IPSJ Journal Vol. 51 No. 4 1175–1191 (Apr. 2010)

Regular Paper

A Ubiquitous Power Management System to Balance Energy Savings and Response Time Based on Device-level Usage Prediction

Hua Si,[†1] Shunsuke Saruwatari,[†1]
Masateru Minami[†1] and Hiroyuki Morikawa[†1]

Power conservation has become a serious concern during people's daily life. Ubiquitous computing technologies clearly provide a potential way to help us realize a more environment-friendly lifestyle. In this paper, we propose a ubiquitous power management system called Gynapse, which uses multi-modal sensors to predict the exact usage of each device, and then switches their power modes based on predicted usage to maximize the total energy saving under the constraint of user required response time. We build a three-level Hierarchical Hidden Markov Model (HHMM) to represent and learn the device level usage patterns from multi-modal sensors. Based on the learned HHMM, we develop our predictive mechanism in Dynamic Bayesian Network (DBN) scheme to precisely predict the usage of each device, with user required response time under consideration. Based on the predicted usage, we follow a four-step process to balance the total energy saving and response time of devices by switching their power modes accordingly. Preliminary results demonstrate that Gynapse has the capability to reduce power consumption while keeping the response time within user's requirement, and provides a complementary approach to previous power management systems.

1. Introduction

Rising global energy demands, increasing costs and limitations on natural resources have raised concerns about energy conservation. As reported by Japanese government [1], the energy consumption in Residential & Commercial sector has increased 40% since 1990, among which the most consumed energy is electricity by various home electronic devices [2]. Ubiquitous computing technologies provide the potential to reduce power consumption. For instance, the lights, heating and other devices all sit on a ubiquitous in-house network. Lights can be automatically controlled by sensors that measure the brightness in the room, and switch off altogether if no one is in the room. The temperature of the living room air conditioning or floor heating changes according to the number of people in the room. We can also check if the house is carbon neutral by seeing how much power is generating from solar panels and how much is used. In this paper, we will discuss the power saving potential of ubiquitous computing technologies in an indoor environment.

People have a lot of methods to reduce power consumption in indoor environments. The most common way is replacing with energy efficient devices. However, replacement cannot solve all the problems. For instance, even if the resident replaces a 60 W bulb with a 30 W energy efficient one, it will still waste a lot of energy if he leaves the bulb ON for 24 h/7d. To eliminate such waste, people try an alternative way to switch lights off or turn devices into low power mode when not in use [3], which we call "power mode switching". This method has drawn special attention from researchers of ubiquitous computing, and shown promising results [4),5),10)–15]. These systems try to automatically switch power modes of devices with sensors and controllers installed in indoor environments. We follow their research and focus on "ubiquitous power mode switching" systems in this paper.

According to our research, we believe the following requirements are important for a practically effective power mode switching system:

- The system should have the capability to adaptively handle multiple devices for human behaviors in different situations. Because of the complexity of human behavior, a resident usually uses different combinations of devices in different situations. Therefore, it is natural to require a system to be able to adaptively coordinate the power mode switching of multiple devices for behaviors in different situations.
- The system should have the capability to precisely predict the exact usage of each device. The key of power mode switching is to proactively determine the future usage of a device, and switch its power mode accordingly. Hence, precise usage prediction is crucial for such a system. In addition, since we can only switch power mode of *each* device, the usage prediction should also

[†1] Research Center for Advanced Science and Technology, The University of Tokyo

be at *device level*. As a result, a system must be able to precisely predict the usage at device level.

- The system should have the capability to balance the response time of devices and their total energy saving. For instance, although a PC can save energy in sleep mode, the resident may feel frustration with its long wake-up time. Therefore, a system must be able to save energy while keeping the response time short enough.

Previous power mode switching systems [4),5),10)–15)] usually make power mode switching according to pre-defined "rules" or the resident's locations. Although they successfully meet one or two requirements above, none of them can fulfill all three requirements. In this paper, we propose a ubiquitous power management system called Gynapse, which uses multi-modal sensors to predict the exact usage of each device, and then switches their power modes based on predicted usage to maximize the total energy saving under the constraint of user required response time. To our best knowledge, it is the first system that fulfills all the requirements above. Gynapse consists of three important components:

- A probabilistic model to learn residents' usage patterns at device level from multi-modal sensors. We build a three-level Hierarchical Hidden Markov Model (HHMM)[6)] to represent multiple residents and their device usage, and use Forwards-Backwards (FB) and Expectation-Maximization (EM) algorithms[7)] to learn the parameters. The sensor data, such as RFID reading from keyboard and current data from power lines, are arranged as vectors to train HHMM. This model provides the capability to adaptively handle multiple devices for complicated behaviors in different situations.
- A predictive mechanism to forecast the usage probability of multiple devices in the future. Based on the learned three-level HHMM, we introduce two variables to represent device's wake-up time and user's required response time, and develop our predictive mechanism in Dynamic Bayesian Network (DBN)[8)] scheme. This mechanism provides the capability to precisely predict the usage of each device, and take response time into consideration.
- A control framework to maximize the energy saving under the constraint of user required response time. With the predicted usage probability of each device, we follow a four-step process to calculate the total probability and energy saving of devices, and switch their power modes according to the scenario with the highest energy saving and probability. This framework puts everything together and balances the response time of devices and their total energy saving.

To conduct our analysis, we use sensor data from MIT PlaceLab [9)] to implement and evaluate Gynapse. We obtain 23 days of data, within which we use 9 days to train the three-level HHMM, and 14 days to verify the system. After learning the parameters of our probabilistic model, the predictive mechanism correctly predicts the device usage for about 90%. Based on the usage prediction, the control framework successfully balances response time and energy saving, and achieves an average 11% power saving of 14 days.

The rest of this paper is organized as follows: Section 2 reviews the previous research on power mode switching systems. Section 3 analyzes the technical problems we are going to solve. Section 4 describes the design of Gynapse. In Section 5, we evaluate our system with PlaceLab data and discuss the preliminary results. Section 6 concludes the paper finally.

2. Related Works

In this section, we will review the previous power mode switching systems according to the three requirements mentioned in Section 1.

The most straightforward way of power mode switching is automatic power-off based on pre-defined "rules". Such systems use infrared or motion sensors to detect the existence/absence of residents, and then turn on/off the lights or air-conditioners through Home Energy Management System (HEMS)[4),10)]. Their motivation is the simplicity to build a rule, such as "*If* no user in the room *for* a time-out period, *then* turn off the lights." However, human behaviors are so complicated that cannot be completely described with such simple rules, e.g., if one sits *still* in a chair when reading, the lights may be incorrectly turned off after a motion time-out period. Additionally, as the number of rules increases, it may become difficult to coordinate rules in different situations. For instance, one rule may say "If no user sits in front of TV for a time-out period, then turn *off* TV"; whereas, another rule may say "If the user is cooking in kitchen, then turn *on* TV", because he usually watches it when cooking. To eliminate such conflicts, people have to define complex rules to describe the situations, which digress from the original standpoint of simplicity. As a result, such "rule-based switching" systems do not fulfill the first requirement, since they cannot properly handle multiple devices for complicated human behaviors in different situations.

To solve the problems of "rule-based switching" systems, researchers have tried probabilistic models for more efficient power management. The first work is Neural Network House 11),12), which builds a neural network to predict the mobility of residents between different "lighting zones", and switch the power setting of lights accordingly. A similar work is MavHome project 13), which predicts the resident's location together with his most likely path based on an information-theoretical framework, and pro-actively switches the power mode of devices along this future route. Harris and Cahill 14),15) introduce the concept of context-aware power management, and concentrate on switching power modes of a desktop PC according to predicted usage. They build a Dynamic Bayesian Network (DBN) for usage prediction based on user's proximity to PC. Although people argue that probabilistic prediction rarely achieve 100% accuracy, it can be improved by integrating high-level information, such as user feedback 16)–18).

A more serious drawback of previous probabilistic power mode switching systems is they are all based on the prediction of resident's *coarse-grained location* instead of *each device's exact usage*. Therefore, they leave two problems unsolved: 1) they imply the device location is fixed, so they can predict usage based on resident's proximity (location) to device. However, for devices such as TV, which are controlled via remote, proximity does not work. Furthermore, they cannot predict proximity to remote, because it is movable. 2) They only discuss the situation that *one* (kind of) device at one coarse-grained location, such as the light in a room. However, it is not unusual that a TV, a DVD player and a light are in the same room. In this case, previous works cannot decide which device to use, even if they can predict the resident will enter this room. Therefore, such "coarse-grained location-based switching" systems do not fulfill the second requirement, because they cannot predict the exact usage of each device.

