

An Empirical Study of the Potential for Context-Aware Power Management

Colin Harris and Vinny Cahill

Distributed Systems Group, Department of Computer Science,
Trinity College, Dublin 2, Ireland

{colin.harris,vinny.cahill}@cs.tcd.ie

<http://www.dsg.cs.tcd.ie>

Abstract. Context-aware power management (CAPM) uses context (e.g., user location) likely to be available in future ubiquitous computing environments, to effectively power manage a building's energy consuming devices. The objective of CAPM is to minimise overall energy consumption while maintaining user-perceived device performance.

The principal context required by CAPM is when the user is NOT USING and when the user is ABOUT TO USE a device. Accurately inferring this user context is challenging and there is a balance between how much energy additional context can save and how much it will cost energy wise. This paper presents results from a detailed user study that investigated the potential of such CAPM.

The results show that CAPM is a hard problem. It is possible to get within 6% of the optimal policy, but policy performance is very dependent on user behaviour. Furthermore, adding more sensors to improve context inference can actually increase overall energy consumption.

1 Introduction

With more and more computing devices being deployed in buildings there has been a steady rise in buildings' electricity consumption. These devices not only consume electricity but also produce heat, which increases loading on ventilation systems, further increasing electricity consumption. At the same time there is a pressing need to reduce overall building energy consumption. For example, the European Union's strategy for security of energy supply [1] highlights energy saving in buildings as a key target area¹. One approach to reducing energy consumption of devices in buildings is to improve the effectiveness of their power management.

Context-aware ubiquitous computing describes a vision of computing everywhere that seamlessly assists us in our daily tasks, i.e., many functions are intelligently automated. Information display, computing, sensing and communication will be embedded in everyday objects and within the environment's infrastructure. Seamless interaction with these devices will enable a person to focus on

¹ About 40% of total energy is used within the building sector.

their task at hand while the devices themselves vanish into the background. Realisation of this vision could exacerbate the building energy problem as more stationary computing devices are deployed but it could also provide a solution. Context information (e.g., user location information) likely to be available in such ubiquitous computing environments could enable highly effective power management for many of a building's energy consuming devices. We term such power management techniques as context-aware power management (CAPM), their principal objective being to minimise overall energy consumption while maintaining user-perceived device performance.

The *oracle* power management policy is a theoretical optimal policy that switches a device down to its low-power standby state the instant it is not being used and switches it back on just before the user requests its service again. Coming close to this oracle policy is a hard problem as it requires accurate prediction of the user's future intent. For example, the user has just left the room but will she be gone long enough to justify powering down the fluorescent lighting. The longer the policy waits to make the power down decision the more energy is being wasted. Then the user re-enters the room but is she going to use the desktop PC or is she just popping in to pick up some lecture notes. To develop effective CAPM policies that approach this ideal we need to obtain useful context from the user of the device.

We have identified several key requirements and designed a framework for CAPM. The principal context required for effective CAPM is (i) when the user is NOT USING for a sufficiently long period (see Section 2.1) and (ii) when the user is ABOUT TO USE a device. Accurately inferring this user context is the most challenging part of CAPM. However, there is also a balance between how much energy additional context can save and how much it will cost energy wise. To date there has been some research in the area of CAPM but to our knowledge there has been no detailed study as to what granularity of context is appropriate and what are the potential energy savings.

We have conducted an extensive user study to empirically answer these questions for CAPM of desktop PCs in an office environment. The sensors used are idle time based on keyboard/mouse input, user presence based on Bluetooth beaconing, near presence based on ultrasonic range detection, face detection, and voice detection. Results from the study show that coming close to the oracle is a hard problem and that performance is very dependent on user behaviour. For those who are HeavyUse users (i.e., use their PC greater than 85% of the time when they are in its vicinity) it is possible to get within 6-9% of the oracle policy and maintain user-perceived performance with a simple policy based on idle time and user presence. For LightUse users the median performance is 22-31% from the oracle and the standard deviation is large (21-28%). For these users a policy based on idle time, user presence, and near presence does better but incurs some user-perceived performance degradation. Beyond this face detection and voice detection consumed more than they saved.

