent variables with a macro-ability (indicating that the client has all abilities higher up the tree to complete the subtask). This elegant decomposition means that a single class of POMDP model can be used at each node. The formulation can also be viewed as a type of resource allocation problem [15] in which the resource is not fully under the control of the system. This more sophisticated form of hierarchical control is our current work, and the examples we present in the next section only use the simple deterministic controller described at the beginning of this section.

3.3 Implementation issues

The central controller described in the last section and the POMDP controllers for each sub-goal are implemented in Java, and run as separate processes on three PCs with 2 GHz processors and 2GB of RAM. The sensors are sampled at 1 second intervals by an observer process. The central controller polls the observer (at step 3 in the algorithm above), and receives the most recent observations. Note that this polling arrangement may miss brief events. For example, if a person opens and then closes a box during the time when the controller is processing at step 1, the most recent sensor values will be read, and no change will be registered. We will see how this *"sensor memory"* issue affects our results in the next section.

There are a number of solutions for the *sensor memory* problem. The simplest is to adjust the sensor reliabilities to reflect this. The POMDP controllers will then be able to adjust their policies to take into account the additional uncertainty related to this. However, this solution is not ideal as it will lead to less "confident" policies, i.e., policies that are based on belief states with more uncertainty, and only due to a lack of a proper temporal model for the sensor readings. The second solution is to include some additional "sticky" virtual sensors that indicate an event happening in the past (e.g. the box has been opened and then closed again). Our experiments in the following section avoided this heuristic solution in order to clearly demonstrate the abilities of the POMDP model to deal even with incorrectly specified sensor reliabilities.

4 Demonstrative Examples

To test our method, we asked two volunteers who had no knowledge of the system to work through two scenarios derived from the original scenario of JF making tea. In the first, *All-Unresponsive*, the participant was instructed to wait for prompts before doing anything, but to respond to all instructional prompt, i.e., to follow exactly the commands given by the system. In the second scenario, *Subgoal-Unresponsive*, the participant was instructed to hesitate after each subgoal has been completed, and to wait for a prompt before continuing. In both scenarios, participants were asked to ignore inappropriate prompts although these were recorded for our analysis.

Our general criteria for success in this test of the system were that in the *All-Unresponsive* scenario the system would come up with the appropriate prompt for each action at the right time, while in the *Subgoal-unresponsive* scenario the system would only prompt at the junction between subgoals. In general, these criteria were met (see [10] for details). Where they were not we were able to infer some possible refinements that could be made to the process. Snapshots of the system state for the *All-Unresponsive* scenario are shown in Figure 3 (see [10] for details of the *Subgoal-unresponsive* scenario). Each snapshot includes a still of the video footage, observations (i.e. sensor readings at the particular step), and the beliefs of the system at the time the system prompted. For the system beliefs, prefixes are used to associate beliefs to either abilities, i.e. recognition (rn), recall (rl), and affordance (af); behaviors (b); task (no prefix). Five snapshots have been selected that correspond to significant events in the sequence.

SŇAP successfully guided the participant through the tea preparation process. In total it provided 21 prompts of which 16 were appropriate, i.e., they were given at the correct time within the tea preparation process. Otherwise, the system appeared to "do nothing" at the appropriate times. The five inappropriate prompts were either: (1) appropriately timed but inappropriate for the situation (i.e., a prompt was needed but the system gave an incorrect one), or (2) inappropriately timed (i.e. not needed or an incorrect prompt). We will refer to the first type of error as *misprompts* and the second type as *false positive prompts*. The errors were due to one of three possible causes: *sensor errors* are sensors that are not behaving according to the reliability measurements; *sensor memory errors* arise from timing issues as discussed in Section 3.3, and *model errors* are due to incorrect specification of the controllers, i.e., transcription errors between the IU analysis and the POMDP specifications. By analyzing the snapshots, and these errors in particular, we can gain an insight into the SŇAP's behavior and overall capability.

Figure 3(a) shows the initial state, with the cup empty and on the tray, and the teabox closed. The abilities optimistically estimated to be very high (close to 1), to give the person a chance if she is capable of doing the task on her own. The first prompt, in Figure 3(b) cues the person to move the cup to the work surface, which she does, as indicated by the non-zero belief in the behavior `b_mv_cup_ws` in Figure 3(c). The first false positive prompt, shown in Figure 3(d), asked for the tea box, although the tea bag had already been taken and put into the cup. The source of the false prompt was a *sensor memory error* (as explained in Section 3.3) that resulted from the participant rapidly opening and then closing the box without a prompt: the sensor event was missed by the SŇAP controllers. When this box-lid open/close event is missed, the POMDP assigns a very low probability to this event having actually happened (it was assigned the minimum miss rate of 0.1% since it was never observed to occur in our sensor reliability study). Since SŇAP also observed the teabag in the cup, which has a much lower reliability (75% miss rate), it concluded that, with high probability, the teabag is actually not in the cup, and continues to prompt the person to recognize the tea box. If the reliability of the tea box sensor had accounted for this sensor memory problem (i.e. had a lower reliability), confidence in the teabag_in_cup sensor would have been greater, and `prompt_rn_box` prompt less likely to occur. Although erroneously prompting, the subgoal was successfully completed (shown in Figure 3(e), where the goal is reached), since subsequent sensor readings lowered the system's propensity to prompt for recognition of the box. This elegantly demonstrates the capabilities of POMDPs to cope with uncertain and noisy data and incorrect sensor reliability measurements, and to recover from potential dead ends.