People may argue that we can solve the preceding problems, if we have a highly fine-grained location system that pinpoints every device no matter movable or not. Unfortunately, no such system exists at current stage. However, researchers of activity recognition provide an alternative way to model the device level usage by using multi-modal sensors. In Refs. 19), 20), they attach RFID tags to devices such as TV remote and washer, and build probabilistic models such as Hidden Markov Model (HMM) or Dynamic Bayesian Network (DBN) to infer the activities in which the devices are used. References 9), 21) use more multi-modal sensors to recognize activities in addition to RFID tags. However, since the objective of these researches is to infer the activity given the observed sensor data, not *predict* the activity and device usage in the future, they do not fulfill the second requirement.

As we are aware, no existing power mode switching system is based on the usage prediction at *device level*. In this paper, we try to build such a system by using multi-modal sensors, since it is more realistic than a fine-grained location-based system. We must make it clear that our system is complementary rather than competitive to previous coarse-grained location-based systems. On one hand, we can use them jointly to increase accuracy. For example, we can predict the room to enter from location system, and predict the device to use from multi-modal sensors. On the other hand, we can use device level prediction independently. For instance, no matter a resident's location is in living room or kitchen, we can always capture his usage of TV from RFID tag attached to the remote. In this paper, we will learn and forecast the exact usage of each device from multi-modal sensors, and switch their power modes based on the usage prediction at device level.

The third requirement comes from a major frustration of previous power mode switching systems, which is the users have to wait for a device to wake up 5),22). To solve this problem, Harle and Hopper 5) classify the electronic devices into three "wake-up time" categories, and optimize their power mode switching based on a location-aware system in an office building. However, they have the same problem as other location-based power management systems, and cannot address the problem such as remote-controlled devices or multiple devices at the same location.

From the discussion above, we can find that no previous systems fulfill all three requirements. To complement previous research, we propose Gynapse, which uses multi-modal sensors to predict the exact usage of each device, and then switches their power modes based on predicted usage to maximize the total energy saving under the constraint of user required response time. In the following section, we will analyze the technical problems need to be solved by Gynapse.

3. Problem Statements

As we explained in Section 1, three components are important for Gynapse: 1) a probabilistic model to learn residents' usage patterns at device level, 2) a predictive mechanism to forecast the usage probability of multiple devices in the future, 3) a control framework to maximize energy saving under the constraint of user required response time. We will analyze the technical challenges of each component, and formulate the problems we are going to solve.

3.1 Challenges for Probabilistic Model

The main challenge for probabilistic model derives from the fact that hierarchical structure exists in human behaviors [23),24)]. For instance, during a high level activity such as "make breakfast", the resident may repeat low level actions[⋆1], such as "use coffee maker" and "use toaster"; or during the activity of "work in office", he may repeat actions of "use PC" and "use lamp". Hence, if the resident is "using coffee maker", it is more possible for him to "use toaster" rather than "use PC" for the next step. This kind of sequences (or transitions) of actions and activities are actually the "patterns" of resident's device usage. To learn them, probabilistic model must properly represent the hierarchical structure in human behaviors and the transitions between actions and activities.

Another challenge is multiple residents in one family. Although one-person families will become the largest single category in Japan by 2010 with a percentage of 31.2%, multi-person families will still be the mainstream[⋆2]. Therefore, probabilistic model need another hierarchy to represent the status of residents.

3.2 Challenges for Predictive Mechanism

After learning the usage patterns, we can predict the usage probability of multiple devices in the future. There are three challenges for the predictive mechanism. The first one is it must be able to handle different wake-up times. **Figure 1** depicts an example. Two devices have different wake-up times, $T_{WU,1}$ and $T_{WU,2}$, respectively. If both of them are awoken at time t, they can only be used after $t_1 = t + T_{WU,1}$ and $t_2 = t + T_{WU,2}$, respectively. Therefore, if a system wants to make wake-up decisions at time t, it has to predict the usage probability of device 1 at time t_1, and that of device 2 at time t_2, which is defined as wake-up probability of device i at time t, $p_{t,i}$.

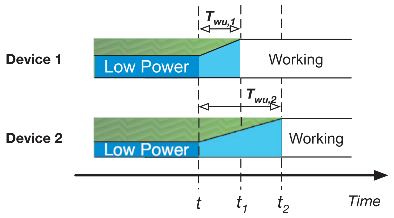


Fig. 1 Different wake-up times of devices.

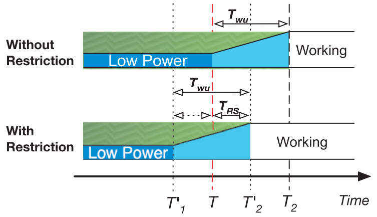


Fig. 2 Required response time.

The second challenge for predictive mechanism is it must consider an acceptable response time[⋆3] for the resident. An example is illustrated in **Fig. 2**. A service request comes at time T. If the device is awoken at that time, the resident cannot use it until $T_2 = T + T_{WU}$. However, if the resident restricts the response time *no more than* $T_{RS} = T_2' - T$, then the device must be awoken *at least* T_1', which is $(T_{WU} - T_{RS})$ before the service request at T[⋆4]. As a result, if we want to make

[⋆1] For clarity, we define resident's direct interaction with a device as a low level "action", and define his goal of a sequence of actions as a high level "activity".

[⋆2] As projected by the National Institute of Population and Social Security Research, because of the falling birthrate and the aging population, the family structure in Japan will have influential change by 2010: 31.2% will be single live, 20.1% will be only husband and wife, 27.9% for parents and kids, 9% for single-parent and kids, and 11.8% for others [25)].

[⋆3] Response time is defined as the duration between a service is requested and a device is ready to provide it, such as $T_2 - T$ or $T_2' - T$ in Fig. 2

[⋆4] The practical upper limit of T_{RS} is T_{WU}, since the user will not accept a device to response longer than its wake-up time.

a wake-up decision at $T_1^{'}$ like t in Fig. 1, the wake-up probability $p_{T_1^{'}}$ is actually the device usage probability at $T_1^{'} + (T_{WU} - T_{RS})$ instead of $T_1^{'} + T_{WU}$.

The third challenge is improving the accuracy of prediction. Because the predicting algorithm itself rarely achieve 100% accuracy, some high-level information should be integrated to improve it.

3.3 Challenges for Control Framework

After predicting the wake-up probability of devices, we can control power mode switching to maximize the total energy saving under the constraint of user required response time. As explained in Section 3.2, the user required response time, T_{RS}, has been considered when predicting wake-up probability $p_{T_1^{'}}$. If this probability is less than 0.5, which means the device should not be awoken at $T_1^{'}$, then the device has to postpone its wake-up, and the response time will exceed user's requirement T_{RS}. Therefore, the restriction that the response time should not exceed T_{RS} actually becomes that the wake-up probability $p_{T_1^{'}}$ should not fall below 0.5. As a result, the challenge for control framework becomes choosing a combination of devices that will save maximum energy after power mode switching, under the constraint that the wake-up probability of this combination is higher than a threshold.

4. Design of Gynapse

To solve the challenges in Section 3, we design a system called Gynapse, which learns and predicts the exact usage of each device, and then switches their power modes accordingly to maximize the total energy saving under the constraint of user required response time. A simplified example of Gynapse is shown in **Fig. 3**[⋆1].

We obtain power status of PC, light, tea-maker, and microwave from current sensors, which are denoted as WK/LP for working/low power modes. Obviously, power consumption depends on device usage, which is closely correlated with sensors, such as RFID tag on microwave or Object Movement (OM) sensor on tea-maker (circles in Fig. 3). Therefore, we can infer the resident's activities and device usages from multi-modal sensors. As shown in Fig. 3, after using PC in office, the resident enters kitchen and turns on light at time 20; then he heats water with tea-maker and warms food with microwave at time 25 and 35 respectively; at time 80, he leaves kitchen and works on PC again. With this analysis, we know the PC is not used from time 20 to 80, so it could be switched into low power mode for saving energy (shaded bar in Fig. 3).

