Artificial Intelligence on the Body, in the Home, and Beyond

Diane J. Cook
Washington State University
EME 121 Spokane Street
Pullman, WA 99164-2752, USA
+1 509-335-4985
cook@eecs.wsu.edu

ABSTRACT
The last few years have seen remarkable advances in the fields of body area networks and pervasive computing. These technologies generate large volumes of data that need to be processed, reasoned about, and acted upon. In this paper, we review the role that artificial intelligence plays in meeting this need and provide an overview of AI research projects that are making use or enhancing these technologies.

Keywords
Artificial intelligence, data mining, user modeling, smart environments

1. INTRODUCTION
Sensors pervade our high-tech world – they link available computational power with physical applications. Because of recent advancements in fields such as body area networks and smart environments, sensors are rapidly catching up with computing devices in popularity and widespread use. As they become more varied and easy to use, the need to analyze and act upon sensor data grows.

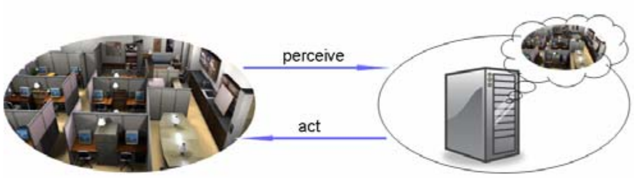

Figure 1. Intelligent agent controlling a smart environment.

Artificial Intelligence plays a key role in reasoning about sensor data. Researchers are striving toward the goal of creating an intelligent agent [1]. An intelligent agent, such as the one illustrated in Figure 1, perceives the state of a body or environment using sensors, reasons about the data using techniques including data mining and machine learning, and acts upon the environment using controllers in such a way that the agent achieves its intended goal.

BodyNets 2008, March 13-15 Tempe, Arizona, USA

DOI 10.4108/ICST.BODYNETS2008.2970

In this paper, we will take a closer look at the AI technologies that are used to process, reason about, and act upon about sensor data. We will also describe these technologies in the context of our smart home research project.

2. SENSING
Because intelligent agents are designed for real-world, physical applications, effective use of sensors is vital. Without physical components that allow an intelligent agent to sense and act upon the environment, we end up with theoretical algorithms that have no practical use.

Intelligent agents rely on sensory data from the real world. As Figure 1 shows, the software algorithm perceives the environment and uses this information to reason about the environment and the action that can be taken to change the state of the environment. Perception is accomplished using a variety of sensors. Sensors on the human body can provide information about physiological signals [1], body position [6], and body movement [5]. Sensors on assistive devices provide information about the device movement patterns [22] and the location of nearby objects [12]. Sensors in a smart environment can track the location of residents [11] and movable objects [1] as well as provide information about the physical environment such as light levels, temperature, and humidity.

Making sense of sensor data is a complex task. Sensor data comes with unique features that challenge conventional data analysis techniques. They generate large volumes of multidimensional data, defying attempts to manually analyze it. If the sensors are imprecise the data can be noisy, and if a sensor fails there may be missing values. Sensor data often needs to be handled on the fly or as streaming data [14], and the data may have a spatial or temporal component to it.

When faced with large amounts of raw sensor data, AI techniques can assist by first searching for patterns in the data that will help characterize the nature of the data and identify underlying models. Hierarchical clustering partitions the data into sets of similar data points, self-organizing maps visualize the data by moving similar data points close together in the map, and association analysis finds repetitive sequences in time-ordered data. A common data analysis goal is to map data points to predefined class labels, which in essence classifies the data. For example, an individual in a smart home can be identified by mapping a sequence of sensor events onto potential resident names. Many types of classification algorithms, or supervised learning algorithms, can be applied to sensor data, including decision trees, neural networks, Bayesian classifiers, instance-based learners, regression algorithms, and support vector machines.