The remainder of this paper is structured as follows. The following section briefly describes the background, requirements for CAPM and related work.

Section 3 describes our CAPM framework design and Section 4 describes the user study. Section 5 presents the results and evaluation. Finally, Section 6 concludes.

2 Background

CAPM is a dynamic power management technique. This section gives an overview of dynamic power management, the requirements for CAPM of stationary (i.e., plugged in) devices and a brief overview of related work.

2.1 Dynamic Power Management

Dynamic power management [2] is the term usually used to categorise techniques that dynamically power manage a device during its runtime operation. It is an effective technique that simply powers down a device (or some of its sub-components) during idle periods that occur during its operation. The two fundamental assumptions are that (i) idle periods will occur during the device's operation and (ii) these periods can be predicted with a degree of certainty. There is also a trade-off in powering down to save energy as, typically, making a power state transition has a significant cost. Possible costs are (i) extra energy is consumed in the power state transition, (ii) device performance is degraded as it becomes slower to respond, and (iii) device lifetime is reduced due to extra wear in powering down and up. Therefore a break-even time is defined as the minimum time a device must spend in a lower power state to justify the cost of transitioning down to that state and back again. For example, break-even time due to energy transition costs for a hard-disk may be around 10 seconds [3], break-even time due to lifetime for fluorescent lighting is around 5 minutes [4]. The resume time is the time taken for the device to resume to the operating state.

The current state of the art in dynamic power management is predominantly focused on developing policies for mobile computing devices. The key trade-off for these policies is increased battery life versus device performance. For example, a hard disk may be aggressively power managed to extend battery life (i.e., minimise energy consumption) but its performance will deteriorate as it will be slower to respond to user requests. Policies for mobile devices typically use low-level information such as the current idle time of the device. They can only predict short idle periods and are not able to predict the time of the next user request. Therefore, they are only suitable for managing devices or sub-components which have relatively short break-even and resume times (order 10 and 1 seconds respectively [3]). An example commonly used policy is the *threshold* policy which simply powers down the device when the idle time for the device is greater than a given threshold limit (e.g., after 10 seconds of inactivity, power down the hard-disk).

A number of issues relating to power management are already dealt with in most operating systems. Background processes can deny a power down request if they are performing a critical task such as file backup. Also, a process can set a

wake-up timer to resume the device at a given time to perform its task. Finally, Wake-on-LAN enables a device to be resumed via the network by another device which requests its service.

2.2 Requirements for CAPM of Stationary Devices

Requirements for power management of stationary devices are different to mobile device requirements. Typically, the most significant power savings for stationary devices are achieved by switching the entire device to standby. For example a typical desktop PC may consume 60 Watts (W) when on-idle and as low as 1W in standby. However, switching to a deep standby state has two implications as the device break-even and resume times are significantly longer. Firstly, since break-even times are longer, policies need to accurately predict longer idle periods (order 1 to 10 minutes). Second, switching to low-power standby states can cause significant user annoyance as resume times are longer (order 10 seconds) and there is the possibility of false power downs (i.e., powering down when the user is actually still using the device). Furthermore, stationary computing devices do not have battery limitations so users expect little or no performance degradation. Therefore, policies need to be near certain before powering down and they need to predict the time of the next user request to avoid resume-time delays. In order to predict longer idle periods and the time of next user request, policies need to use high-level user context.

We define context-aware power management as a dynamic power management technique that employs high-level user context to effectively power manage users' devices while maintaining user-perceived device performance. CAPM requires the accurate prediction of the context NOT USING for at least the given break-even period and ABOUT TO USE at least the resume time beforehand. However, achieving this accurate context has a cost energy wise.

