43

Algorithms for Smart Spaces

Diane J. Cook, G. Michael Youngblood, and Gaurav Jain
Department of Computer Science Engineering, The University of Texas at Arlington

To many people, home is a sanctuary. For those people who need special medical care, they may need to be pulled out of their homes to meet their medical needs. As the population ages, the percentage of people in this group is increasing and the effects are expensive as well as unsatisfying. We hypothesize that many people with disabilities can lead independent lives in their own homes with the aid of at-home automated assistance and health monitoring. In order to accomplish this, robust methods must be developed to collect relevant data and process them dynamically and adaptively to detect and/or predict threatening long-term trends or immediate crises. The main objective of this chapter is to describe techniques for using agent-based smart home technologies to provide this at-home health monitoring and assistance. Inhabitant modeling and automation algorithms that are found in smart environments can also provide remote health monitoring for caregivers. Specifically, we address the following technological challenges: (1) identifying lifestyle trends, (2) detecting anomalies in current data, and (3) designing a reminder assistance system. We discuss one such smart environment implementation in the MavHome project and present results from testing these techniques in simulation and with volunteers in an apartment setting.

43.1 INTRODUCTION AND MOTIVATION

Since the beginning, people have lived in places that provide shelter and basic comfort and support, but as society and technology advance there is a growing interest in improving the intelligence of the environments in which we live and work. The MavHome (*M*anaging an *a*daptive *v*ersatile *home*) project is focused on providing such environments [1]. We take the viewpoint of treating an environment as an intelligent agent,

The Engineering Handbook of Smart Technology for Aging, Disability and Independence,
Edited by A. Helal, M. Mokhtari and B. Abdulrazak

which perceives the state of the environment using sensors and acts on the environment using device controllers in a way that can optimize a number of different goals, including maximizing comfort of the inhabitants, minimizing the consumption of resources, and maintaining safety of the environment and its inhabitants. In this chapter we discuss methods by which we can adapt a smart home environment such as MavHome to perform health monitoring and assistance for persons with disabilities and for aging adults.

As Lanspery et al. [2] state, “For most of us, the word ‘home’ evokes powerful emotions [and is] a refuge.” They note that older adults and people with disabilities want to remain in their homes even when their conditions worsen and the home cannot sustain their safety. In a national survey, researchers found that 71% of the respondents felt strongly that they wanted to remain in their current residence as long as possible, and another 12% were somewhat likely to remain there [3]. Nearly 25% of the respondents expected that they or a member of their household would have problems getting around their house in the next 5 years. Of these respondents, 86% stated that they had made at least one modification to their homes to make them easier to live in, and nearly 70% believed that the modifications would allow them to live in the current homes longer than would have otherwise been possible. A separate study supported these results and found that the most common modifications were an easy-to-use climate control system and a personal alert system.

Zola [4] maintains that the problems of aging and disability are converging. Improvements in medical care are resulting in increased survival into old age, thus problems of mobility, vision, hearing, and cognitive impairments will increase [5,6]. As the baby boomers enter old age, this trend will be magnified. By 2040, 23% will fall into the 65 + category [2]. An American Association of Refired Persons (AARP) report [3,7] strongly encourages increased funding for home modifications that can keep older adults with disabilities independent in their own homes.

While use of technology can be expensive, it may be more cost effective than the alternative [8]. Nursing home care is generally paid either out of pocket or by Medicaid. Typical nursing home costs are about $40,000 a year, and the $197 billion of free care offered by family members comes at the sacrifice of independence and job opportunities by the family caregivers.

Our goal is to assist the elderly and individuals with disabilities by providing smart space capabilities that will monitor health trends and assist in the inhabitant’s day to day activities in their own homes. The result will save money for the individuals, their families, and the state.

43.2 OVERVIEW OF THE MAVHOME SMART HOME

Since the beginning, people have lived in places that provide shelter and basic comfort and support, but as society and technology advance, there is a growing interest in improving the intelligence of the environments in which we live and work. We define an *intelligent environment* as one that is “able to acquire and apply knowledge about its inhabitants and their surroundings in order to adapt to the inhabitants and meet the goals of comfort and efficiency” [9]. Smart space algorithms cover a broad spectrum of technologies, including prediction, decisionmaking, robotics, wireless and sensor networking, multimedia, mobile computing, and databases. With these capabilities, the space can adaptively control many aspects of the environment such as climate, water, lighting, maintenance, and multimedia entertainment. Intelligent automation of these activities can reduce the amount of interaction required by inhabitants, reduce energy consumption and other potential wastages, and provide a mechanism for ensuring the health and safety of the environment occupants [10].

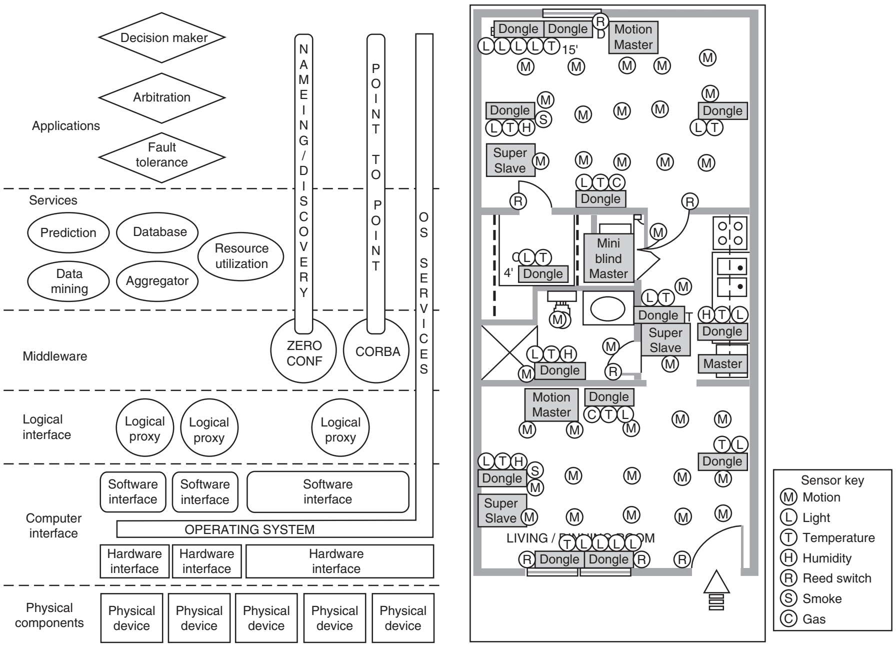
As the need for automating these personal environments grows, so does the number of researchers investigating this topic. Some design-interactive conference rooms, offices, kiosks, and furniture with seamless integration between heterogeneous devices
Q1 and multiple user applications have been constructed in order to facilitative collaborative work environments [11–14]. Abowd and Mynatt's work [15] focuses on ease of interaction with a smart space, and work such as the Gator Tech smart house [16] focuses on development of devices to support elder care. Research on smart environments has become so popular that NIST has identified seamless integration of mobile components into smart spaces as a target area for identifying standardizations and performance measurements [17], although no performance metrics have yet been produced by the group.