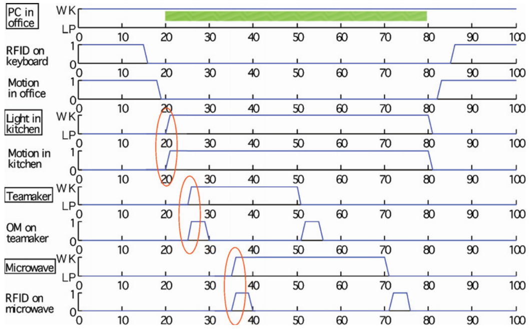


Fig. 3 An example of Gynapse.

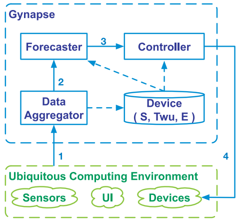


Fig. 4 System architecture.

To achieve this goal, we must proactively switch power modes of PC according to its predicted usage: if we predict the user will head for kitchen to make a snack, we shall turn PC into low power mode; if we predict he will return to work, we shall turn PC into working mode. Therefore, we need a functional part in Gynapse to predict device usage. Another consideration is, the user may feel frustrated because the long wake-up time of PC. Therefore, we need a functional part in Gynapse to take response time into consideration.

With these considerations, we design Gynapse's architecture as in **Fig. 4**. We assume a user interface (UI), sensor and control infrastructure exists in an in-

[⋆1] We rescale time line and ignore details of sensor data that will not detract our discussion.

door ubiquitous computing environment, which could be used to collect user feedback, detect human activities and switch power mode of devices. Gynapse consists of one adjunct database of device information, and three functional parts: Data Aggregator, Forecaster, and Controller. Solid arrows are main data flows. Data Aggregator receives information from different types of sensors/devices (arrow 1), normalizes their values, and builds them into a time series of vectors $\vec{O}_t = (s_t^1, s_t^2, \ldots, s_t^M)$, where M is the number of sensors and t denotes time. After receiving these vectors (arrow 2), Forecaster builds a three-level Hierarchical Hidden Markov Model (HHMM) to learn the usage pattern of devices, and forecast their usage probabilities through the predictive mechanism. With the predicted probabilities (arrow 3), Controller makes decisions of power mode switching to maximize the total power saving under response time constraint. These decisions (arrow 4) are then sent to the ubiquitous computing infrastructure to switch devices. If Gynapse makes any wrong decisions, user feedback can be obtained through user interface. The adjunct database provides the necessary information of current status (S), wake-up time (T_{WU}), and power saving (E) of each device.

In the following sections, we will discuss the details of probabilistic model, predictive mechanism in Forecaster, and control framework in Controller.

4.1 The Probabilistic Model

Before predicting the usage probability of devices, Forecaster must build a probabilistic model to learn the usage patterns of them. We use a three-level Hierarchical Hidden Markov Model (HHMM) to represent and learn residents status and device usages, because its hierarchical nature easily characterizes natural hierarchies in human activity and allows for model reusability and more efficient learning [6),26)]. Our contribution here is applying HHMM to model the device usage in a power management system, instead of developing a new extension of HHMM. Therefore, we focus on explaining the model of single resident and its generalization to multiple residents, rather than discussing the details of learning algorithms.

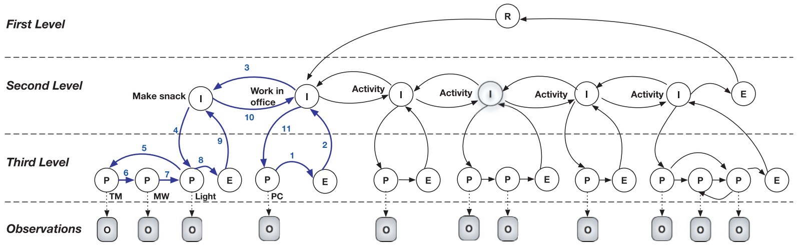


Fig. 5 State transition diagram of three-level HHMM. Shaded nodes are observed; the remaining nodes are hidden. The numbered arrows show the sequence of state transitions.

Figure 5 illustrates a model, whose left part with numbered arrows corresponds to our example in Fig. 3. Five different types of nodes are in this model: R, I, P, E, and O. The R node at the first level is Root state representing the status of residents at home. For single resident, it is state (1) if he is at home, otherwise it is state (0). The I nodes at the second level are Internal states, which represent activities of the resident, such as "make snack", or "work in office". They may have an arbitrary number of P nodes. The P nodes at the third level are Production states, which represent the device status, such as "Tea-maker (TM) is in use", or "microwave (MW) is not in use". Production state is the only one within HHMM that can emit observations, O nodes, which are the vectors sent by Data Aggregator. The E nodes are End states, which exist only to signal the horizontal transition is ended, and a vertical transition to upper level is needed.

Arrows between nodes represent probabilistic dependencies: solid arrows are transitions between hidden states of HHMM, and dotted line arrows are dependencies between Production states and observations. There are two kinds of state transitions: 1) horizontal transition at the same level, which means the resident is going to use another device or to another activity; 2) vertical transition between different levels, which means the resident starts or finishes using devices in an activity. For instance, the resident "turns off light" after "using microwave" (7), a transition to End state (8) means he has finished "making snack" and a transition back to the second level (9) is needed. Then he goes to "work in office" and "use PC", which are represented by transition (10) and (11). This model clearly represents the hierarchical structure in human behaviors and the transitions between device usages and activities.

When multiple residents are at home, it becomes a little more complex. If they always do the same activities, such as “watch TV” together, it will not be a problem. If they do different activities at the same time, we have to represent them separately in the model. As explained before, R node represents the status of residents at home. If there are two people, we add one dimension to R node, so the state (1, 1) represent both of them are at home. It is the same for I and P nodes. For instance, one state of I node will become (make snack, watch TV) to represent their activities respectively. Of course, as the dimension increases, the number of state will also increase. Fortunately, it will not increase exponentially, because the residents in a family tend to do the same activity most of the time, such as “have dinner” or “watch TV”.

After solving the representation challenges in Section 3.1, we now focus on learning usage patterns, which is actually learning the probabilistic dependencies: horizontal transition probabilities, vertical transition probabilities, and observation probabilities. We follow the notation in Ref. 6) to define them. The Root, Internal, Production and End states in Fig. 5 are uniformly represented by q_i^d, where $d \in [1, 2, 3]$ denotes the hierarchy level and i is the state index relative to its parent. E.g., q_i^2 means an Internal state at the second level. The probability of state q^{d-1} vertically transitioning to its children q_i^d is specified as $\pi^{q^{d-1}}(q_i^d)$. The probability of state q_i^d making a horizontal transition to state q_j^d is written as $a_{ij}^{q^d}$. Hence, for Root and each Internal state, vertical and horizontal transition probabilities are defined as the vector Π^{q^d} and the matrix A^{q^d} respectively. Production states at the third level q^3 are the only type can emit observations. Therefore, the observation probabilities of Production nodes is represented as the vector B^{q^3}, which defines the probability of state q^3 producing observation $O = k$ as $b^{q^3}(k) = P\left(O = k|q^3\right)^{\star 1}$. The model parameters are denoted in a compact form as $\lambda = (A, B, \Pi)$.

We use the forwards-backwards and expectation-maximization algorithms [7] to learn the parameters from historical data, and reduce the time complexity to $O(T)$, compared with $O(T^3)$ in the original paper of Ref. 6). The learned parameters $\lambda = (A, B, \Pi)$ are used to predict the probability of device usage via the predictive mechanism.

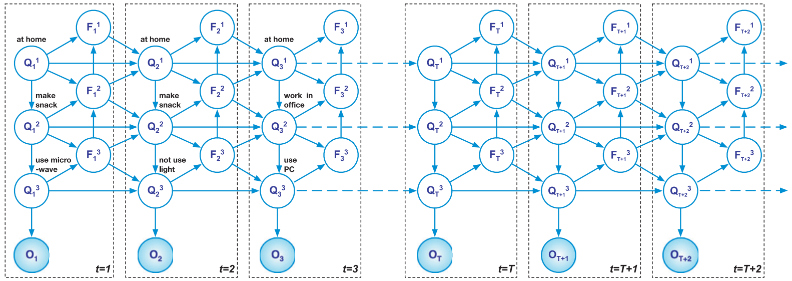


Fig. 6 Dynamic Bayesian Network of three-level HHMM. Shaded nodes are observed; the remaining nodes are hidden. Q_t^d is the state at time t, level d; $F_t^d = 1$ if the HHMM at level d has finished (entered its end state), otherwise $F_t^d = 0$.