For example, coarse-grained location/presence information can be a good cue for a person ABOUT TO USE their PC, if for example, their behaviour is that they always check their email when they re-enter the office. For other users it may not be a good cue, they may often pop in and out of their office without checking their email and without using the PC². Adding additional sensors to the scene may help us do better. For example, knowing the time of day could help in determining whether they are about to use their PC (e.g., they always check their email first thing in the morning, after lunch and at the end of the day). Time of day is cheap to obtain but does depend on the user being very regular in their behaviour. Adding other physical sensors could improve the situation, for example, to detect whether the user is alone or with others, where the user is in the space, or whether the user is moving towards the PC. These observations could possibly be made with additional sensors such as acoustic, video, object ranging, and more accurate location, to try to establish finer-grained context.

As the primary objective of CAPM is to minimise overall energy consumption, there is a bound on the granularity of context that is appropriate. Overstepping

² These users may check their email with a hand-held device.

this bound and the system will start consuming more sensor energy than the device energy it is saving due to the additional context. The additional consideration is that user-perceived device performance must also be maintained. To date there has been some research in the area of CAPM but to our knowledge there has been no detailed study as to what granularity of context is appropriate and what are the potential energy savings.

2.3 Related Work

Our previous work [5] evaluates the use of location as a key piece of context for CAPM of desktop PCs in an office environment. A simple location-aware policy was implemented that uses location context derived from detecting the user's Bluetooth-enabled mobile phone. This standby/wakeup on Bluetooth (SWOB) policy runs on the users PC. When the PC is on the policy polls for the user's phone via the Bluetooth discovery mechanism. If the phone is not found (after 5 discovery attempts) the PC powers down to standby. The PC is powered up again the next time the phone is found in range. Six user trials were performed, each over a period of a week.

The results from the user trials highlight that user behaviour significantly affects the performance of the policy. The results show two clear user types, HeavyUse users who use the PC a lot when in its 10-metre Bluetooth vicinity and LightUse users who use the PC occasionally when in its vicinity. The policy performs very well energy-wise for HeavyUse users (8% from oracle), whereas for LightUse users the performance deteriorates (>50% from oracle), consuming energy when the user is in the vicinity but not using the device. The user-perceived performance is good for both HeavyUse and LightUse users as the PC remains on while the user is in its vicinity. The threshold-5³ policy's user-perceived performance deteriorates significantly for LightUse users as the PC goes into standby many times while in the vicinity. The results demonstrate that coarse-grained location alone is not sufficient to determine the detailed user behaviour necessary for effective CAPM for all users.

Finer-grained context is needed to predict (i) the user in the vicinity but NOT USING the device and (ii) the user in the vicinity and ABOUT TO USE the device. The second scenario is difficult to achieve, as one key advantage of coarse-grained location is that it is a distant sensing device, i.e., it senses the user at a distance thereby enabling time for the device to resume before the user requests its use. Saving energy by switching off devices in the near vicinity of the user is difficult to achieve transparently. In this paper we examine in a detailed user study, how much better can be done than the SWOB policy with the use of finer-grained context.

Mozer's Adaptive House project [6] employs a sophisticated framework, ACHE, that has been developed over eight years of actual implementation and experimentation, to power manage devices in a house. ACHE is the most advanced CAPM application that we know of, employing in total 75 sensors monitoring

³ The threshold-5 policy initiates a power down after 5 minutes of idleness.

temperature, light, motion, sound, door and window positions, and weather. The techniques used are a neural network for mobility prediction, and a reinforcement learning technique for the decision making policy. The book chapter [7] concentrates on the issue of lighting control, the objective being to automate the setting of lighting levels within the house to maximise inhabitant comfort and minimise energy consumption.

The real-life experience from this project highlights the subtle requirements for effective power management. The two main discomforts the author experienced were the slow response of the system (due mainly to X10 communication delay) and the occasional false anticipation of zone entry. This caused switching of lights on in unoccupied zones. Results from the project show the cost of lighting and user discomfort values dropping over time as the system learns the user behaviour. However, the control of lighting level is based on user activity prediction, which currently is only inferred from the motion sensors. For example, the user being still for 5 minutes equating to the user `READING` and frequent zone change equating to `CLEANING HOUSE`. There is no evaluation as to how well the current activity classification works, but Mozer states that it is an area for future research.