Mozer's adaptive home [18] uses neural network and reinforcement learning to control lighting, heating–ventilation–air conditioning (HVAC), and water temperature to reduce operating cost. In contrast, the approach taken by the iDorm project [19] is to use a fuzzy expert system to learn rules that replicate inhabitant interactions with devices, but will not find an alternative control strategy that improves on manual control for considerations such as energy expenditure.

These projects have laid a foundation for the MavHome project. However, unlike related projects, we learn a decision policy to control an environment in a way that optimizes a variety of possible criteria, including minimizing manual interactions, improving operating efficiency, and ensuring inhabitant health and safety. We also ensure that our software need not be redesigned as new devices are registered, new spaces are tested, or new inhabitants move into the environment. To accomplish this goal, our intelligent environment must harness the features of multiple heterogeneous learning algorithms in order to identify repeatable behaviors, predict inhabitant activity, and learn a control strategy for a large, complex environment.

Q2 The MavHome project is focused on providing such environments [1]. We take the viewpoint of treating an environment as an intelligent agent, which perceives the state of the environment using sensors and acts on the environment using device controllers in a way that can maximize the comfort of the inhabitants; minimize the consumption of resources; and maintain safety, security, and privacy of the environment and its inhabitants.

The MavHome architecture shown in Figure 43.1 consists of cooperating layers [9,20]. Perception is a bottom–up process. Sensors monitor the environment using physical components (e.g., sensors) and make information available through the interface layers. The database stores this information, while other information components process the raw information into more useful knowledge (e.g., patterns, predictions). New information is presented to the decisionmaking applications (top layer) on request or by prior arrangement. Action execution flows top–down. The decision action is communicated to the services layer, which records the action and communicates it to the physical components. The physical layer performs the action using powerline control, and other automated hardware, thus changing the state of the world and triggering a new perception.



(a) (b)

FIGURE 43.1 *MavHome architecture (a) and MavPad sensor layout (b).*

All of the MavHome components are implemented and are being tested in two physical environments: the MavLab workplace environment and an on-campus apartment. Powerline control automates all lights and appliances, as well as HVAC, fans, and miniblinds. Perception of light, humidity, temperature, smoke, gas, motion, and switch settings is performed through a sensor network developed inhouse. Inhabitant localization is performed using passive infrared sensors yielding a detection rate of 95% accuracy [21].

Communication between high-level components is performed using common object request broker architecture (CORBA), and each component registers its presence using zero configuration (ZeroConf) technologies. Implemented services include a PostgreSQL database that stores sensor readings, prediction components, data-mining components, and logical proxy aggregators. Resource utilization services monitor current utility consumption rates and provide usage estimates and consumption queries.

MavHome is designed to optimize a number of alternative functions, but for this evaluation we focus on minimization of manual interactions with devices. The MavHome components are fully implemented and are automating the environments shown in Figure 43.2 [22]. The MavLab environment contains work areas, cubicles, a break area, a lounge, and a conference room. MavLab is automated using 54 X-10 controllers, and the current state is determined using light, temperature, humidity, motion, and door/seat status sensors. The MavPad is an on-campus apartment hosting a full-time student occupant. MavPad is automated using 25 controllers and provides sensing for light, temperature, humidity, leak detection, vent position, smoke detection, carbon monoxide detection, motion, and door/window/seat status sensors. Figure 43.1 shows the MavPad sensor layout.

(a) (b)

FIGURE 43.2 *The MavLab (a) and MavPad (b) environments.*

43.3 CORE TECHNOLOGIES

To automate our smart environment, we collect observations of manual inhabitant activities and interactions with the environment. We then mine sequential patterns from these data using a sequence mining algorithm. Next, we predict the inhabitant's upcoming actions using observed historical data. Finally, a hierarchical Markov model is created using low-level state information and high-level sequential patterns, and is used to learn an action policy for the environment. Figure 43.3 shows how these components work together to improve the overall performance of the smart environment. Here we describe the learning algorithms that play a role in this approach.

43.3.1 Mining Sequential Patterns Using ED

In order to minimize resource usage, maximize comfort, and adapt to inhabitants, we rely on machine learning techniques for automated discovery, prediction, and decisionmaking. A smart home inhabitant typically interacts with various devices as part of his/her routine activities. These interactions may be considered as a sequence of events, with some inherent pattern of recurrence. Agrawal and Srikant [23] pioneered work in mining

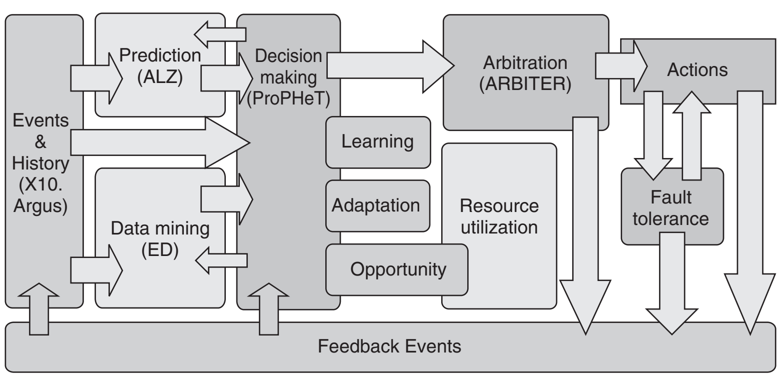


FIGURE 43.3 *Integration of AI techniques into MavHome architecture.*

sequential patterns from time-ordered transactions, and our work is loosely modeled on this approach.