5 Conclusions and discussion

This paper has demonstrated a method for generating a working partially observable Markov decision process (POMDP) prompting system from a psychologically justified description of the task and the particular environment in which it is to be carried out (IU analysis) together with a specification of the available sensors and effectors. In the process of developing this method a working system was constructed in the Ambient Kitchen at Newcastle University. This was demonstrated by having volunteers work through certain scenarios of use. The POMDP successfully modeled uncertainty in sensor measurements and in the dynamics of interaction. The resulting controller was able to deal flexibly and gracefully with errors in sensor readings and with unpredicted user interac-

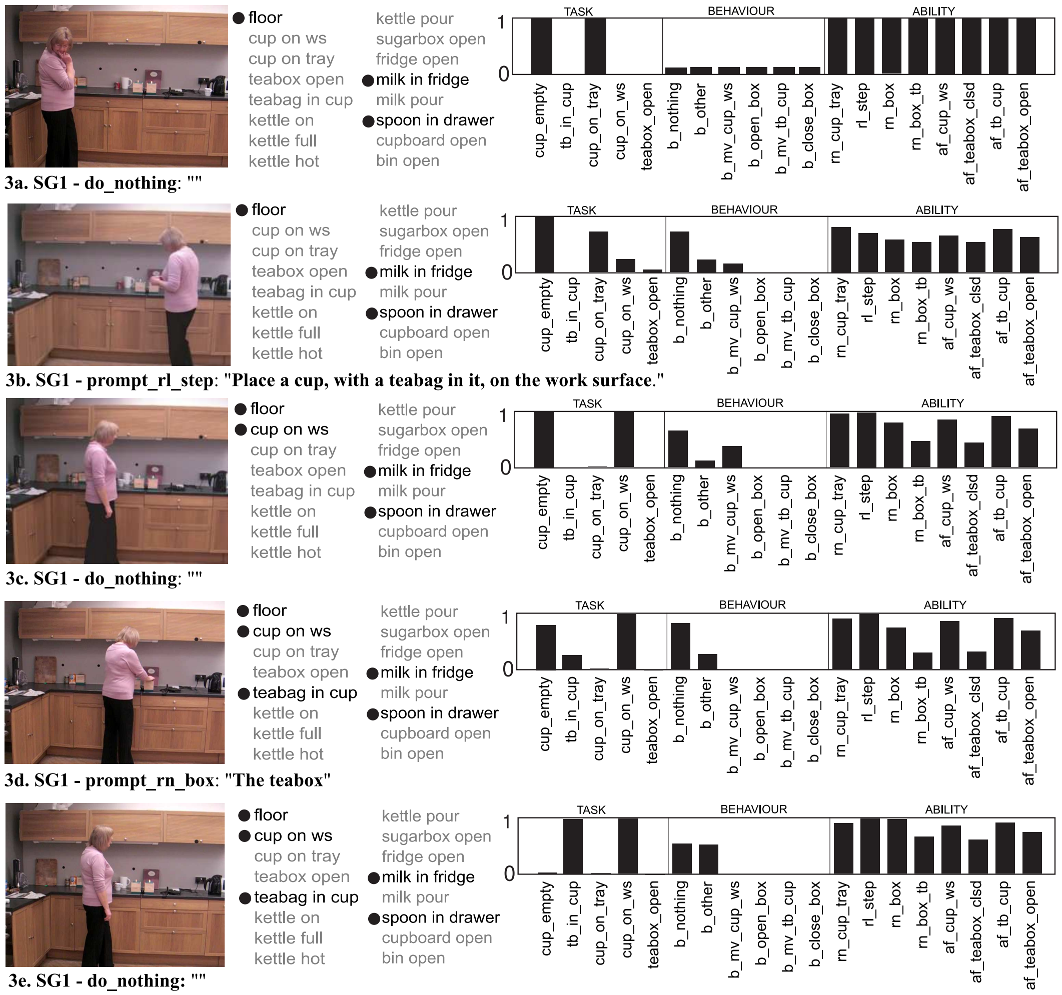

Figure 3: Experiment for *All-Unresponsive* case (selected scenes for overview): snapshots with observations and beliefs. Each snapshot includes a still of the video footage, observations (i.e. sensor readings at the particular step with dark color and a dot indicating the sensor is 'on'), and the beliefs of the system at the time the system prompted. For the system beliefs, prefixes are used to associate beliefs to either abilities, i.e. recognition (rn), recall (rl), and affordance (af); behaviors (b); task (no prefix).

tions. However, this exercise also pointed up a number of outstanding issues still to be addressed, including more sophisticated hierarchical control, sensor error handling, and appropriate prompting.