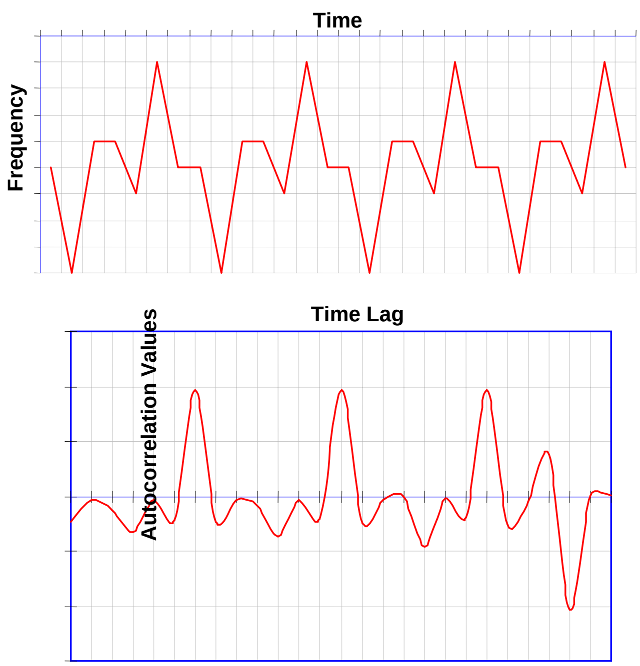

Figure 2. An example of cyclic data (top) and the corresponding autocorrelation plot (bottom).

Sensor data that has a time component can be further analyzed to determine trends. Trend analysis techniques often use temporal autocorrelation plots as shown in Figure 2. Such plots show the correlation between time-shifted values in a time series and show whether values along a particular dimension of the data are fairly stable (values are fairly constant), steadily increasing, decreasing, or exhibiting cyclic patterns. A special class of trend analysis is anomaly detection or outlier detection, which can be performed using statistical tests such as the Grubbs test.

3. REASONING

Sensing and acting provide links between intelligent algorithms and the real world in which they operate. In order to make such algorithms responsive, adaptive, and beneficial to users, a number of types of reasoning must take place. For example, one feature that separates general computing algorithms from those that are responsive to the user is the ability to model user behavior. If such a model can be built, it can be employed to customize the behavior of the software the user. If the model results in an accurate enough baseline, it can provide a basis for detecting anomalies and changes in user behavior. If the model has the ability to be automatically refined, the software can adapt itself to these changing patterns. As an example, Loke [13] builds a model from sensor data of smart environment resident actions and device states, and pulls information from similar situations to provide a context-aware environment. Doctor, et al. [7] model resident behavior by learning fuzzy rules that map sensor states to actuator readings representing resident actions.

A second contribution that AI reasoning algorithms offer is the ability to predict and recognize events and higher-level activities. The Neural Network House, the Intelligent Home, the House_n, and the MavHome smart environment projects adaptively control home environments by anticipating the location, routes, and activities of the residents. Prediction algorithms have been developed for both the single [8] and the multiple resident [18] cases. Predicting events allows the intelligent agent to anticipate a user's needs and assist with (or possibly automate) the event. If prelabeled user activity is available, then supervised learning approaches can be used to build a model of activities and these models can be employed to recognize current user tasks [16][21].

Very little work can be done on body area networks or on smart environments without an explicit or implicit reference to where and when the data was collected and the meaningful events occurred. For a system to make sensible decisions it has to be aware of where the users are and have been during some period of time. Spatial and temporal reasoning are two well-established areas of AI. Gottfried, et al. [9] has shown how traditional spatial and temporal reasoning frameworks can be enhanced to yield a better understanding of the activities in pervasive computing applications. Jakkula, et al. [10] uses an existing temporal formalism to describe frequent temporal relationships between events in a smart home, while Augusto and Nugent [3] describe a new language which allows the specification of situations involving repetitions, sequences, frequencies, and durations of activities.

4. ACTING

Intelligent agents tie reasoning to the real world through sensing and acting. Automated decision making and control techniques are useful in transitioning reasoning approaches to action selection and execution. Simpson, et al. discuss how AI planning systems can be employed in a smart environment to remind individuals of their next daily activity. Mozer's Adaptive Home uses a neural network and a reinforcement learner to determine and control ideal light and fan settings for the house. Another notable example in this area is the research of Amigoni, et al. who employs a Hierarchical Task Network planner to generate sequences of actions and contingency plans for a smart environment. The planner will, for example, respond to a sensed health need by calling a medical specialist and sending health vitals collected by body sensors using any available device (cell phone, email, or fax). If there is no response from the specialist, the planner would phone the nearest hospital and request ambulance assistance.

5. CASE STUDY: THE MAVHOME SMART HOME

The cycle of sensing, reasoning, and acting is the hallmark of intelligent agent applications. We describe here how these components fit together in the designing of the MavHome smart home application.