4.2 The Predictive Mechanism

After learning the usage patterns, Forecaster predicts the usage of multiple devices in future through predictive mechanism. In order to show the temporal relationship more clearly, we redraw the three-level HHMM as a Dynamic Bayesian Network (DBN)[7],[8] in **Fig. 6**, which is identical to the state transition diagram in Fig. 5. Although Fig. 5 emphasize the parent-children relationship between states, Fig. 6 highlights the temporal sequence of them.

Figure 6 shows the three-level HHMM as time slices. At time slice t, the observation is $\vec{O}_t = \left(s_t^1, s_t^2, \ldots, s_t^M\right)$; the state at level d is denoted as Q_t^d, so the Root, Internal, and Production states are Q_t^1, Q_t^2, and Q_t^3 respectively. F_t^d is a binary indicator that is 1 if it has entered an End state; otherwise it is 0. The downward arrows between the Q variables represent a state “call” its child. The upward arrows between the F variables enforce the fact that a higher-level state can only change when lower-level one finished.

The example in Fig. 5 is also shown in Fig. 6. Suppose at $t = 1$, Q_1^1 represents the resident is at home, Q_1^2 represents he is “making snack”, and Q_1^3 represents he is “using microwave”. Then at $t = 2$, Q_2^1 and Q_2^2 keep unchanged, while Q_2^3 changes its state to “turn off light”. Since it is the end of “make snack”, F_2^3 becomes 1, which means it has finished and Q_2^2 can change state for the next time slice. At $t = 3$, Q_3^1 is still unchanged, Q_3^2 changes to “work in office”, and Q_3^3 changes to “use PC”.

⋆1 This can be considered as if q^3 is in the state “using PC”, then we can observe reading from RFID tag on keyboard.

Given the learned parameters $\lambda = (A, B, \Pi)$ in Section 4.1, we first explain how to predict the states at the next *one* time slice, and then generalize to next N time slices.

Suppose we are at $t-1$, and know the states Q_{t-1}^d and F_{t-1}^d, where $d \in [1, 2, 3]$. We try to predict the states at t, Q_t^d and F_t^d. We explain the Production, Internal and Root states separately.

Production state (the third level): Production state Q_t^3 follows a Markov chain with parameters determined by higher-level states $Q_t^{1:2}$, which we denote as k for brevity.

If Q_{t-1}^3 does not enter an End state, the value of Q_t^3 should be drawn from horizontal transition probability A. Otherwise, it turns F_{t-1}^3 to be 1 to signal it is finished, and draws the value of Q_t^3 from vertical transition probability Π. Formally, we can write the probability that Q_t^3 will be in state j as follows:

$$P\left(Q_t^3 = j | Q_{t-1}^3 = i, F_{t-1}^3 = f, Q_t^{1:2} = k\right) = \begin{cases} a_{ij}^{q^3} & \text{if} \quad f = 0 \\ \pi^{q^2}(q_j^3) & \text{if} \quad f = 1 \end{cases} \tag{1}$$

where $a_{ij}^{q^3}$ and $\pi^{q^2}(q_j^3)$ are the horizontal and vertical transition probabilities explained in Section 4.1. For clarity, $i, j \neq$ End state here, because the End state is indicated with F_t^3 as:

$$P\left(F_t^3 = 1 | Q_t^{1:2} = k, Q_t^3 = i\right) = a_{i,end}^{q^3} \tag{2}$$

Internal state (the second level): Similar to Production state, Internal state Q_t^2 follows a Markov chain with parameters determined by Q_t^1, and F_{t-1}^2 specifies whether we should use the horizontal or vertical transition probability. The difference is that we now also get a signal from lower-level F_{t-1}^3: if it has finished, we are free to change Internal state, otherwise we must remain in the same state. Formally, we can write the probability that Q_t^2 is in state j as:

$$P\left(Q_t^2 = j | Q_{t-1}^2 = i, F_{t-1}^3 = b, F_{t-1}^2 = f, Q_t^1 = k\right) = \begin{cases} \delta(i,j) & b = 0 \\ a_{ij}^{q^2} & b = 1 \text{ and } f = 0 \\ \pi^{q^1}(q_j^2) & b = 1 \text{ and } f = 1 \end{cases} \tag{3}$$

where $\delta(i, j) = 1$ if $i = j$, otherwise $\delta(i, j) = 0$. F_t^2 should turn to 1 only if Q_t^2 is "allowed" to enter an End state. Formally, we can write this as follows:

$$P\left(F_t^2 = 1 | Q_t^1 = k, Q_t^2 = i, F_t^3 = b\right) = \begin{cases} 0 & \text{if} \quad b = 0 \\ a_{i,end}^{q^2} & \text{if} \quad b = 1 \end{cases} \tag{4}$$

Root state (the first level): The Root state Q_t^1 differs from Internal states in that it has no parent to specify which distribution to use. The equations are the same as above, except we eliminate the conditioning on $Q_t^1 = k$.

Following Eqs. (1)–(4), we can obtain the probabilities of each state at t, $P(Q_t^d)$ and $P(F_t^d)$, given the parameters $\lambda = (A, B, \Pi)$ and the states Q_{t-1}^d and F_{t-1}^d. Now we generalize it to next N time slices.

As explained in Section 3.2, we must consider wake-up time T_{WU} and user required response time T_{RS} for predicting. Suppose we are at time T, and need predict the probability of device usage at $T+(T_{WU}-T_{RS})$. Suppose $(T_{WU}-T_{RS})$ is equal to the length of 2 time slices[⋆1], so our problem becomes predicting the probability of Q_{T+2}^d, given the current states Q_T^d. We follow Eqs. (1)–(4) to obtain the probability of Q_{T+1}^d at first, and then re-use Eqs. (1)–(4) to calculate the probability of Q_{T+2}^d based on Q_{T+1}^d. For devices with different $(T_{WU}-T_{RS})$, we need repeat this process for different times.

The third challenge discussed in Section 3.2 is the accuracy of prediction. For instance, Forecaster predicts the resident will "work in office" after "making snack", whereas he may actually go to "watch TV". Essentially, it is impossible to *completely* eliminate the incorrect prediction. However, we believe the possibility still exists to improve Gynapse's accuracy. A potent candidate is integrating user feedback with predictive mechanism. As shown in Refs. 16), 17), the accuracy can be improved by introducing high-level information, such as user feedback. It actually involves two separate problems: the first is soliciting feedback from user. This one has been intensively studied by researchers of computer-human interface, such as explicit feedback from mobile touch screens [27),28)] or implicit feedback from sensor networks [29),30)]. With this kind of technologies, we can easily

[⋆1] For simplicity, we use discrete time Markov chain here, so $(T_{WU} - T_{RS})$ is equal to an integral number of time slices.

obtain feedback from users. The second and key problem is integrating feedback with learning and predicting mechanism in Forecaster. The general approaches are: 1) treating user feedback as hard constraints to Bayesian Network's learning algorithms [18),31),32)], or 2) incorporating them in the prior knowledge [33)]. Since the correctness of the learned model obviously depends on the training samples, we use a training sample selection method similar to Ref. 18). Suppose we obtain one feedback to indicate the "make snack→watch TV" transition is correct, while "make snack→work in office" is incorrect. We replace one set of observation vectors that represent "make snack→work in office", with one set representing "make snack→watch TV" in training samples; we will replace two sets if we get two feedbacks. As a result, we punish the wrong prediction and reward the user's correction, so we can gradually improve prediction accuracy when we re-learn the model parameters from new training samples.

In this way, we have solved all the challenges in Section 3.2. At last, we send the predicted probabilities of Production states to Controller, since they are the wake-up probabilities of devices.

4.3 The Control Framework

After receiving the predicted wake-up probabilities, Controller switches power mode of multiple devices to maximize the total energy saving under the constraint of user required response time. As explained in Section 3.3, since the user required response time T_{RS} has been considered in predicted wake-up probabilities, the problem actually becomes choosing a scenario that will save the maximum energy after power mode switching, under the constraint that the wake-up probability of that scenario is higher than a threshold. Four steps are necessary.