Typically, each inhabitant–home interaction event is characterized as a triple consisting of the device manipulated, the resulting change that occurred in that device, and the time of interaction. We move a window in a single pass through the history of events or inhabitant actions, looking for episodes (sequences) within the window that merit attention. Candidate episodes are collected within the window together with frequency information for each candidate. Candidate episodes are evaluated, and the episodes with values above a minimum acceptable compression amount are reported. The window size can be selected automatically using the size that achieves the best compression performance over a sample of the input data.

When evaluating candidate episodes, the episode discovery (ED) algorithm [24] looks for patterns that minimize the description length of the input stream, O, using the minimum description length (MDL) principle [25]. The MDL principle targets patterns that can be used to minimize the description length of a database by replacing each instance of the pattern with a pointer to the pattern definition. The description length (*DL*) of the input sequence using the set of patterns Θ is thus defined as $DL(O, \Theta) = DL(O|\Theta) + DL(\Theta)$, or the description length of the input sequence compressed using Θ plus the description length of the patterns Θ. The compression of the corresponding encoding can be computed as $\Gamma(\Theta|O) = DL(O)/DL(O, \Theta)$. With this formula, it is easily seen that finding the model that yields the minimum description length of the data is equivalent to finding the patterns that provide the largest compression value, or $MDL(O) = \text{argmax}_{\Theta}\{\Gamma(\Theta|O)\}$.

Our MDL-based evaluation measure thus identifies patterns that balance frequency and length. Periodicity (daily, alternate-day, weekly occurrence) of episodes is detected using autocorrelation and included in the episode description. If the instances of a pattern are highly periodic (occur at predictable intervals), the exact timings do not need to be encoded, (just the pattern definition with periodicity information) and the resulting pattern yields even greater compression. Although event sequences with minor deviations from the pattern definition can be included as pattern instances, the deviations need to be encoded, and the result thus increases the overall description length. ED reports the patterns and encodings that yield the greatest MDL value.

Deviations from the pattern definition in terms of missing events, extra events, or changes in the regularity of the occurrence add to the description length because extra bits must be used to encode the change, thus lowering the value of the pattern. The larger the potential amount of description length compression a pattern provides, the more representative the pattern is of the history as a whole, and thus the potential impact that results from automating the pattern is greater.

In this way, ED identifies patterns of events that can be used to better understand the nature of inhabitant activity in the environment. Once the data are compressed using discovered results, ED can be run again to find an abstraction hierarchy of patterns within the event data. As the following sections show, the results can also be used to enhance performance of predictors and decisionmakers that automate the environment.

43.3.2 Predicting Activities Using ALZ

To predict inhabitant activities, we borrow ideas from text compression, in this case the LZ78 compression algorithm [26]. By predicting inhabitant actions, the home can

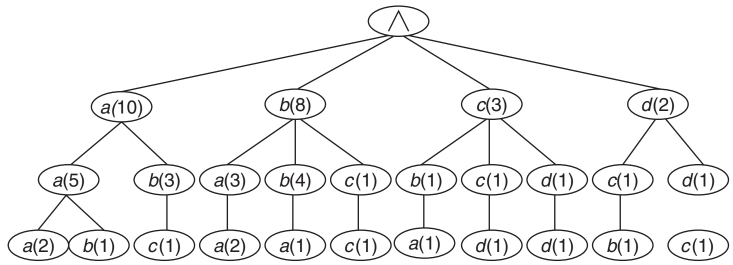


FIGURE 43.4 *Trie formed by ALZ parsing.*

automate or improve on anticipated events that inhabitants would normally perform in the home. Well-investigated text compression methods have established that good compression algorithms also make good predictors. According to information theory, a predictor with an order (size of history used) that grows at a rate approximating the entropy rate of the source is an optimal predictor. Other approaches to prediction or inferring activities often use a fixed context size to build the model or focus on one attribute such as motion [27,28].

LZ78 incrementally processes an input string of characters, which in our case is a string representing the history of device interactions, and stores them in a trie. The algorithm parses the string $x_1, x_2, \ldots, x_i$ into substrings $w_1, w_2, w_{c(i)}$ such that for all $j > 0$, the prefix of the substring w_j is equal to some w_i for $1 < i < j$. Thus, when parsing the sequence of symbols *aaababbbbbaabccddcbaaaa*, the substring *a* is created, followed by *aa*, *b*, *ab*, *bb*, *bba*, and so forth.

Our active-LeZi (ALZ) algorithm enhances the LZ78 algorithm by recapturing information lost across phrase boundaries. Frequency of symbols is stored along with phrase information in a trie, and data from multiple context sizes are combined to provide the probability for each potential symbol, or inhabitant action, as being the next one to occur. In effect, ALZ gradually changes the order of the corresponding model that is used to predict the next symbol in the sequence. As a result, we gain a better convergence rate to optimal predictability as well as achieve greater predictive accuracy. Figure 43.4 shows the trie formed by the active-LeZi parsing of the input sequence *aaababbbbbaabccddcbaaaa*.

To perform prediction, ALZ calculates the probability of each symbol (inhabitant action) occurring in the parsed sequence, and predicts the action with the highest probability. To achieve optimal predictability, we use a mixture of all possible higher-order models (phrase sizes) when determining the probability estimate. Specifically, we incorporate the *prediction by partial match* strategy of *exclusion* [29] to gather information from all available context sizes in assigning the next symbol its probability value.

We initially evaluated the ability of ALZ to perform inhabitant action prediction on synthetic data on the basis of six embedded tasks with 20% noise. In this case the predictive accuracy converges to 86%. Real data collected from six students in the MavLab for one month was much more chaotic, and on these data ALZ reached a predictive performance of 30% (although it outperformed other methods). However, when we combine ALZ and ED by performing predictions only when the current activity is part of a sequential pattern identified by ED, ALZ performance increases by 14% [30,31].