Since the beginning, people have lived in places that provide shelter and basic comfort and support. As society and technology advance there is a growing interest in improving the intelligent of the environments in which we live and work. The MavHome project is focused on providing such environments [23]. We take the viewpoint of treating an environment as an intelligent agent which perceives the state of the environment using sensors and acts upon the environment using device controllers in a way that can maximize the comfort and productivity of the residents, minimize the consumptions of resources, and maintain the safety and security of the environment and its residents.

The MavHome architecture shown in consists of cooperating layers. Perception is a bottom-up process. Sensors 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 decision making applications (top layer) upon 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.

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 in-house. Resident localization is performed using passive infrared sensors yielding a detection rate of 95% accuracy.

Communication between high-level components is performed using 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.

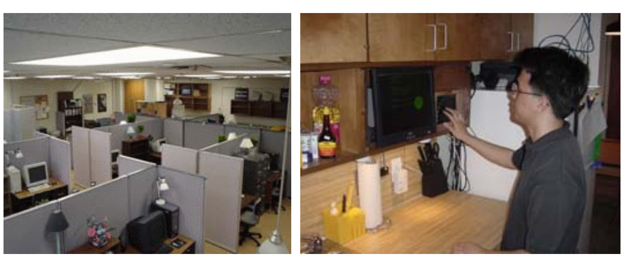

Figure 3. The MavLab (left) and MavPad (right) environments.

MavHome is designed to optimize a number of alternative functions, but for this case study we focus on minimization of manual interactions with devices. The MavHome components are fully implemented and have automated the environments shown in Figure 3. 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, CO detection, motion, and door/window/seat status sensors. Figure 4 shows the MavPad sensor layout.

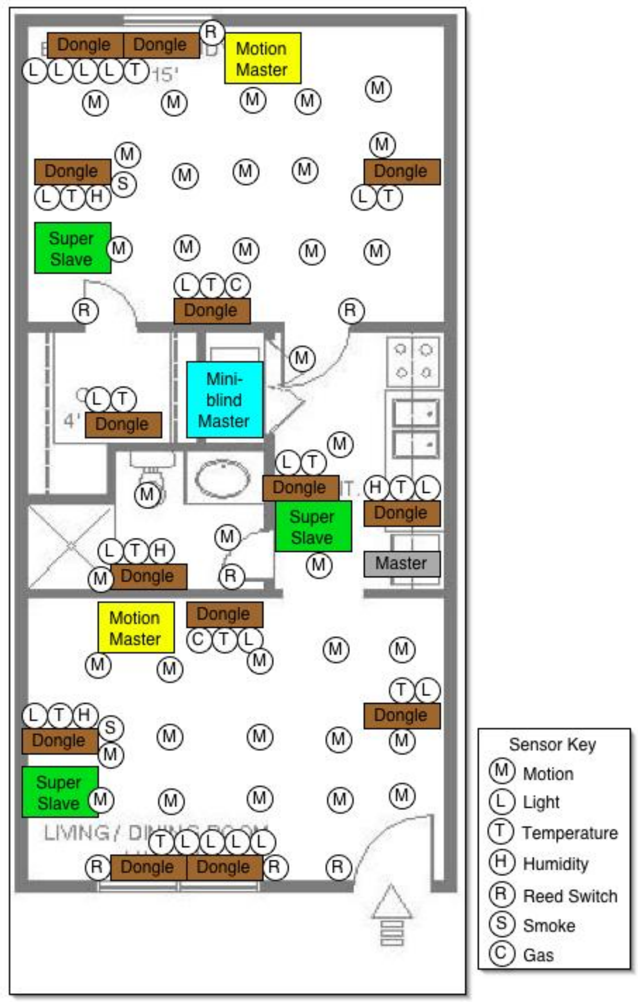

Figure 4. MavPad sensor layout.

To automate our smart environment, we collect observations of sensor events and manual resident interactions with the environment. We mine sequential patterns from this data using a sequence mining algorithm, ED. Next, we predict the resident's upcoming actions using our ALZ algorithm. Finally, a hierarchical Markov model is created using low-level state information and high-level sequential patterns, and is used by our ProPHeT algorithm to learn an action policy for the environment.