The first step is to calculate the predicted probability of all scenarios. We use I_i to indicates the usage of device i, where $I_i = 1$ if it is awoken to working mode, and 0 if it is in low power mode. Hence, the predicted probability of device i can be written as:

$$P_i(I_i) = \begin{cases} p_i & I_i = 1 \\ 1 - p_i & I_i = 0 \end{cases}$$

where p_i is the wake-up probability received from Forecaster.

If we have N devices, the predicted scenario can be represented as $(I_1, I_2, \ldots, I_N)$, and its probability is:

$$P(I_1, I_2, \ldots, I_N) = \prod_{1 \le i \le N} P_i(I_i) \tag{5}$$

As explained in Section 3.3, we choose scenarios whose probability is higher than 0.5^N as the "possible" pool. We use a pool of scenarios instead of the top one, because the actual scenario more likely falls into a high probability pool, rather than exactly matches the top 1 predicted scenario[⋆1]. At the same time, we also narrow down the number of scenarios that need to be processed for following steps.

The second step is to calculate the energy saving of selected scenarios. We use e_i to represent the energy saving of device i, which is the difference of energy consumption between working and low power modes. Then the energy saving of scenario $(I_1, I_2, \ldots, I_N)$ is:

$$ES(I_1, I_2, \ldots, I_N) = \sum_{1 \le i \le N} e_i * (1 - I_i)$$

which means device i can save energy when it is in low power mode ($I_i = 0$). To make it comparable, we normalize the energy saving as:

$$\widetilde{ES}(I_1, I_2, \ldots, I_N) = \frac{ES(I_1, I_2, \ldots, I_N)}{\sum\limits_{1 \le i \le N} e_i * 1} = 1 - \frac{\sum\limits_{1 \le i \le N} e_i * I_i}{\sum\limits_{1 \le i \le N} e_i * 1} \tag{6}$$

The third step is to calculate the product of energy saving and predicted probability for the selected pool. The product of scenario $(I_1, I_2, \ldots, I_N)$ can be calculated by multiplying Eq. (5) and Eq. (6) as:

[⋆1] We can consider the following example: Suppose we predict the top 1 scenario is S_1. If our prediction is perfect, the actual scenario must match S_1. However, since no existing algorithms can achieve 100% accuracy, there must be some actual scenarios mismatch the top 1 prediction. Now we consider an "extreme" pool, which includes *ALL* the possible scenarios. In this case, the actual scenario must fall in this pool with 100% accuracy. This phenomenon has been broadly observed across predicting algorithms. Some numerical results can be found in Refs. 34), 35), where the top 4 or 5 prediction always achieve higher accuracy than the top 1 prediction.

$$U(I_1, I_2, \ldots, I_N) = \widetilde{ES}(I_1, I_2, \ldots, I_N) * (1 + P(I_1, I_2, \ldots, I_N)) \tag{7}$$

We choose the scenario with the maximum product as the "ideal" one, because it has both high probability and large energy saving. The "1" in Eq. (7) guarantees that we first compare energy saving of scenarios; if energy saving are equal, then we compare their probabilities. In this way, we choose a scenario that maximizes the total energy saving and its wake-up probability is higher than the threshold of 0.5^N to solve the challenge in Section 3.3.

At the last step, the current scenario is compared with the "ideal" one, and decisions of power mode switching are sent to the ubiquitous computing environment for execution.

After learning the usage patterns from historical data, Gynapse repeats prediction and these four steps every time slice. In this way, it can handle power mode switching to maximize the total power saving while ensuring that response time does not exceed the required level.

5. Implementation and Evaluation

In this section, we shall discuss the details of implementation and evaluation of Gynapse. We first describe the data and environment of our experiments, and then discuss the preliminary results.

5.1 Experimental Environment

We use sensor data from MIT PlaceLab [9] to evaluate our system. PlaceLab is a 1,000 sq. ft. apartment consisting of a living room, dining area, kitchen, small office, bedroom, full bath and half bath. 15 types of sensors are installed in PlaceLab: Interior conditions of the apartment are captured using distributed temperature, humidity, bright, and barometric pressure sensors. The PlaceLab also features electrical current sensors, water flow and gas flow sensors. Small, wired switches detect open/close events, such as the opening of linen closet. RFID tags and MITes object movement sensors can be easily taped onto any non-wired objects such as chairs, cups, remotes, tea-maker, and other objects people may manipulate [36]. A resident in the PlaceLab can wear up to three wireless 3-axis, 0-10 G accelerometers that measure limb motion. A wireless heart rate monitor (using a standard Polar chest strap) can also be worn. Five receivers spread throughout the apartment collect all wireless object motion, accelerometer, and heart rate data sent via the MITes wireless sensors [36]. Nine infrared cameras, 9 color cameras, and 18 microphones are distributed throughout the apartment in cabinet components and above working surfaces. From these multi-modal sensors a resident's activities can be inferred [9].

Table 1 Sensors used in Gynapse.

Type	Living	Dining	Kitchen	Office	Bedroom	Hallway
Current	3	3	7	2	2	3
Switch	4	7	20	3	5	5
MITes OM	25	16	40	14	15	25
RFID	10	3	27	13	12	8
Motion	2	1	1	1	1	2
Water	—	—	4	—	—	2
Gas	—	—	1	—	—	—

Table 2 Devices used in Gynapse.

	devices
Living	2 lights, TV, DVD player, speaker,
Kitchen	2 lights, microwave, coffee maker, tea maker, toaster dish washer, mixer, can opener, stove, garbage disposer
Dining	light, answer machine
Bedroom	light, lamp, alarm,
Hallway	light, washer, dryer
Office	light, computer, monitor, fax machine

People may argue that, in real life, no residents will install so many sensors in their house because of cost and complexity. However, it is important for research purpose to reveal the effectiveness of energy saving for all sensors before we can narrow them down. As a result, we select a set of 286 sensors that have close correlation with device usage to build our system, which are shown in **Table 1**.

Since the sensor values are quite different, Data Aggregator normalizes them into a range between 0 and 1, and builds them into time series as $\vec{\Theta} = [\vec{O}_1, \vec{O}_2, \ldots, \vec{O}_T]$, where vector $\vec{O}_t = \left(s_t^1, s_t^2, \ldots, s_t^M\right)$ is the observations of HHMM, as explained in Section 4.

We use multi-modal sensors excluding current sensors to obtain the usage information of 28 electronic devices, which are listed in **Table 2**.

We use current sensors to obtain a single device's wake-up time, and energy

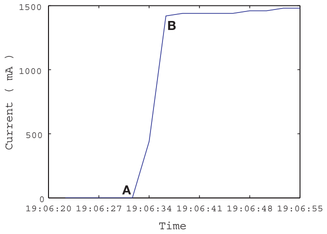


Fig. 7 Current reading of light.

saving (which is the difference current readings between working and low power mode). **Figure 7** shows the current reading of living room light. Time at point A and B are 19:06:29.693 and 19:06:34.287 respectively, so the wake-up time of this light is about 4.5 seconds. We can also find the current difference between working and low power mode is about 1,500 mA. For multiple devices, we can calculate their correlation of current status. **Figure 8** shows the current status of coffee maker, toaster, and PC in a morning. We can observe the obvious linkage between coffee maker and toaster. Statistically, we have the time series of current consumed by coffee-maker and PC as $C_{CM,t}$ and $C_{PC,t}$, so their normalized cross-correlation can be calculated as:

$$corr = \frac{1}{n-1}\sum_{t=t_1}^{t_n}\frac{\left(C_{CM,t}-\bar{C}_{CM}\right)\left(C_{PC,t}-\bar{C}_{PC}\right)}{\sigma_{CM}\sigma_{PC}}$$

where $\bar{C}$ and σ are mean and standard deviation of time series, and n is the total number of data points. During this period, the correlation between coffee maker and toaster is 0.7870, whereas the correlation between coffee maker and PC is -0.0968. The numerical results also prove our argument in Section 3.1.

There are totally 23 days of data available, within which we use 9 days to train our three-level HHMM, and the rest 14 days for evaluation.