43.3.3 Decisionmaking Using ProPHeT

In our final learning step, we employ reinforcement learning to generate an automation strategy for the intelligent environment. To apply reinforcement learning, the underlying system (i.e., the house and its inhabitants) could be modeled as a *markov decision process* (MDP). This can be described by a four-tuple $< S, A, Pr, R >$, where S is a set of system states, A is the set of available actions, and $R : S \rightarrow R$ is the reward that the learning agent receives for being in a given state. The behavior of the MDP is described by the transition function, $Pr : S \times A \times S \rightarrow [0, 1]$, representing the probability with which action a_t executed in state s_t leads to state s_{t+1}.

With the increasing complexity of tasks being addressed, more recent work in decisionmaking under uncertainty has popularized the use of *partially observable markov decision processes* (POMDPs). Many published hierarchical extensions have allowed for the partitioning of large domains into a tree of manageable POMDPs [32,33]. Research has shown that strategies for new tasks can be learned faster if policies for subtasks are already available [34]. Although a hierarchical POMDP (HPOMDP) is appropriate for an intelligent environment domain, current approaches generally require a priori construction of the hierarchical model. Unlike other approaches to creating a hierarchical model, our decision learner, ProPHeT, actually automates model creation by using the ED-mined sequences to represent the nodes in the higher levels of the model hierarchy.

The lowest-level nodes in our model represent a single event observed by ED. Next, ED is run multiple iterations on these data until no more patterns can be identified, and the corresponding abstract patterns comprise the higher-level nodes in the Markov model. The higher-level *task* nodes point to the first event node for each permutation of the sequence that is found in the environment history. Vertical transition values are labeled with the fraction of occurrences for the corresponding pattern permutation, and horizontal transitions are seeded using the relative frequency of transitions from one event to the next in the observed history. As a result, the n-tiered hierarchical model is thus learned from collected data. An example hierarchical model constructed from MavHome test data is shown on the left in Figure 43.5a.

Given the current event state and recent history, ED supplies membership probabilities of the state in each patterns identified. Using this information along with the ALZ-predicted next action, ProPHeT maintains a belief state and selects the highest-utility action.

To learn an automation strategy, the agent explores the effects of its decisions over time and uses this experience within a temporal-difference reinforcement learning framework [35] to form control policies that optimize the expected future reward. Using the structure defined earlier, the utility value, $Q(s, a)$, is incrementally estimated for state–action pairs. This value represents the predicted future reward that will be achieved if the agent executes action a in state s. After each action, the utility is updated as $Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$. This formula increments the value of $Q(s, a)$ by the reward r received for being in state s' plus a portion of the difference between the current value of Q and the discounted value of $Q(s', a')$, where a' is chosen according to the current Q policy. The current version of MavHome receives negative reinforcement (observes a negative reward) when the inhabitant immediately reverses an automation decision (e.g., turns the light back off) or an automation decision contradicts ARBITER-supplied safety and comfort constraints.

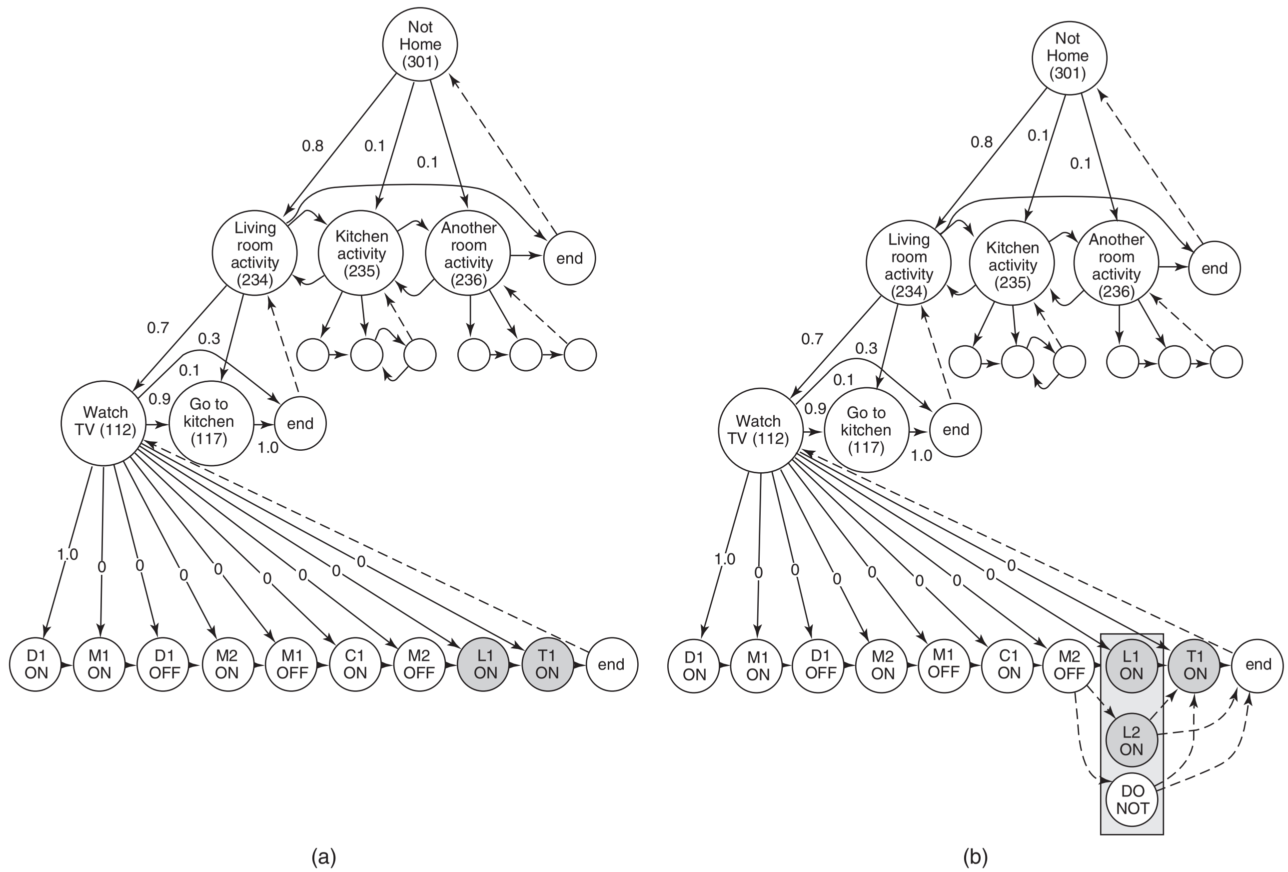


FIGURE 43.5 *Hierarchical model constructed from static (a) and dynamic (b) smart home data.*

Before an action is executed it is checked against the policies in the policy engine, ARBITER. These policies contain designed safety and security knowledge and inhabitant standing rules. Through the policy engine the system is prevented from engaging in erroneous actions that may perform actions such as turning the heater to 120°F or from violating the inhabitant's stated wishes (e.g., a standing rule to never turn off the inhabitant's night light).

