# User Model and Utility Based Power Management

**Chih-Han Yu**[1], **Shie Mannor**[2], **Georgios Theocharous**[3], **and Avi Pfeffer**[1]
[1]School of Engineering and Applied Science, Harvard University, Cambridge, MA
[2]Department of Electrical and Computer Engineering, McGill University, Quebec, Canada
[3]Intel Research, Santa Clara, CA

## Introduction

Advances in hardware and wireless technology have made mobile devices ubiquitous in our daily life. Consequently, extending the battery life has become a major challenge needed to improve the usability of laptops. The purpose of a power management (PM) policy is to prolong a laptop's battery life while not affecting the system performance as perceived by the user. The optimal strategy is to turn off certain components when their services are not going to be needed and turning them back on just before they are needed. This uncertainty about the future is the core challenge in PM.

Most current PM techniques are based on timeout policies. These policies turn off a component if it hasn't been used for some predefined time. They are fairly simple and robust. However, these policies may be too fast or too slow to react. Recent research has addressed the importance of *adaptive* decision making for PM. This research can be roughly divided into two categories: direct prediction and stochastic optimization. The direct prediction approach (e.g. (Hwang & Wu 1997)) generally attempts to predict future idleness by associating current observable events with future idleness for PM policies. The stochastic optimization approach considers formulation of the system's state as stochastic processes, for example, a Markov model (Benini *et al.* 1999; Simunic 2002).

The main contribution of our work is to demonstrate the importance of incorporating a *user model* into adaptive PM. We use a Dynamic Bayesian Network (DBN) (Thomas & Kanazawa ) to capture the relationship between the *latent state of the user* and his/her observable activities. The DBN model is learned from the user's data and is therefore adapted to individual user. Besides, the future idle duration probability density functions (PDF) differ significantly when conditioned on different latent states. Based on the PDF associated with each individual latent state, our estimated future idle duration is more accurate. This information allows us to estimate the *expected power savings* and the *expected next service requested time*. By trading-off these two factors, the devised PM strategy is able to adjust to different level of aggressiveness. Furthermore, the PM decisions are also calibrated according to the latent states of different users.



## The System

There are three main entities: the user, the operating system (OS), and the hardware platform. The idleness prediction module measures inputs from these three entities and infers the user's future idleness. The future idleness estimation is used by decision modules. There are several sensors, e.g. mouse, keyboard, active applications..etc, as input for idleness prediction module. There are two available operating modes of the system: active mode (the machine is in full power state) and standby mode.

## Method

The structure of our DBN is shown as Figure 1. The variables at the top level, $s_t$ and $s_{t-1}$, are latent and represent the user's behavior of using the computer at time $t$ and $t-1$, respectively. The transition probability between $s_t$ and $s_{t-1}$ captures the tendency of the user to change his "mental state" between two time frames. Currently, our observable variables are: First, *recent idle events*. We consider the recent duration of idle period. We take a weighted sum, discounting older idle periods exponentially. Second, *current idleness*. How long the user has been idle from last indication of activity (keyboard or CPU activity). Third, *rush hours*. This is the variable that defines historically which hours the user tends to be idle longer. Finally, *mouse and keyboard activities*. The frequency of mouse and keyboard activities within a time window. The parameter learning is done using the Expectation Maximization (EM) algorithm.

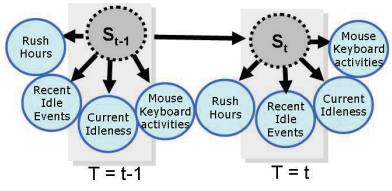



Figure 1: The Dynamic Bayesian Network structure used in our model.

**The Effect of the Latent State** Our hypothesis is that the latent state has a significant effect on the distribution of the

duration. We define the future idle length as *the duration between current time frame and next active state*. At each time point, we can query the latent state probability $p(s_t)$ from the DBN. Based on $p(s_t)$, the mental state of the user at each time point can be classified according to the maximal belief: $c_t = \max_i\{p(s_t = i)\}$, where $c_t$ is the cluster label (we chose $s_t$ as a binary variable, so $c_t \in \{1, 2\}$). For different values of the latent state we want to study the distribution $f_{c_t}(t) = Prob(t_{future} = t|c_t)$ where $c_t = \{1, 2\}$ and $t_{future}$ is the future idle time and $c_t$ is the latent state - either 1 or 2. We observe that the future idleness varies significantly according to the latent states. This verifies our hypothesis that the latent state is significant for understanding the idleness duration. It is worth noting that the future idle duration PDFs: $f_1(t)$ and $f_2(t)$ both are well approximated by a General Pareto Distributions (GPDs).