5.2 Preliminary Results

In this paper, we have mainly discussed three components of Gynapse: 1) a probabilistic model to learn residents' usage patterns at device level, 2) a predictive mechanism to forecast the usage probability of devices, and 3) a control framework to maximize the energy saving under the constraint of user required response time. Correspondingly, we evaluate Gynapse from four prospects: 1) the learning curve of probabilistic model, 2) the precision and recall rate of predictive mechanism, 3) the balance of energy saving and user required response time, and 4) the total energy saving on each day.

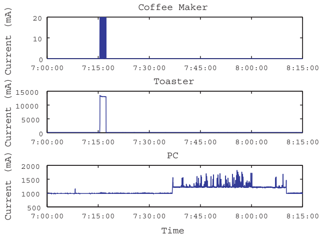


Fig. 8 Usage correlation of devices.

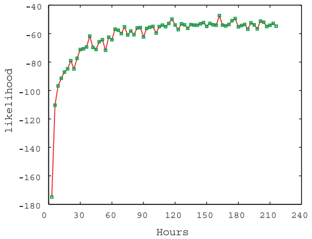


Fig. 9 Learning curve.

5.2.1 Learning Curve

Figure 9 shows the learning curve of our three-level HHMM. At first the likelihood is very low, because the training data is not enough and the parameters have not been properly learned. As we add more training data, the likelihood improves gradually. After about 100 hours of training data, the likelihood becomes stable. Roughly speaking, we can consider the training data are enough. However, we have to be aware of two things. First, strictly speaking, the exact hours needed depends on the initial values of parameters, training policy, and convergence criterion of the learning algorithm [37]. E.g., if we require the converge threshold to be 0.001, we obviously need more training data than a

Table 3 Precision and recall of predicted usage.

Devices	Precision	Recall	Devices	Precision	Recall
Living			**Bedroom**		
light	95%	93%	light	94%	93%
TV	93%	92%	lamp	91%	89%
DVD player	89%	85%	alarm	84%	81%
Kitchen			**Dining**		
light	96%	93%	light	93%	92%
microwave	87%	84%	answer machine	86%	82%
coffee maker	85%	83%	**Hallway**		
tea maker	86%	84%	light	93%	91%
toaster	83%	82%	washer	89%	88%
dish washer	86%	83%	dryer	88%	85%
mixer	82%	81%	**Office**		
can opener	80%	78%	light	94%	93%
stove	81%	80%	PC	92%	90%
garbage disposer	79%	77%	fax machine	85%	82%

threshold of 0.1. Meek et al. provide more discussion about sample size and learning curve in Ref. 38). Second, for some devices such as a stove, the resident does not use it everyday. Therefore, in practice, we need more days to collect training sample of stove than a heavily used device such as TV. The small peaks and valleys on the curve after 100 hours reflect this point.

5.2.2 Precision and Recall

Table 3 shows the precision and recall of predicted device usage, which are defined as:

$$Precision = \frac{TP}{TP+FP} \quad Recall = \frac{TP}{TP+FN}$$

where TP (True Positive) means we predict a device to be used, and it is actually used; FP (False Positive) means we predict a device to be used, but actually it is not used; FN (False Negative) means we predict a device not to be used, but it is actually used. Please note that, as explained in Section 5.1, we determine the actual usage from multi-modal sensors excluding current sensor. In the example of Fig. 3, we use RFID tag on keyboard to determine the resident actually starts or stops using PC, but not the current sensor of PC.

From Table 3, we find the predictive mechanism has better performance for some devices such as lights, TV, and PC. This is because they are used more frequently, so more training samples are available.

Table 4 Combination of required response time.

Combination	Lights (sec)	TV (sec)	PC (sec)
Comb_0	4	4	4
Comb_1	2	2	2
Comb_2	4	6	6
Comb_3	4	10	10
Comb_4	4	10	30
Comb_5	4	10	44

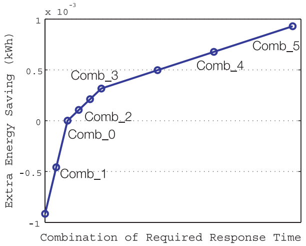


Fig. 10 Energy saving vs. Response time.

5.2.3 Energy Saving and Required Response Time

As explained in Section 4.3, control framework maximizes the energy saving under the constraint of user required response time T_{RS}. Since the upper limit of T_{RS} is a device's wake-up time T_{WU}, we use lights, TV, and PC to represent three categories of devices according to their T_{WU}: lights represent the devices with a short wake-up time such as 4.5 seconds. Therefore, the resident should expect the lights to response within 4 seconds[⋆1]. On the other end, PC represents the devices with a long wake-up time such as 44 seconds. Then the required response time of PC can range from 0 to 44 seconds. TV represents the category between these two. We set a benchmark (Comb_0), where the required response time is 4 seconds for all devices, and then calculate the extra energy saving of different combinations of required response time. **Table 4** shows the different combinations.

Figure 10 depicts the extra energy saving (compared with Comb_0) of different combinations. We can find that, if the resident tolerates slow response from devices, such as Comb_5, more energy saving is achieved. In contrast, if the

[⋆1] For simplicity, we only use integer here.

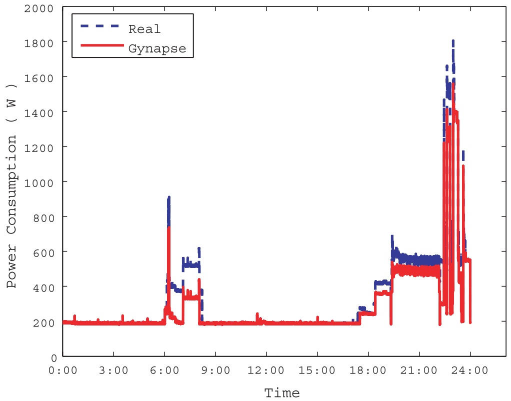


Fig. 11 The real and Gynapse-adjusted power consumption of one day.

resident need immediate response from devices, such as Comb_1, less energy saving is achieved. It clearly shows Gynpase's capability of balancing energy saving and required response time.

5.2.4 Total Energy Saving

After setting the user required response time, control framework calculates the "ideal" scenario, compares it with the current one, and switches power mode accordingly. If a device is not used in "ideal" scenario, but its power status is in working mode, then we can turn it into low power mode for saving energy. In this way, we can calculate the total power consumption based on the decisions from Gynapse.

Figure 11 shows the real and Gynapse-adjusted total power consumption of one day after the HHMM has been properly learned. We can find that the energy saving are mainly achieved during morning and evening, when multiple electronic devices are used. While in the early morning and noon, since no devices are used, almost no power saving is achieved.

Table 5 shows the real power consumption, Gynapse-adjusted power consumption, and saving ratio of 14 evaluation days⋆1. We find that Gynapse averagely saves about 11% of energy on these 14 days. The fluctuation of the saving ratio depends on the resident's behavior. For instance, since the resident was not at home on the 4th day, less power was consumed and no power saving was achieved.

Table 5 Power consumption and saving ratio of each day.

Day	1	2	3	4	5	6	7
Real (kWh)	8.64	9.82	9.36	3.11	7.68	12.55	7.72
Gynapse (kWh)	7.79	8.57	8.66	3.11	6.83	11.37	7.09
Saving Ratio	9.78%	12.72%	7.46%	0.00%	11.08%	9.36%	8.13%
Day	8	9	10	11	12	13	14
Real (kWh)	9.22	14.35	10.63	9.10	9.09	10.51	9.59
Gynapse (kWh)	8.58	12.37	9.00	7.83	7.90	9.26	8.47
Saving Ratio	7.01%	13.77%	15.32%	14.03%	13.01%	11.89%	11.77%

6. Conclusions and Discussions

In this paper, we have designed a ubiquitous power management system called Gynapse, which uses multi-modal sensors to predict the exact usage of each device, and then switches their power modes based on predicted usage to maximize the total energy saving under the constraint of user required response time. We have discussed the challenges and solutions for three important components of Gynapse:

- We have built a three-level Hierarchical Hidden Markov Model (HHMM) to represent multiple residents and hierarchical structure of their activities, and learn the adaptive device usage patterns in different situations from multi-modal sensors.
- Based on the learned HHMM, we have developed our predictive mechanism in Dynamic Bayesian Network (DBN) scheme to precisely predict the usage of each device, with device's wake-up time and user's required response time under consideration.
- With the predicted usage probability of each device, we have followed a four-step process to balance the total energy saving and response time of devices by switching their power modes according to the scenario with the highest energy saving and probability.