Being able to infer these finer-grained user activities could increase the performance of the control decisions but it will also require the use of additional sensing and thereby increase energy cost.

Oliver et al. [8] present a system, `SEER`, that infers user activity from real-time streams of video, acoustic, and computer interactions in an office environment. The system is based on a set of layered hidden Markov models (LHMMs) that combine the readings from binaural microphones, a USB web camera, and a keyboard and mouse, to infer the activities, `PHONE CONVERSATION`, `FACE TO FACE CONVERSATION`, `ONGOING PRESENTATION`, `DISTANT CONVERSATION`, `Nobody in the office`, and `USER PRESENT ENGAGED IN SOME OTHER ACTIVITY`. The results claim a prediction accuracy of 99% for the system.

A further paper [9] presents an extension to `SEER`, `S-SEER`, which attempts to address the significant CPU usage of the feature processing algorithms. They say that, although the methods have performed well, a great deal of video and acoustic feature processing has been required by the system, consuming most of the resources available in a PC. They have developed an approach, expected value of information (EVI), which uses the principle of maximum expected utility to determine dynamically which are the most useful features to extract from the sensors in different contexts.

This is possibly the closest work to our research in the sense it uses similar sensors (video, audio, keyboard/mouse) in a similar office environment. They appear to achieve very good results for activity recognition but at a very high computational cost. Even with their EVI sensor selection policy the activity recognition algorithm consumes 33.4% of CPU time (on average). A typical desktop PC power consumes 60W when on and the CPU is idle. Running the activity recognition algorithm will increase the power consumption to about 80W. Increasing the power consumption of the PC by one-third is clearly too

costly energy wise. For CAPM it is necessary to find much more energy-efficient ways of accurately determining the context NOT USING and ABOUT TO USE.

3 CAPM Framework Design

The CAPM framework is composed of three main components: data capture and feature extraction, context inference, and decision making. The data capture and feature extraction layer captures the raw sensor events and in most cases preprocesses the data to obtain relevant features. For example, a video signal could be preprocessed to obtain features such as density of foreground pixels, density of motion, density of face pixels and density of skin colour [8]. A Bluetooth detection event could be preprocessed using a counter to count the number of times a Bluetooth tag has not been detected. This gives a history of when the tag was last seen, for example, a count of 10 means the tag has not been detected in the last ten polls. Sensors can include any physical sensing devices, power state changes of other devices, and "software sensors", such as time of day, or day of week.

The context inference layer is responsible for inferring more certain, higher-level context from the low-level sensor and feature data. The inference can involve combining redundant data to achieve more certainty and multi-modal data to infer context based on a combination of the multiple sensor modes. This is the most challenging part of the data processing as it is trying to infer high-level notions from low-level, noisy, and incomplete sensor data and/or their features. Bayesian networks are used to deal with this uncertainty in the data. This gives a level of confidence/probability in the proposition of a certain context being true. Furthermore, the Bayesian network can be trained to the specific user's usage pattern. The probabilistic inference of the contexts NOT USING and ABOUT TO USE is made available for use by the decision layer.

The decision layer is responsible for taking the power-management actions. For the initial version of the framework the decision layer was designed as a simple threshold rule. If the probability of NOT USING exceeded the given threshold (80%) then the action to power down was taken. Likewise, if the probability of ABOUT TO USE exceeded the given threshold (60%) the action to power up was taken.

3.1 Choice of Sensors

The CAPM framework was implemented for the power management of desktop PCs in an office environment. Users were given dedicated Bluetooth tags to attach to their key ring in order to achieve coarse-grained user location. The choice of additional sensors was motivated by the preference to use sensors that we believe will be part of future ubiquitous computing environments.

It is likely that a web camera and microphone will become integrated in the PC display as the popularity of video communication applications increases. Therefore we chose to use both of these sensors to capture finer-grained context.