43.4 INITIAL CASE STUDY

As an illustration of the techniques described above, we have evaluated a week in an inhabitant's life with the goal of reducing the manual interactions in the MavLab. The data were generated from a virtual inhabitant based on captured data from the MavLab and were restricted to motion and lighting interactions, which account for an average of 1400 events per day.

ALZ processed the data and converged to 99.99% accuracy after 10 iterations through the training data. When automation decisions were made using ALZ alone, interactions were reduced by 9.7% on average. Next, ED processed the data. Figure 43.6 shows the four-tier HPOMDP that is automatically constructed from the ED patterns. Because of space limitations, only the nodes at the higher levels of the model are shown. ED found eight interesting episodes with actions that could be automated, and further abstracted these to two metatasks. Livingroom patterns consisted of lab entry and exit patterns with light interactions, and the office also reflected entry and exit patterns. The other patterns occurred over the remaining eight areas and usually involved light interactions at desks and some equipment upkeep activity patterns. As a point of comparison, we automated the environment using a hierarchical Markov model with no abstract nodes. This flat model reduced interactions by 38.3%, and the combined learning system (the hierarchical ProPHeT-generated model bootstrapped using ED and ALZ) was able to reduce interactions by 76%, as shown in Figure 43.7a.

Experimentation in the MavPad using real inhabitant data has yielded similar results. In this case, ALZ alone reduced interactions from 18 to 17 events, the HPOMDP with

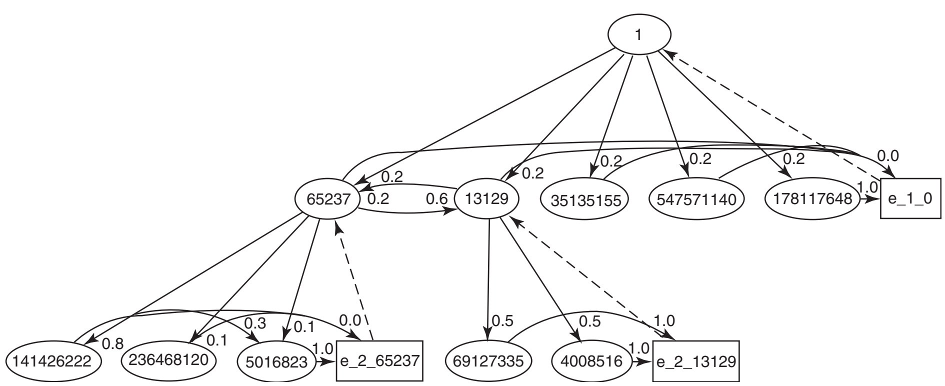
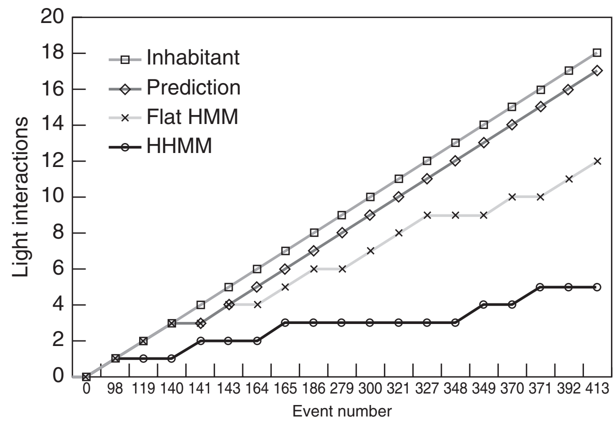
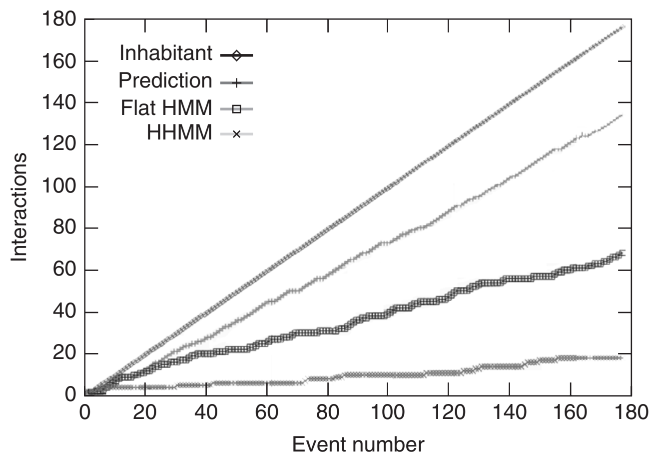


FIGURE 43.6 *ProPHeT-generated hierarchical POMDP (only the higher levels of the model are shown). There are eight abstract tasks found in the first iteration of ED and two metatasks (nodes 65237 and 13129) found in the second iteration. Boxes represent end nodes for each task sequence.*



(a)



(b)

FIGURE 43.7 *Interaction reduction.*

no abstract nodes reduced interactions by 33.3% to 12 events, while the bootstrapped HPOMDP reduced interactions by 72.2% to 5 events. These results are graphed in Figure 43.7b.

43.5 USING A SMART HOME TO ASSIST THE ELDERLY AND PEOPLE WITH DISABILITIES

The data-mining, prediction, and multiagent technologies available in MavHome can be employed to provide healthcare assistance in living environments. Specifically, models can be constructed of inhabitant activities and used to learn activity trends, detect anomalies, intelligently predict possible problems and make healthcare decisions, and provide automation assistance for inhabitants with special needs.