Correspondingly, we have used PlaceLab data set to evaluate Gynapse from four prospects: 1) the learning curve of three-level HHMM, 2) the precision

⋆1 According to Ref. 39), the average household power consumption in Massachusetts is 21.17 kWh per day. Since PlaceLab is a 1,000 sq. ft. apartment with a single person, it consumes less electricity than a typical 2,000+ sq. ft. house with four persons.

and recall rate of predictive mechanism, 3) the balance of energy saving and user required response time, and 4) the total energy saving on each day. The preliminary results have demonstrated that as a ubiquitous power management system, Gynapse has the capability to reduce power consumption while keeping response time not exceed user requirement. It provides a complementary rather than competitive appoach to previous power mode switching systems.

6.1 Discussions

In this section, we provide some discussions about Gynapse as follows:

Undesirable aspects of overly energy saving: "Overly energy saving" means a device should be in working mode, but it is switched off for saving energy. As a result, it may cause two "undesirable aspects" as follows:

- *Incorrect switch-on/off:* e.g., when the resident goes to restroom during his work on PC, Gynapse incorrectly turns PC off. Although the resident is in a short break, Gynapse switches PC off incorrectly for saving energy.
- *Delay of response:* e.g., Gynapse does not turn on PC until the resident sits in front of it, so he has to wait for a long wake-up time. In this case, the resident wants to use a device, and Gynapse correctly predicts that. However, for saving energy, Gynapse does not switch it on until the last moment.

In case 1, Gynapse incorrectly predicts the resident's activity. Essentially, it is impossible to *completely* eliminate the incorrect prediction. However, we believe the possibility still exists to improve Gynapses accuracy. As we discussed in Section 4.2, a potential method is integrating user feedback with predictive mechanism. Of course, the incorrect prediction cannot be completely avoided even if we introduce user feedback into the system. When Gynapse incorrectly predict device usage, the resident has to explicitly define his policies, like in a rule-based system, or manually switch the devices. Incorrect prediction is the limitation of Gynapse, and the choice among Gynapse, rule-based systems and manual switching depends on the resident and his situation.

In case 2, Gynapse correctly predicts the resident's activity, but the resident may still feel frustrated because of the long wake-up time. To solve this problem, we provide a variable in predictive mechanism to reflect user's required response time. Then Gynapse tries to maximize the energy saving under the constraint of user's required response time.

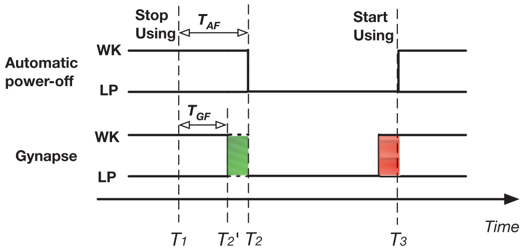


Fig. 12 Automatic power-off vs. Gynapse.

Comparison of Gynapse and automatic power-off mechanism in terms of energy saving: We have already discussed in Section 2 that the automatic power-off mechanism cannot handle complicated situations, such as the resident may watch TV when cooking in kitchen. However, in terms of energy saving, it is still a simple and effective method, which can be found in infrared-controlled lights or electric pot. A simple example is depicted in **Fig. 12**. Assume the resident stops using an electric pot at T_1; after a certain time T_{AF}, it is automatically switched off to low-power model (LP) at T_2. When the resident starts using the pot again at T_3, it is switched on. We compare the energy saving of automatic power-off mechanism with Gynapse at switch-off (around T_2) and switch-on (around T_3) respectively.

At switch-off, Gynapse uses the resident's activity to help make decision. For instance, if Gynapse finds the resident has left home after using the pot, it can turn off the pot at $T_2^{'}$ without waiting for a certain time. Whether Gynapse saves more or less energy than automatic mechanism at switch-off actually depends on the length of T_{AF} and T_{GF}: if T_{AF} is longer, then Gynapse saves more energy; if T_{AF} is shorter, then Gynapse saves less energy.

At switch-on, since Gynapse predicts the resident's activity to shorten response time of devices, it cannot save more energy than automatic power-off mechanism. For instance, an electric pot may need some time to heat water to a certain temperature, such as 100°C for tea. If Gynapse predicts that the resident is going to make some tea, it can turn on the pot to heat water before T_3. Hence, the resident need not wait for a long response time. In contrast, the automatic power-off mechanism will not start heating the water until the resident turns it on. As a result, Gynapse can help the resident by shortening response time, though this means less power saving.

Based on the discussion above, we find that for a simple device, whether the automatic power-off mechanism may save more or less energy depends on the summation of switch-off and switch-on parts. However, Gynapse can save energy for devices used in complicated situations, which may not be easily handled by the automatic power-off mechanism. In addition to energy saving, Gynapse also considers the response time of devices.

Appropriate information for human activity prediction: Gynapse aims at providing a flexible scheme that can incorporate a variety of information into the prediction of human activity, rather than comparing which information is the most appropriate. As we explained in Section 4, the input of our system is a time series of vectors $\vec{O}_t = \left(s_t^1, s_t^2, \ldots, s_t^M\right)$, where s_t^i corresponds to the value of sensor i at time t. No matter this "sensor" is a real sensor or a virtual one, such as "a day of the week", it can always be put into the vector. This scheme provides great flexibility to our choice of information. With the development of sensors and human activity recognition, we can incorporate new information into our system.

Comparison with other power management systems: We are fully aware of the importance of comparing Gynapse with previous power management systems. However, previous systems are all built in their own environments with different settings of devices and sensors. The lack of benchmark makes the comparison very difficult if not impossible. Our evaluation with MIT PlaceLab data set reflects our effort to use a public data source, so it may be easier for other researchers to compare with us. However, since PlaceLab data is not designed for evaluating energy consumption, it inevitably has some limitations, and we have to leave some situations untested, such as multi-resident models and integration of user feedback. Currently, we are building a multi-modal sensor database mainly for power consumption evaluation in our laboratory. We believe the release of our data will provide a high quality database for further research and comparison.

References

1) Agency for Natural Resources and Energy: FY 2007 Energy Supply and Demand Report (Final Report) (2009). (in Japanese).
2) Ministry of Internal Affairs and Communications: Workshop Report concerning ICT Policy for Correspondence to Global Warming Issue (2008). (in Japanese).
3) Yamazaki, T., Jung, J., Kim, Y. and Hahn, M.: Energy Management in Home Environment Using a Power Sensor Network, Technical Report 430, IEICE (2008). (in Japanese).
4) Kudo, H.: IT Applications for Energy Conservation, *J. IEICE*, Vol.90, No.3, pp.183–190 (2007). (in Japanese).
5) Harle, R.K. and Hopper, A.: The Potential for Location-aware Power Management, *Proc. UbiComp 2008*, pp.302–311 (2008).
6) Fine, S., Singer, Y. and Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications, *Mach. Learn.*, Vol.32, pp.41–62 (1998).
7) Murphy, K. and Paskin, M.: Linear Time Inference in Hierarchical HMMs, *Proc. Neural Information Processing Systems* (2001).
8) Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning, PhD Thesis, UC Berkeley, Berkeley (2002).
9) Intille, S.S., Larson, K., Tapia, E.M. and Beaudin, J.: Using a Live-in Laboratory for Ubiquitous Computing Research, *Proc. PERVASIVE 2006*, pp.349–365, Springer-Verlag (2006).
10) Ishida, K. and Ito, Y.: Trend of Measurement and Control Technologies in IT Era (4): Consumer Electronic Synchronization Control by HEMS, *J. Society of Heating, Air-Conditioning and Sanitary Engineers of Japan*, Vol.80, No.5,pp.385–393 (2006). (in Japanese).
11) Mozer, M.C.: The Neural Network House: An Environment that Adapts to its Inhabitants, *Proc. AAAI Spring Symp. on Intell. Environ.*, Menlo Park, CA, AAAI, pp.110–114 (1998).
12) Mozer, M.C.: *Lessons from an Adaptive House, Smart environments: Technologies, Protocols, and Applications*, chapter12, pp.273–294, J. Wiley & Sons (2005).
13) Roy, A., DasBhaumik, S., Bhattacharya, A. and Basu, K.: Location Aware Resource Management in Smart Homes, *Proc. PerCom2003*, pp.481–488, IEEE (2003).
14) Harris, C. and Cahill, V.: Exploiting User Behaviour for Context-aware Power Management, *Proc. WiMob 2005*, pp.122–130 (2005).
15) Harris, C. and Cahill, V.: An Empirical Study of the Potential for Context-Aware Power Management, *Proc. UbiComp 2007*, pp.235–252, Springer Berlin (2007).
16) Lu, Y.H. and DeMicheli, G.: Comparing System-Level Power Management Policies, *IEEE Des. Test Comput.*, Vol.18, No.2, pp.10–19 (2001).
17) Stumpf, S., Sullivan, E., Fitzhenry, E. and Oberst, I.: Integrating Rich User Feedback into Intelligent User Interfaces, *Proc. IUI '08*, ACM (2008).
18) Culotta, A., Kristjansson, T., McCallum, A. and Viola, P.: Corrective Feedback and Persistent Learning for Information Extraction, *Artif. Intell.*, Vol.170, No.14, pp.1101–1122 (2006).