For our system we chose a standard face detection algorithm [10] as the single feature to extract from the video stream and a basic voice activity detection algorithm [11] for the acoustic stream. The face detection only detects a face that is looking straight at the camera and the voice activity gives a measure from 0% to 100% of the level of voice activity detected in the environment.

The final sensor chosen was an ultra-sonic object range sensor in order to detect near presence of the user. We do not envisage this sensor as being part of a standard ubiquitous computing environment and so it would need to be installed as part of a CAPM solution. The motivation for choosing it was that it could possibly give good information at low energy cost, as there is no need for expensive feature processing.

3.2 Bayesian Network Design

We divided the problem into designing a distinct type of BN for each inference task, i.e., NOT USING for at least the break-even period and ABOUT TO USE at least the resume time beforehand. The structure of both types of BN is similar but they use different sensors and different sensor models. Development of the two types of BN was an iterative approach, starting with simple models and evolving the models to cope with issues that occurred. Fig. 1 details the final BN for the NOT USING (NU) context. The IsNotUsing node is the query node, we are asking what is the probability (P) that the user is not using the device for at least the given break-even time. This node has causal connections to each of the sensor nodes. This represents the fact that the user using (U) or not using the device is causing the observed sensor readings. The conditionality probability tables (CPTs) are initially set to default values and are subsequently learnt using supervised training data for each of the users. Given the full set of CPTs the model can reason diagnostically from the sensor values to give the probability of the user NOT USING the device. The actual value for each of the sensors is input into the BN and the probability of the context NOT USING is calculated. The power-down decision threshold was set at a probability of NOT USING greater than or equal to 0.8 (80%). This threshold figure was derived from trial and error experimentation with the model.

We also experimented with using a dynamic Bayesian network to capture possible temporal patterns but we found no significant improvement in performance.

4 User Study

The user study focuses on power management of desktop PCs in an office environment. We defined the population of the study to be all workers in an office environment that use a desktop PC. Our previous work [5] shows that the performance of the location-aware SWOB policy depends primarily on whether the user is LightUse or HeavyUse. A random sample of twenty users from the academic, administrative, technical and postgraduate staff and also office users from outside the college were selected for the trial. This ensured a wide range of different users and office types and layout. The trial length was set to run for 5

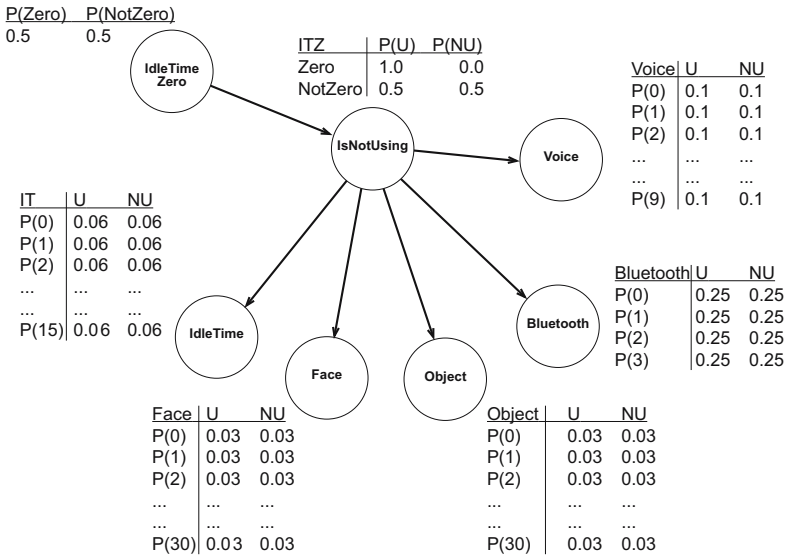


Fig. 1. Bayesian network for NOT USING context

working days. Based on initial results, we believed this would give enough data for training and simulation of the CAPM policies.