There were three sources of errors in our initial prototype: sensors that are not responding according to the reliability measurements; sensor memory errors arising from timing issues (Section 3.3), and model errors due to transcription errors between the IU analysis and the POMDP specifications. Perhaps the most challenging errors are timing problem in the implementation. The sensors are operating on a fast update schedule ($\sim 1Hz$), whereas the POMDP controllers are updating much more slowly, in part due to the large size of the models and somewhat inefficient implementations. The mapping between these two timescales was achieved by having the controller simply poll the sensors whenever it needed new readings. This led to sensor memory errors where an event could take place, be registered by the sensors, but go unnoticed by the POMDP controllers, since they were busy doing other processing. Solutions to this problem include "sticky" virtual sensors that maintain some memory of their previous state, ensuring that the sensor reliabilities take timing into account, and making the POMDP operations more efficient.

References

[1] Eurostat, projected old-age dependency ratio. Available from: http://epp.eurostat.ec.europa.eu/portal/page/portal/product_details/dataset?p_product_code=TSDDE511., 2008.

[2] World alzheimer's report. Available from: http://www.alz.co.uk/research/worldreport/, 2009.

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

[4] A. Burns and P. Rabins. Carer burden in dementia. *International Journal of Geriatric Psychiatry*, 15(S1):S9–S13, 2000.

[5] Liming Chen, Chris D. Nugent, M.D. Mulvenna, D.D. Finlay, X. Hong, and M. Poland. A logical framework for behaviour reasoning and assistance in a smart home. *International Journal of Assistive Robotics and Mechatronics*, 9(4):20–34, Dec 2008.

[6] Liming Chen, Chris D. Nugent, M.D. Mulvenna, D.D. Finlay, X. Hong, and M. Poland. Semantic smart homes: Towards knowledge rich assisted living environment. In Sally McClean, Peter Millard, El-Darzi, and Chris Nugent, editors, *Intelligence on Intelligent Patient Management*, pages 279–296. Springer, 2009.

[7] D.J. Duke, P.J. Barnard, D.A. Duce, and J. May. Syndetic modelling. *Human-Computer Interaction*, 13(4):337, 1998.

[8] A.G. Fisher. The assessment of iadl motor skills: An application of many-facet rasch analysis. *American Journal of Occupational Therapy*, 47(4):319–329, 1993.

[9] T.M. Gill and B. Kurland. The burden and patterns of disability in activities of daily living among community-living older persons. *Journal of Gerontology Series A: Biological Sciences and Medical Sciences*, 58A(1):M70–M75, 2003.

[10] Jesse Hoey, Thomas Plötz, Dan Jackson, Andrew Monk, Cuong Pham, and Patrick Olivier. Rapid specification and automated generation of prompting systems to assist people with dementia. To appear, *Pervasive and Mobile Computing*, 2010.

[11] Jesse Hoey, Pascal Poupart, Craig Boutilier, and Alex Mihailidis. POMDP models for assistive technology. In *Proc. AAAI Fall Symposium on Caring Machines: AI in Eldercare*, 2005.

[12] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, May 2010.

[13] B. Kirwan and L.K. Ainsworth. *The task analysis guide*. Taylor and Francis, London, 1992.

[14] Fulvio Mastrogiovanni, Antonio Sgorbissa, and Renato Zaccaria. An integrated approach to context specification and recognition in smart homes. In *Smart Homes and Health Telematics*, pages 26–33. Springer, 2008.

[15] Nicolas Meuleau, Milos Hauskrecht, Kee-Eung Kim, Leonid Peshkin, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Solving very large weakly coupled Markov decision processes. In *Proceedings AAAI*, pages 165–172, Madison, WI, 1998.

[16] Alex Mihailidis, Jen Boger, Marcelle Candido, and Jesse Hoey. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC Geriatrics*, 8(28), 2008.

[17] Patrick Olivier, Andrew Monk, Guangyou Xu, and Jesse Hoey. 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*, Corfu, Greece, 2009.

[18] William Pentney, Matthai Philipose, and Jeff Bilmes. Structure learning on large scale common sense statistical models of human state. In *Proc. AAAI*, Chicago, July 2008.

[19] C. Pham and P. Olivier. Slice&dice: Recognizing food preparation activities using embedded accelerometers. In *European Conference on Ambient Intelligence*, pages 34–43, Berlin: Salzburg, Austria, 2009. Springer-Verlag.

[20] Hokyoung Ryu and Andrew F. Monk. Interaction unit analysis: A new interaction design framework. *Human-Computer Interaction*, 24(4):367–407, 2009.

[21] D. Salber, Anind Dey, and Gregory Abowd. The context toolkit: Aiding the development of context-enabled applications. In *Proc. of the Conference on HUman Factors in Computing Systems (CHI)*, pages 434–441, Pittsburgh, 1999.

[22] Joseph P. Wherton and Andrew F. Monk. Problems people with dementia have with kitchen tasks: the challenge for pervasive computing. *Interacting with Computers*, 22(4):253–266, 2009.