A variety of approaches have been investigated to automate caregiver services. Many of the efforts offer supporting technologies in specialized areas, such as using computer vision techniques to track inhabitants through the environment and specialized sensors to detect falls or other crises. Some special-purpose prediction algorithms have been implemented using factors such as measurement of stand–sit and sit–stand transitions and medical history [36–38], but are limited in terms of what they predict and how they use the results. Remote monitoring systems have been designed with the common motivation that learning and predicting inhabitant activities is key for health monitoring, but very little work has combined the remote monitoring capabilities with prediction for the purpose of health monitoring. Some work has also progressed toward using typical behavior patterns to provide reminders, which is particularly useful for the elderly and patients suffering from various types of dementia [39,40].

Our smart environment can identify patterns indicating or predicting a change in health status and can provide inhabitants with needed automation assistance. Collected data include movement patterns of the individual, periodic vital signs (blood pressure, pulse, body temperature), water and device usage, use of food items in the kitchen, exercise regimen, medicine intake (prescribed and actual), and sleep patterns [10,41]. Given these data, models can be constructed of inhabitant activities and used to learn lifestyle trends, detect anomalies, and provide reminder and automation assistance.

43.5.1 Capability 1: Identify Lifestyle Trends

Many of the smart space algorithms can provide particular benefit to individuals with particular health needs who are living independently. The first such benefit is to process the captured data in order to identify lifestyle trends that may highlight a growing need for the individual.

As a motivating example, consider a scenario involving an elderly man recuperating at home alone after hospitalization. The patient's son lives several hundred miles away but wants to be informed of his father's state of health. If the patient is a smart space inhabitant, he can be regularly monitored for changes in health measurements, including heart rate, blood pressure, and body temperature. However, these data may not provide a complete picture of his health status. As such, the data need to be integrated with information on changes in other parameters such as the room temperature and humdity and the individual's movement around the house, eating patterns, medicine intake, and adherence to his daily routine. The smart environment algorithms learn the inhabitant behaviors and start reporting timely information about changes in his health. A few weeks later the son notices in a system report that his father has a sudden decrease in his movements around the house. He calls his father and finds out that in fact his father has not been feeling well the last few days.

A variety of approaches have been investigated to automate caregiver services. Many of the efforts offer supporting technologies in specialized areas, such as using computer vision techniques to track inhabitants through the environment and specialized sensors to detect falls or other crises. Some special-purpose prediction algorithms have been implemented using factors such as measurement of stand–sit and sit–stand transitions and medical history [36–38,42,43], but are limited in terms of what they predict and how they apply the results. Remote monitoring systems have been designed with the common motivation that learning and predicting inhabitant activities is key for health monitoring, but very little work has combined the remote monitoring capabilities with prediction for the purpose of health monitoring. Some work has also progressed toward using typical behavior patterns to provide emergency notifiers or inhabitant reminders, which is particularly useful for the elderly and patients suffering from various types of dementia [39,40,44–46].

In the MavHome project, collected data can be analyzed not only to provide automation but also to assess trends. In particular, our algorithms currently classify slow changes in collected data as one of a number of types of pattern *drifts*: cyclic, increasing, decreasing, chaotic, and stable. The size of sample windows is chosen in such a way that it is approximately 4 times to length of the longest detectable cycle and twice the length of other trend classes.

Tests for various classes of drifts are performed using temporal autocorrelation plots, which measure the correlation between timeshifted values in a time series. The test for a stable pattern is performed first. This describes data that are nearly constant (within a tolerance threshold) for the entire window of data. A cyclic trend, which is checked next, shows high upward peaks in the autocorrelation graph because correlation between cylic values is high. In Figure 43.8a, frequencies of an action are shown and the corresponding autocorrelation plot (Figure 43.8b shows upward-facing peaks at intervals of seven. This indicates that the length of the cycle is seven.

For increasing or decreasing trends, a high degree of autocorrelation is seen between adjacent and near-adjacent observations. For this type of drift, the autocorrelation plot
Q3 will show a high correlation at lag 1 and will steadily decrease as the lag increases. The direction of the change can be determined by calculating the sum of the deviation in the adjacent data points. Any pattern in the sample window that is not classified as another type of drift is classified as choatic. This type of drift may be caused by a large number of irregular changes, by a change in the type of drift, or by noise in the data.

Pattern drifts are reported by MavHome if their urgency is high. Urgency is calculated as a combination of the confidence in the drift and the criticality of the analyzed data (drifts involving blood pressure are more critical than those based on changes in television-watching schedules).

We analyzed seven weeks of MavPad inhabitant data for drifts and made the following observations. For most of the collected activity data, patterns were classified as stable or chaotic. Increasing and decreasing trends were detected at points based on motion detector data, which is due to the increased (or decreased) amount of time that the inhabitant is spending at home. An increased amount of time that the light was on was also observed, possibly because of longer night hours as the days grew shorter. Cyclic

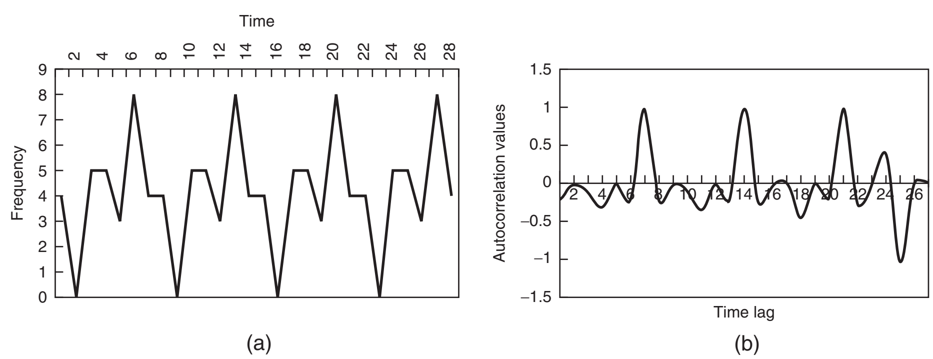


FIGURE 43.8 *An example of cyclic data (a) and the corresponding autocorrelation plot (b).*

drifts were the rarest. Although two cycles were detected, they only involved the use of lights and both were assigned a low confidence and a low criticality. In the case of health data, a decreasing trend was found throughout much of the collected time window. The inhabitant in this scenario is young and fairly healthy. We would expect different results when monitoring an elderly individual at home.