5.1. Episode Discovery

A smart home resident typically interacts with various devices and triggers sensors while conducting routine activities. These interactions may be considered as a sequence of events, with some inherent pattern of recurrence. We represent each sensor event as a triple consisting of the sensor fire (or device manipulated), the value of the sensor reading (or the new state of the device), and the time of the event. Our Episode Discovery (ED) algorithm [10] moves a window in a single pass through the history of events, looking for episodes (sequences) within the window that merit attention. Candidate episodes above a minimum acceptable value are reported.

When evaluating candidate episodes, ED looks for patterns that minimize the description length of the input stream, O, using the Minimum Description Length (MDL) principle. 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, θ) = DL(O| θ) + DL(θ), or the description length of the input sequence compressed using θ plus the description length of the patterns θ. ED reports episodes, or sequences, that yield the largest compression ratio (ratio of the original DL to the DL of the compressed sequence).

Our MDL-based evaluation measure identifies patterns that balance frequency and length. Periodicity (daily, every other day, weekly occurrence) of episodes is detected using autocorrelation and included in the episode description. In this way, ED identifies patterns of events that enable understand of resident activities. Once the data is compressed using discovered patterns, ED can be run again on the compressed data to generate a hierarchy of patterns within the event data.

5.2. Event Prediction

To predict events (sensor readings or resident interactions with the environment), we borrow ideas from LZ78 text compression [24]. By predicting events, a smart environment can automate or improve upon anticipated events in the environment. Well-investigated text compression methods have established that good compression algorithms also make good predictors.

LZ78 incrementally processes an input string of characters and stores them in a tree. In our case the string represents the history of sensor events and device interactions. The algorithm parses the string $x_1, x_2, ..., x_i$ into substrings $\omega_1, \omega_2, .., \omega_{c(i)}$ such that for all j>0, the prefix of the substring ω_j is equal to some ω_i for 1<i<j. Thus each newly-encountered substring is stored in the tree as an extension of a substring already in the tree.

Our Active Lezi (ALZ) algorithm [8] enhances LZ78 by recapturing information lost across phrase boundaries. Frequency of symbols is stored along with phrase information in the tree, and information from multiple context sizes is combined to provide the probability for each potential symbol (sensor event or manual interaction) 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.

To perform prediction, ALZ calculates the probability of each symbol occurring, and predicts the event 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.

We initially evaluated the ability of ALZ to perform resident action prediction on synthetic data based on six embedded tasks with 20% noise. In this case the predictive accuracy converges to 86%.

5.3. Decision Making

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 residents) 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 $R: S \rightarrow [0,1]$ 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 at executed in state s_t leads to state s_{t+1}.

With the increasing complexity of tasks being addressed, recent work in decision making under uncertainty has popularized the use of Hierarchical MDPs. While they are 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 [24], 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 this 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-tier hierarchical model is thus learned from collected data. An example hierarchical model constructed from MavHome data is shown in Figure 5.

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

To learn an automation strategy, ProPHeT employs a temporal-difference reinforcement learning strategy to form control policies which optimize the expected future reward. In particular, MavHome receives negative reinforcement when the resident immediately reverses an automation decision (e.g., turns the light back off) or an automation decision contracts safety or user-specified comfort constraints.

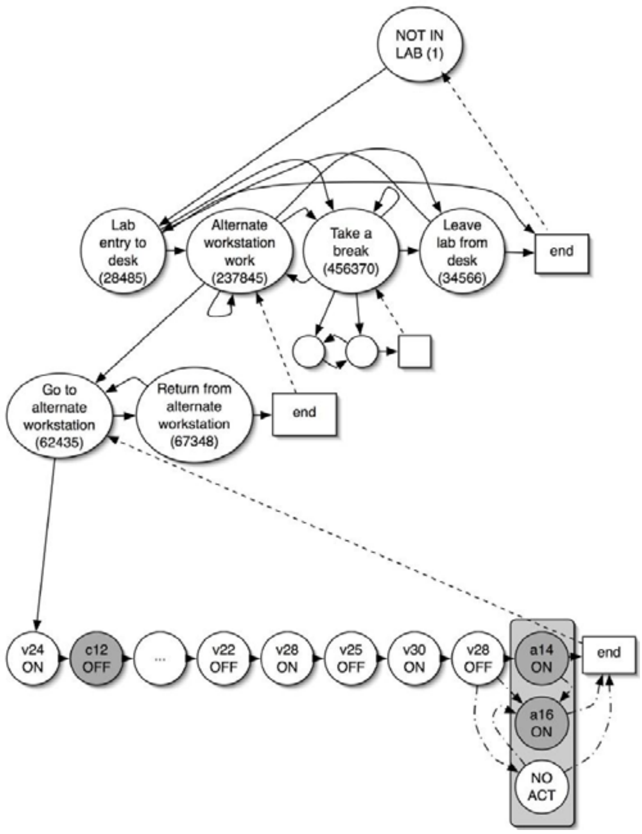

Figure 5. Hierarchical model constructed from MavLab data.