**Power Management Policy** We now describe our adaptive PM control scheme which is based on the current belief of the user's latent state. Our general approach is to switch machine to standby mode if the probability of moving to standby mode by mistake (i.e., having the user reverse the action) is smaller than a threshold. Namely, the PM policy is based on the estimated $\hat{f}_i(t), i = 1, 2$ and leads to policy: $\pi(t) =$ Standby, if $\sum_{i=1}^{n} \int_0^{\Delta} \hat{f}_i(\tau) \cdot p(s_t = i) \cdot u(\tau) \cdot d\tau \leq \rho$. Otherwise, it stays in active mode. The parameter $\Delta$ is the time point until there is no annoyance penalty if the system becomes active. The time duration which wrong decision leads to annoyance penalty is determined by estimating the user's utility function[1]. $n$ is the number of latent states. $u(\tau)$ is an exponentially decreased penalty function. We can infer the probability of the latent state $p(s_t = i)$ by querying the DBN. The integral term in the equation is the expected probability of system needs to be activated within time frame $t$ and $t + \Delta$ given our current belief on the user's latent state times penalty function. In other words, it measures the expected penalty of taking a PM action for $\Delta$ seconds at the time $t$. The decision threshold $\rho$ is used to determine the aggressiveness level of the power management policy.

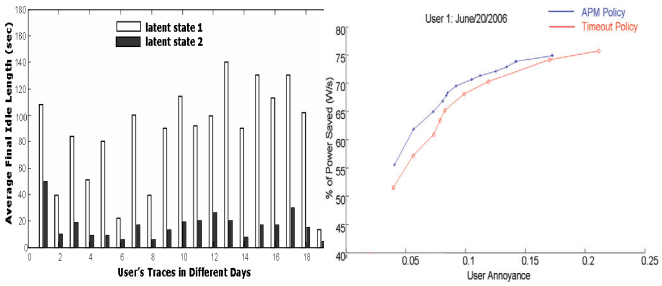
## Experimental Results

We present initial results of the proposed algorithm:

**User's Latent State v.s. Future Idle Duration** Figure 2(a) compares the average future idle duration between two different latent states in the testing data set. When our model believes the user is in latent state 1, the system usually ends up being in the idle state longer (empty bars in the graph). When the user is believed to be latent state 2, it usually leads to shorter idle period. This indicates that the latent state is a reasonable indicator of the future idle period.

**Learned PM Policies VS Timeout Policy** Figure 2(b) shows a sample ROC curve from one of our test subjects. We use three fold cross validation and use training set to learn both the DBN and the idle duration distribution parameters. We found that the DBN-based policies outperform the timeout policies within this user's annoyance range. This phenomenon is consistent across our test subjects.

[1]In our formulation, the parameters for utility function is estimated by formulating users' feedback as a linear programming problem.




(a) (b)

Figure 2: (a) User's Latent State v.s. Future Idle Duration. The x axis represents traces from different days and the y axis is average future idle duration under different latent states.(b) Power saving(y axis) v.s. annoyance(x axis) ROC curve for the DBN policies(upper curve) and timeout policies(lower curve).

## Conclusion and Future Work

The objective of user modeling in our work is to facilitate a control policy for PM in mobile computers. Our experimental results show that using the user model leads to improved performance. Our approach has the advantage that the PM policy designer can choose the desired annoyance level. This is in contrast to a previous work (Theocharous *et al.* 2006) where we considered a direct approach that tries to classify annoy/not annoy decision.

In our opinion, user modeling may be the key for successful adaptive power management for mobile computers. There are several challenges that are left for future research. First, the DBN was engineered manually. Instead, we can try to learn the structure by using structural EM. This might reveal interesting and non-trivial probabilistic structures. Second, we would to explore other methods to estimate user's utility function, e.g. generative approach. Third, the combination of additional power management actions such as turning the LCD off, turning the wireless radio off, and reducing CPU speed might lead to improved overall performance.

## References


Benini, L.; Bogliolo, A.; Paleologo, G. A.; and Micheli, G. D. 1999. Policy optimization for dynamic power management. *IEEE Transaction on Computer-Aided Design* 18(6):813–833.

Hwang, C.-H., and Wu, A. 1997. A predictive system shutdown method for energy saving of event-driven computation. In *Proceedings of the International Conference on Computer Aided Design*, 28–32. Morgan Kaufmann, San Francisco, CA.

Simunic, T. 2002. Dynamic management of power consumption. In Graybill, R., and Melhem, R., eds., *Power Aware Computing*. Kluwer Academic. chapter 6.

Theocharous, G.; Mannor, S.; Shah, N.; Gandhi, P.; Kveton, B.; Siddiqi, S.; and Yu, C. 2006. Machine learning for adaptive power management. *Intel Technology Journal* 10(4).

Thomas, D., and Kanazawa, K. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.