43.5.2 Capability 2: Detect Anomalies in Current Data

MavHome employs two techniques to detect outliers or anomalies in activity and health data. For the first method, we define an outlier as an extremely high or low value when compared to the rest of the data in the sample window. We use a z score, or standard score, to detect such outliers. This check is performed before looking for possible drifts.

The second approach makes use of the active-LeZi (ALZ) algorithm. ALZ predicts the expected next action of the inhabitant. As a side effect of the process, the algorithm generates a probability distribution over possible next events. If the probability of the observed event is greatly different from probabilities for alternative events, then the observed event (health data or observed data) is flagged as an anomaly.

In the case of the MavPad inhabitant, outliers were detected on day 31 for three different actions. As the graph in Figure 43.9 shows, the inhabitant's systolic value is zero in this day and the corresponding graph correlation is 1.5, which is identified as an outlier. We also see that the systolic values slowly decrease between days 10 and 23, which was identified as a decreasing drift of 11 days in length. The detected outlier is most likely due to an error in measurement, as the inhabitant was healthy on that day.

As with detected drifts, anomalies of a high criticality are identified for reporting. When a critical anomaly occurs, the home will first try to contact the inhabitant (through the interactive display for a lesser critical anomaly, or through the sound system for a more critical anomaly). If the inhabitant does not respond and the criticality of the anomaly is high, the caregiver will be made aware of the situation.

43.5.3 Capability 3: Design Reminder Assistance System

Reminders can be triggered by two situations: (1) if the inhabitant queries the home for her/his next routine activity, the activity with the highest probability will be given according to the ALZ prediction; and (2) if a critical anomaly is detected, the environment

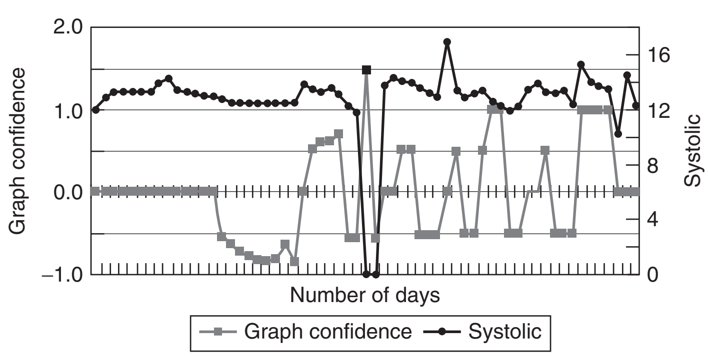


FIGURE 43.9 Plot of graph confidence and systolic values versus number of days for MavPad data.

will initiate contact with the inhabitant and remind her/him of the next typical activity. Such a reminder service will be particularly beneficial for individuals suffering from dementia.

As described in the initial MavHome design, automation assistance is always available for inhabitants, which is beneficial if some activities are difficult to perform. A useful feature of the architecture is that safety constraints are embedded in the ARBITER rule engine. If the inhabitant or the environment is about to conflict with these constraints, a preventative action is taken and the inhabitant notified. This can prevent accidents such as forgetting to turn off the water in the bathtub or leaving the house with doors unlocked.

43.6 CONCLUSION

We have demonstrated that the MavHome software architecture can successfully monitor and provide automation assistance for volunteers living in the MavPad site. However, there is much work to be done to enhance and test the benefits of the smart space algorithms for assisting the elderly and people with disabilities. We are currently collecting health-specific data in the MavHome sites and will be testing in the living environments of recruited residents at the C. C. Young Retirement Community in Dallas, Texas.

ACKNOWLEDGMENTS

This work is supported by US National Science Foundation under ITR grant IIS−0121297.

REFERENCES

1. Das SK, Cook DJ, Bhattacharya A, Heierman EO, Lin. T-Y: The role of prediction algorithms in the mavhome smart home architecture, *IEEE Wireless Commun*, **9**(6):77–84 (2002).
2. Lanspery S, Miller JR, Hyde J: Introduction: Staying put, in Lanspery S, Hyde J, eds, *Staying Put: Adapting the Places Instead of the People*, Baywood Publishing Company, 1997, pp 1–22.
3. AARP: *Fixing to Stay: A National Survey of Housing and Home Modification Issues*, 2000.
4. Zola IK: Living at home: The convergence of aging and disability, in Lanspery S, Hyde J, eds, *Staying Put: Adapting the places Instead of the People*, Baywood Publishing Company, 1997, pp 25–40.
5. Pynoos J: Neglected areas in gerontology: Housing adaptation, *Panel Presentation at the Annual Scientific Meeting of the Gerontological Society of America*, 2002.
6. Parr R, Russell S: Reinforcement learning with hierarchies of machines, in *Advances in Neural Information Processing Systems 10*, MIT Press, Denver, Dec 1997.
7. AARP: *These Four Walls . . . Americans 45+ Talk about Home and Community*, 2003.
8. Grayons PJ: Technology and home adaptation, in Lanspery S, Hyde J, eds, *Staying Put: Adapting the Places Instead of the People*, Baywood Publishing Company, 1997, pp 55–74.
9. Q4 Cook DJ, Das. SK: MavHome: Work in progress, *IEEE Pervasive Comput* (2004).
10. Das SK, Cook DJ: Health monitoring in an agent-based smart home by activity predition, in Zhang D, Mokhari M, eds, *Toward a Human-Friendly Assistive Environment*, IOS Press, 2004, pp 3–14.