In order to compare the range of sensors and policies for each user’s usage trace it was necessary to collect all sensor data simultaneously as the user was using the PC during the trial and subsequently run policy simulations on the real usage data. Hence, the trial is broken into a data collection and processing phase, and a subsequent simulation phase. The simulation enables comparison of a range of CAPM policies, the SWOB policy, the optimal oracle policy, threshold policies and an always-on policy.

4.1 Data Collection and Processing

A set of Windows services were implemented to collect data for each of the sensors every 5 seconds; each value was date stamped and stored to file. Furthermore, all idle periods (i.e., periods when there is no mouse/keyboard events) of 30 seconds or more and all power events were logged to the Windows event log. The most difficult data to collect was the actual usage of the display and PC, i.e., when the user was actually USING or NOT USING these devices. To measure the actual usage, the NOT USING service attempts to power off the user’s display if the PC has been idle for greater than 60 seconds. The 60-second period of time attempts to balance accuracy against the experiment causing excessive user disruption. A message box appears asking the user if they are still using the PC; if they are the message box disappears by simply moving the mouse. This short mouse input is removed in the data processing as it would not have happened during normal device usage.

We imagined the effect that the NOT USING service may have on the users' behaviour was that they would tend to make more inputs than normal to stop the monitor being powered off. This would lead to a higher frequency of short idle periods than usual occurring in the usage trace. For this reason, idle period data was collected for several weeks after each trial and compared with the idle periods that occurred during the trial. This evaluation suggests that there was no significant effect from the trial.

The event logs are processed to create the measured device-usage trace, of when the user was USING and NOT USING the display or PC. Such a trace is graphed in Fig. 2 where the top-level line represents USING and the bottom-level line represents NOT USING the PC. The level below USING represents IDLE PERIODS that occurred when using the PC. There are periods in every trace where we don't know whether the user was using the PC or not. These are the idle periods before the service attempts to power down the display. The user may or may not be using the device at this time. These idle periods are scanned for face detection events. If a face is detected, we assume the state is USING (i.e., the user was looking at the display during this idle period). If a face is not detected, a DON'T KNOW state is inserted to indicate that we don't know whether the user was using the PC or not at this time. This DON'T KNOW state is represented by the line below the IDLE PERIODS state.

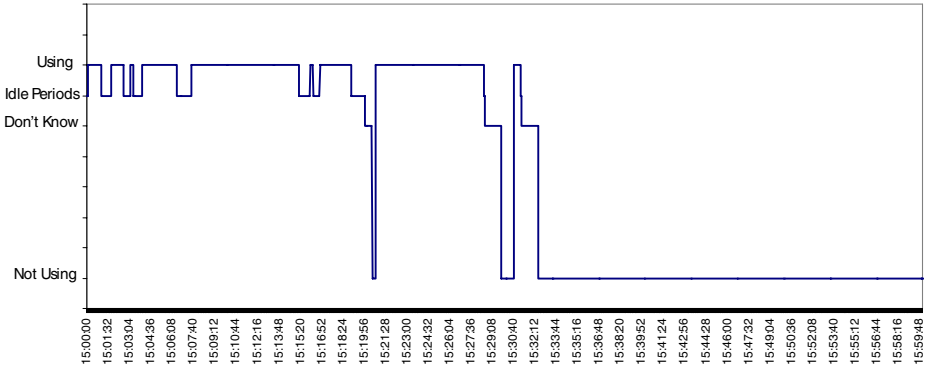


Fig. 2. Measured device-usage trace

4.2 Simulation of Policy Traces

All of the policy traces are generated from the measured usage trace. We assume the behaviour of all the users is good in that they power down their PC when leaving the office for the evening. So, for all traces the device is switched to off for the night time period. For all policies, the policy is allowed to power down in the DON'T KNOW and NOT USING states; attempting to power down when the user is still USING represents a false power down.

From the measured trace, it is straightforward to generate the estimated oracle policy trace and the always-on trace. The oracle policy trace is generated by placing the device in its standby state for all NOT USING periods that are greater than the device break-even time. The always-on trace is generated by leaving the device on for the duration of the day.