5.4. Evaluation

As a validation of the performance of the MavHome core algorithms, we evaluated a week in a resident's life with the goal of reducing the number of manual interactions with the MavLab environment. The data was restricted to just motion sensor data and lighting interactions which account for an average of 1400 events per day.

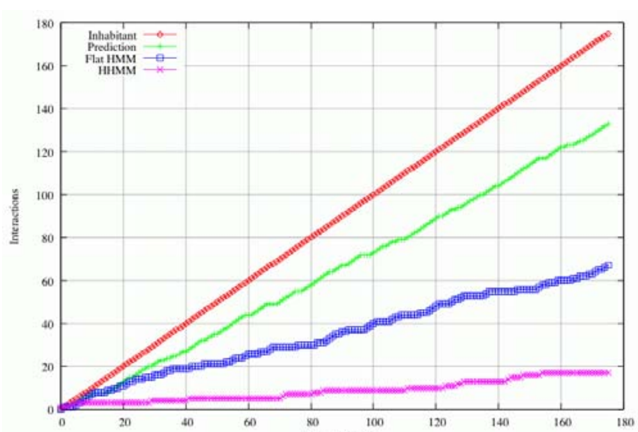

Figure 6. Interaction reduction in the MavLab.

When automation decisions were made using ALZ alone, interactions were reduced by 9.7% on average. ED next processed the data and ProPHeT used these results to automatically construct a model with eight interesting episodes and two meta-tasks. Automation using a flat model with no abstract nodes reduced interactions by 38.3%, and the hierarchical model was used to reduce interactions by 76%, as shown in Figure 6.

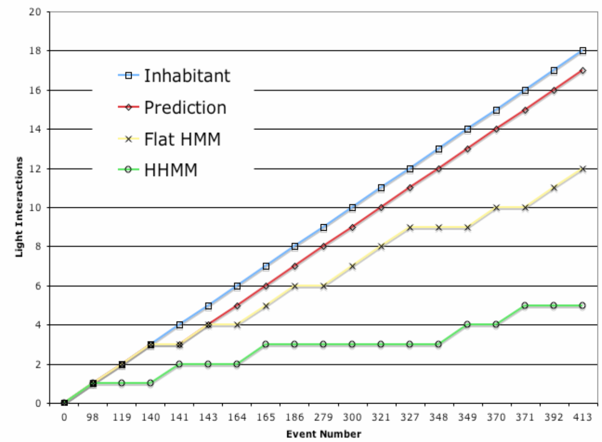

Figure 7. Interaction reduction in the MavPad.

Experimentation in the MavPad using real resident data (with an average of 18 manual interactions per day) yielded similar results. In this case, ALZ alone reduced interactions by 1 event, the flat model reduced interactions by 33.3%, and the hierarchical model reduced interactions by 72.2% to 5 events. The results are shown in Figure 7.

6. CONCLUSIONS

Both body area networks and smart environments are establishing fast as areas where a confluence of topics can converge to help society through technology. In this paper we summarized the beneficial role that Artificial Intelligence can play in making these algorithms robust and adaptive to the user. We have also illustrated the benefits of AI algorithms in the MavHome smart environment project.

There are still many challenges to face in the development of Artificial Intelligence algorithms for use in these fields. For example, attributing events to individuals in a multi-resident setting is an ongoing challenge. We would also like to see the notion of "environment" extend from a single setting to encompass all of an individual's spheres of influence by fusing information from these multiple settings. Similarly, researchers can share data between multiple body area networks and between multiple environments to benefit the group of individuals as a whole, and in the case of smart environments, actually create "smart communities". Finally, the concerns about creating intelligent and adaptive algorithms while maintaining privacy needs to be addressed in ongoing research efforts.

7. ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation grant IIS-0121297.