11. AIRE Group: *MIT Project AIRE—About Us*, Jan 2004 (http://www.ai.mit.edu/projects/aire).
12. Fox A, Johanson B, Hanrahan P, Winograd T: Integrating information appliances into an interactive space, *IEEE Comput Graph Appl* **20**(3):54–65 (2000).
13. Romn M, Hess CK, Cerqueira R, Ranganathan A, Campbell RH, Nahrstedt K: Gaia: A middleware infrastructure to enable active spaces, *IEEE Pervasive Comput* 74–83 (2002).
Q5 14. Streitz NA, Geiler J, Holmer T, Konomi S, Mller-Tomfelde C, Reischl W, Rexroth P, Seitz P, Steinmetz R: i-LAND: An interactive landscape for creativity and innovation, *CHI*, 1999.
15. Abowd GD, Mynatt ED: Designing for the human experience in smart environments, in Cook DJ, Das SK, eds, *Smart Environments: Technology, Protocols, and Applications*, Wiley, Hoboken, NJ, 2005, pp 153–174.
16. Helal A, Mann W, El-Zabadani H, King J, Kaddoura Y, Jansen E: The gator tech smart house: A programmable pervasive space, *IEEE Comput* **38**(3):50–60 (2005).
17. NIST: *Smart Space NIST Laboratory* (http://www.nist.gov/smartspace/).
18. Mozer MC: Lessons from an adaptive home, in Cook DJ, Das SK, eds, *Smart Environments: Technology, Protocols, and Applications*, Wiley, Hoboken, NJ, 2005, pp 273–298.
19. Hagras H, Callaghan V, Colley M, Clarke G, Pounds-Cornish A, Duman H: Creating an ambient-intelligence environment using embedded agents, *IEEE Intell Syst* **19**(6) (2004).
20. Das SK, Cook DJ: Smart home environments: A paradigm based on learning and prediction, in *Wireless Mobile and Sensor Networks*, Wiley, Hoboken, NJ, 2005.
21. Youngblood GM, Holder LB, Cook DJ: A learning architecture for automating the intelligent
Q6 environment, *Proc Conf Innovative Applications of Artificial Intelligence*, 2005.
22. Youngblood GM, Holder LB, Cook DJ: Managing adaptive versatile environments, *Proc International Conf Pervasive Computing*, 2005.
23. Agrawal R, Srikant R: Mining sequential patterns, *Proc 11th Int Conf Data Engineering*, 1995, pp 3–14.
24. Heierman EO, Cook DJ: Improving home automation by discovering regularly occurring device usage patterns, *Proc Int Conf Data Mining*, 2003.
25. Rissanen J: *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing, Singapore, 1989.
26. Ziv J, Lempel A: Compression of individual sequences via variable rate coding, *IEEE Trans Inform Theory* **IT-24**:530–536 (1978).
27. Cielniak G, Bennewitz M, Burgard W: Where is ... ? Learning and utilizing motion patterns of persons with mobile robots, *Proc 18th Int Joint Conf Artificial Intelligence*, 2003, pp 909–914.
28. Philipose M, Fishkin K, Perkowitz M, Patterson D, Fox D, Kautz H, Hahnel D: Inferring activities from interactions with objects, *Pervasive Comput* **3**(4):50–56 (2004).
29. Bell TC, Cleary JG, Witten IH: *Text Compression*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
30. Gopalratnam K, Cook DJ: Active LeZi: An incremental parsing algorithm for sequential prediction, *Int J Artif Intell Tools* **14** (1–2):917–930, (2004).
31. Gopalratnam K, Cook DJ: Online sequential prediction via incremental parsing: The Active
Q7 LeZi algorithm, *IEEE Intell Syst* (2005).
32. Pineau J, Roy N, Thrun S: A hierarchical approach to POMDP planning and execution, *Workshop on Hierarchy and Memory in Reinforcement Learning (ICML)*, 2001.
33. Theocharous G, Rohanimanesh K, Mahadevan S: Learning hierarchical partially observable markov decision processes for robot navigation, *IEEE Conf Robotics and Automation*, 2001.
34. Precup D, Sutton RS: Multi-time models for temporally abstract planning, *Adv Neural Inform Process Syst* **10**:1050–1056 (1997).

35. Sutton RS, Barto AG: *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
36. Cameron K, Hughes K, Doughty K: Reducing fall incidence in community elders by telecare using predictive systems, *Proc Int IEEE-EMBS Conf*, 1997, pp 1036–1039.
37. Najafi B, Aminian K, Loew F, Blanc Y, Robert P: Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly, *IEEE Trans Biomed Eng* **49**(8):843–851 (2002).
38. Najafi B, Aminian K, Paraschiv-Ionescu A, Loew F, Bula C, Robert P: Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly, *IEEE Trans Biomed Eng* **50**(6):711–723 (2003).
39. Kautz H, Arnstein L, Borriello G, Etzioni O, Fox D: An overview of the assisted cognition project, *Proc AAAI Workshop on Automation as Caregiver*, 2002.
40. Pollack ME, Brown L, Colbry D, McCarthy CE, Orosz C, Peintner B, Ramakrishnan S, Tsamardinos I: Autoreminder: An intelligent cognitive orthotic system for people with memory impairment, *Robot Auton Syst* **44**:273–282 (2003).
41. Das SK, Cook DJ: Health monitoring in an agent-based smart home, *Proc Int Conf Smart Homes and Health Telematics (ICOST)*, 2004.
42. Chen D, Wactlar H, Yang J: A study of detecting social interaction in a nursing home environment, *Proc Int Conf Computer Vision Workshop on Human-Computer Interaction*, 2005.
43. Glascock AP, Kutzik DM: Behavioral telemedicine: A new approach to the continuous non-intrusive monitoring of activities of daily living, *Telemed J* **6**(1):33–44 (2000).
44. Chan M, Hariton C, Ringeard P, Campo E: Smart house automation system for the elderly and the disabled, *Proc IEEE Int Conf Systems, Man and Cybernetics*, 1995, pp 1586–1589.
45. Inada H, Horio H, Sekita Y, Isikawa K, Yoshida K: A study on a home care support information system, *Proc 7th World Congress on Medical Informatics*, 1992, pp 349–353.
46. Lawton MP: Aging and performance of home tasks, *Human Factor* **32**:527–536 (1990).

Queries in Chapter 43

Q1. OK? This was not a sentence no verb.

Q2. Defined earlier, on p 1303 (beginning of Sec. 43.1?

Q3. lag 1 (First lag) correct?

Q4. Please give vol # (also pp, if known)?

Q5. Incomplete ref—is this a conf paper or a journal article?

Q6. This must have been published by now; please try to give page #s?

Q7. Please give vol # (20?) and pp if known?