The SWOB policy was generated by running through the measured trace at a 5 second time step. The policy attempts to power down if the Bluetooth tag is not detected more than 5 times and the current idle time is greater than 60 seconds. If the measured usage is still in the USING state a false power down is reported, otherwise the policy powers down the device. The policy powers up when it detects the Bluetooth tag.

The range of threshold policy traces were generated by running through the measured trace at a 5 second time step. If the current idle time was greater than the given threshold, then the policy initiates a power down. There is no automated power up for the threshold policies.

The Bayesian CAPM policies required a learning stage and a simulation stage. To be rigorous, we employed a five-fold cross-validation strategy to the learning and simulation of the policies [12]. This involved training the model on one day of data and simulating for the other four days, and repeating this five times, training on each of the days. The resulting values were then estimated as the average of the five simulation results, giving a more robust analysis of the policies.

The Bayesian policies we chose to compare, were idle time (IT), IT-Bluetooth (IT-BT), IT-object range (IT-OR), IT-BT-OR, IT-BT-face detect (IT-BT-FD), IT-BT-OR-FD and IT-BT-OR-FD-voice activity (IT-BT-OR-FD-VA). This selection of models gives an increasing order of sensor granularity to enable comparison of each sensor's affect on the CAPM policy (see Table 1). Idle time was included in every model as it is also used in the SWOB and threshold policies, and Bluetooth was included in all but one model as it gives the coarse-grained user presence, which is the basis of the SWOB policy. We selected IT-OR as a special case to investigate the affect of only having the near presence information. Parameter learning of the models was carried out using the standard Spiegelhalter Lauritzen algorithm [13].

The set of BN policy traces were generated by running through the measured usage at a 5-second time step in both the power-down cycle and the power-up cycle. The power-down cycle operates when the device is on waiting to be powered down and the power-up cycle operates when the device is in standby. At each time step sensor values from the sensor records are applied to the BN power-down or power-up model and the model is updated to give the new probability of NOT USING or ABOUT TO USE. If the probability exceeds the threshold the policy initiates a power down or power up of the device. Fig. 3 shows the IT-BT BN policy powering down soon after the idle period begins.

The 7 Bayesian models plus the oracle, SWOB, six threshold policies and always-on policy resulted in 16 different policies to compare in total for each of the devices.

Table 1. Bayesian CAPM policies

IT	Idle time
IT-BT	Idle time, Bluetooth
IT-OR	Idle time, Object range
IT-BT-OR	Idle time, Bluetooth, Object range
IT-BT-FD	Idle time, Bluetooth, Face detection
IT-BT-OR-FD	Idle time, Bluetooth, Object range, Face detection
IT-BT-OR-FD-VA	Idle time, Bluetooth, Object range, Face detection, Voice activity