8. REFERENCES

[1] Abowd, G.D. and Mynatt, E.D. Designing for the human experience in smart environments. In *Smart Environments: Technology, Protocols, and Applications* (D. Cook and S. Das, eds). Wiley, 2004. 153-174.

[2] Amigoni, F., Gatti, N., Pinciroli, C., and Roveri, M. What planner for ambient intelligence applications? *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35, 1 (2005). 7-21.

[3] Augusto, J. and Nugent, C. The use of temporal reasoning and management of complex events in smart homes. In *Proceedings of the European Conference on Artificial Intelligence* (2004). 22-27.

[4] Bao, S., Zhang, Y., and Shen, L. Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems. In *Proceedings of the IEEE Conference on Engineering in Medicine and Biology* (Shanghai, China, 2005). 2455-2458.

[5] Binkley, P. Predicting the potential of wearable technology. *IEEE Engineering in Medicine and Biology*, 22, 3 (May/June 2003), 23-24.

[6] Biswas, S., and Quwaider, M. Body posture identification using hidden Markov model with wearable sensor networks. In *Proceedings of the International Conference on Body Area Networks* (Tempe, Arizona, 2008).

[7] Doctor, F., Hagras, H., and Callaghan, V. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35, 1 (2005). 55-65.

[8] Gopalratnam, K., and Cook, D.J. Online sequential prediction via incremental prasing: The Active LeZi algorithm. IEEE Intelligent Systems, 22, 2 (2007).

[9] Gottfried, B., Guesgen, H., and Huebner, S. Spatiotemporal reasoning for smart homes. In *Designing Smart Homes – The Role of Artificial Intelligence*. Springer, 2006.

[10] Heierman, E. and Cook, D.J. Improving home automation by discovering regularly occurring device usage patterns. In Proceedings of the International Conference on Data Mining (2003). 537-540.

[11] Jakkula, V., Crandall, and Cook, D.J. Knowledge discovery in entity-based smart environment resident data using temporal relations based data mining. In *Proceedings of the ICDM Workshop on Spatial and Spatio-Temporal Data Mining* (2007).

[12] Kurschl, W., Gottesheim, W., Mitsch, S., Prokop, R., Schönböck, and Beer, W. A two layered deployment schema for wireless sensor network based location tracking. In *Proceedings of the International Conference on Information Technology* (Las Vegas, NV, 2008).

[13] Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y., and Borenstein, J. The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7, 4 (December 1999). 443-451.

[14] Loke, S.W. Representing and reasoning with situations for context-aware pervasive computing: A logic programming perspective. The Knowledge Engineering Review, 19, 3 (2005). 213-233.

[15] Madden, S. and Franklin, M.J. Fjording the stream: An architecture for queries over streaming sensor data. In Proceedings of the International Conference on Data Engineering (2002). 555-566.

[16] Mozer, M.C. Lessons from an adaptive home. In *Smart Environments: Technology, Protocols, and Applications* (D. Cook and S. Das, eds.). Wiley, 2004. 273-298.

[17] Muehlenbrock, M., Brdiczka, O., Snowdon, D., and Meunier, J. Learning to detect user activity and availability from a variety of sensor data. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications* (2004).

[18] Rissanen, J. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[19] Roy, N., Roy, A., and Das, S.K. Context-aware resource management in multi-inhabitant smart homes: A framework based on Nash *h*-learning. *Journal of Pervasive and Mobile Computing*, 2, 4 (2006). 372-404.

[20] Russell, S.J. and Norvig, P. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2003.

[21] Simpson, R., Schreckenghost, D., LoPresti, E.F., and Kirsch, N. Plans and planning in smart homes. In *Designing Smart Homes: The Role of Artificial Intelligence*. Springer Verlag, 2006.

[22] Tapia, E.M., Intille, S.S., and Larson, K. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of Pervasive* (2004). 158-175.

[23] Vahdatpour, A., Sarrafzadeh, M., Wu, W., Au, L., Jordan, B., Stathopoulos, T., Batalin, M., Kaiser, W., Fang, M., and Chodosh, J. The SmartCane system: An assistive device for geriatrics. In *Proceedings of the International Conference on Body Area Networks* (Tempe, Arizona, 2008).

[24] Youngblood, G.M. and Cook, D.J. Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37, 4 (2007). 1-12.

[25] Ziv, J. and Lempel, A. Compression of individual sequences via variable rate coding. IEEE Transactions on Information Theory, IT-24 (1978). 530-536.