19) Philipose, M., Fishkin, K.P., Perkowitz, M. and Patterson, D.J.: Inferring Activities from Interactions with Objects, *IEEE Pervasive Comput.*, Vol.3, No.4, pp.50–57 (2004).

20) Yamahara, H., Takada, H. and Shimakawa, H.: A Behavioral Pattern Adapted to Individual for Providing Ubiquitous Service in Intelligent Space, *Proc. 5th WSEAS Intl. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics*, WSEAS, pp.84–92 (2006).

21) Moncrieff, S., Venkatesh, S., West, G. and Greenhill, S.: Multi-modal Emotive Computing in a Smart House Environment, *Pervasive Mob. Comput.*, Vol.3, No.2, pp.74–94 (2007).

22) Chetty, M., Brush, A.J.B., Meyers, B.R. and Johns, P.: It's Not Easy Being Green: Understanding Home Computer Power Management, *Proc. CHI 2009*, pp.1033–1042, ACM (2009).

23) Naeem, U., Bigham, J. and Wang, J.: Recognising Activities of Daily Life Using Hierarchical Plans, *Proc. EuroSSC 2007*, Vol.4793, pp.175–189, Springer-Berlin (2007).

24) Patterson, D., Fox, D., Kautz, H. and Philipose, M.: Fine-grained Activity Recognition by Aggregating Abstract Object Usage, *Proc. Wearable Computers 2005*, pp.44–51, IEEE (2005).

25) National Institute of Population and Social Security Research: Summary of Household Projection for Japan (2008). (in Japanese).

26) Theocharous, G.: Hierarchical Learning and Planning in Partially Observable Markov Decision Processes, PhD Thesis, Michigan State University, Lansing (2002).

27) Hoggan, E., Brewster, S.A. and Johnston, J.: Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens, *Proc. CHI 2008*, ACM (2008).

28) Lee, J.H. and Spence, C.: Assessing the Benefits of Multimodal Feedback on Dual-task Performance under Demanding Conditions, *Proc. 22nd British HCI Group Annual Conf. on HCI 2008* (2008).

29) Martins, D.S., Biajiz, M., Prado, A.F. and Souza, W.L.: Implicit Relevance Feedback for Context-aware Information Retrieval in UbiLearning Environments, *Proc. SAC '09*, ACM (2009).

30) Kaufmann, O., Lorenz, A., Oppermann, R. and Schneider, A.: Implicit Interaction for Pro-active Assistance in a Context-adaptive Warehouse Application, *Proc. Mobility '07*, ACM (2007).

31) Niculescu, R.S., Mitchell, T.M. and Rao, R.B.: Bayesian Network Learning with Parameter Constraints, *J. Mach. Learn. Res.*, Vol.7 (2006).

32) Stumpf, S., Rajaram, V., Li, L., Wong, W.K. and Burnett, M.: Interacting Meaningfully with Machine Learning Systems: Three Experiments, Technical Report 46, EECS Oregon State University (2007).

33) Zigoris, P. and Zhang, Y.: Bayesian Adaptive User Profiling with Explicit & Implicit Feedback, *Proc. CIKM '06*, ACM (2006).

34) Li, L., Muller, M.J., Geyer, W. and Dugan, C.: Predicting Individual Priorities of Shared Activities Using Support Vector Machines, *Proc. 16th Conf. on Information and Knowledge Management*, pp.515–524, ACM (2007).

35) Tanaka, K., Kishino, Y., Terada, T. and Nishio, S.: A Destination Prediction Method Using Driving Contexts and Trajectory for Car Navigation Systems, *Proc. SAC '09*, pp.190–195, ACM (2009).

36) Tapia, E.M., Intille, S.S., LOpez, L. and Larson, K.: The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection, *Proc. PERVASIVE 2006* (2006).

37) Meek, C., Thiesson, B. and Heckerman, D.: The Learning Curve Method Applied to Clustering, Microsoft Research Technical Report 34 (2001).

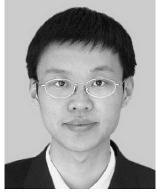
38) Meek, C., Thiesson, B. and Heckerman, D.: The Learning-Curve Sampling Method Applied to Model-Based Clustering, *J. Mach. Learn. Res.*, Vol.2, pp.397–418 (2002).

39) Energy Information Administration: U.S. Average Monthly Bill by Sector, Census Division, and State 2007 (2009).

(Received July 8, 2009)

(Accepted December 17, 2009)

(Original version of this article can be found in the Journal of Information Processing Vol.18, pp.147–163.)



Hua Si received his B.E. from Tsinghua University, Beijing, China, in 2002, M.E. from the University of Tokyo, Tokyo, Japan, in 2005, and M.F.E. from University of California, Berkeley, United States, in 2008, respectively. He is currently a doctoral student of the Research Center for Advanced Science and Technology at the University of Tokyo. His research interest is in the area of ubiquitous computing and machine learning. He is a student member of IEEE, ACM, and IEICE.

Shunsuke Saruwatari received his Dr. Sci. degrees from the University of Tokyo, Tokyo, Japan, in 2007. Since 2008, he had been in the University of Tokyo and currently a research associate of the Research Center for Advanced Science and Technology at the University of Tokyo. From 2007 to 2008, he stayed in University of Illinois at Urbana-Champaign as a visiting researcher. His research interests are in the areas of wireless sensor networks, real-time computing, real-time communication, and wireless networks. He is a member of IEEE, ACM, IEICE, and IPSJ.

Masateru Minami received his B.E. and M.E. degrees from Shibaura Institute of Technology, Tokyo, Japan, in 1996 and 1998, respectively, and the Ph.D. degree from the University of Tokyo in 2001. He is currently an Associate Professor of the Research Center for Advanced Science and Technology at the University of Tokyo. From 2003 to 2007 he was an Assistant Professor of the Department of Electronic Engineering at Shibaura Institute of Technology. From 2005 to 2006, he stayed in the Ecole Polytechnique Federale de Lausanne (EPFL) as a visiting researcher. His research interest is in the area of ubiquitous computing and sensor networks. He is a member of the IEEE, ION GPS, IEICE and IPSJ.

Hiroyuki Morikawa received his B.E. M.E. and Dr. Eng. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively. Since 1992, he had been in the University of Tokyo and currently a full professor of the Research Center for Advanced Science and Technology at the University of Tokyo. From 1997 to 1998, he stayed in Columbia University as a visiting research associate. From 2002 to 2006, he was a group leader of NICT Mobile Networking Group. His research interests are in the areas of computer networks, ubiquitous networks, mobile computing, wireless networks, photonic Internet, and network services. He served as a technical program committee chair of many IEEE/ACM conferences and workshops, Director of IEICE, Editor-in-Chief of IEICE Transactions of Communications, and sits on numerous telecommunications advisory committees and frequently serves as a consultant to government and companies. He received more than 20 awards including the IPSJ best paper award in 2006, IEICE best paper award in 2002 and 2004, Info-Communications Promotion Month Council President Prize in 2008, DoCoMo Mobile Science Award in 2009. He is a fellow of IEICE, and a member of IEEE, ACM, ISOC, IPSJ, and ITE.