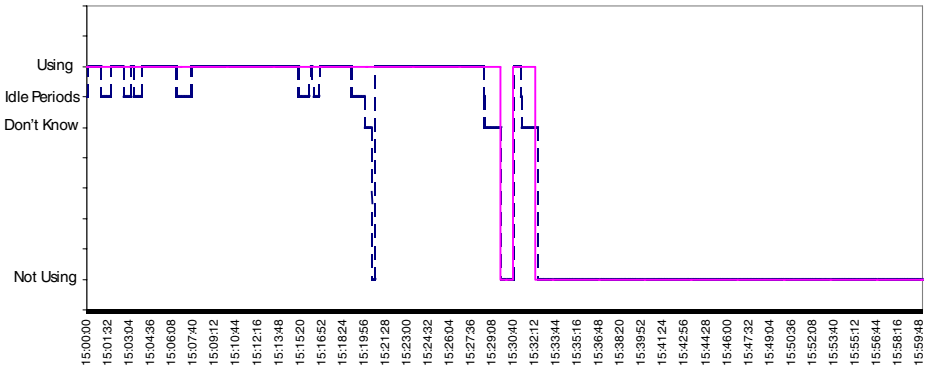


Fig. 3. IT-BT BN policy versus Measured device-usage trace

5 Results

Of the 18 user studies selected for analysis, 7 users were HeavyUse ($\leq 15\%$ NotUsing when in the vicinity) and the remaining 11 were LightUse users. The policies were run for both the display unit and the desktop PC. The energy consumption of the display unit is 45.8W when on and 1.8W in standby. The transition energy is assumed to be negligible. Its estimated break-even period is 1 minute and its resume time is 2 seconds. The PC’s break-even and resume times are significantly longer at 5 minutes and 7 seconds respectively. The energy consumption is 60.0W on and 2.8W in standby. The transition energy was measured to be 0.19Wh per transition.

The results first highlight the potential extra energy that can be saved for both the display unit and PC if we can do better than the simple location-aware SWOB policy. Results are given in terms of the actual Watt hours (Wh) of energy consumed per day and the percentage from the oracle policy.

5.1 Potential Extra Energy from SWOB

Fig. 4 shows the potential extra energy per day that could be saved for both LightUse and HeavyUse users of the display and PC. The potential extra energy is

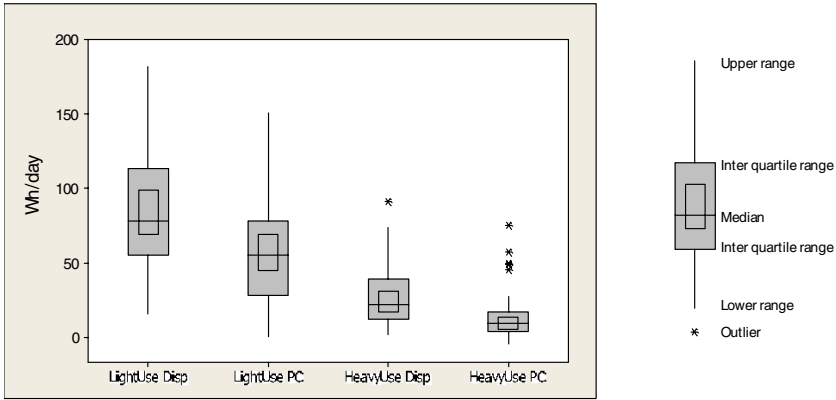


Fig. 4. Potential extra energy per day from SWOB

calculated as the difference in energy consumption of the SWOB policy and the oracle policy for each of the users. The box plots show the range, interquartile range, median and 95% confidence interval around the median for each of the groups.

The results show there is significantly more energy to be saved for LightUse users and also there is a larger variance in their energy consumption. For LightUse of the display, the median percentage from oracle is 31.1% (median 78Wh, range 20Wh to 185Wh). Similarly, for LightUse of the PC, the median percentage from oracle is 22.1% (median 59Wh, range 5Wh to 150Wh).

For HeavyUse of the display, the median percentage from oracle is 9.0% (median 22Wh, range 5Wh to 80Wh). For HeavyUse of the PC, the median percentage from oracle is 5.7% (median 13Wh, range 0 to 30Wh). Trying to do better than the SWOB policy for HeavyUse users is therefore difficult.

We next estimate the energy consumption of the sensors used by the CAPM policies.

5.2 Energy Consumption of Sensors

The power of each sensor was inferred by measuring the difference in the average power of the PC with and without each of the sensors running. The average power of the Bluetooth detection is 0.41W. This is based on the Bluetooth attempting to make a connection every 5 seconds and includes the energy cost of recharging the Bluetooth tag. The face detection’s average power is significantly higher at 2.47W⁴. Again the sample rate is every 5 seconds and the high power consumption is due to the large amount of processing needed to perform the face detection algorithm. The voice activity detection’s average power is 0.78W as its algorithm is less CPU intensive. Finally the object range detection’s average power is based on the average power of the sensor hardware, which was measured at 0.07W.

⁴ This is significantly less than the estimated 20W used by the feature processing in Oliver (see Section 2.3).

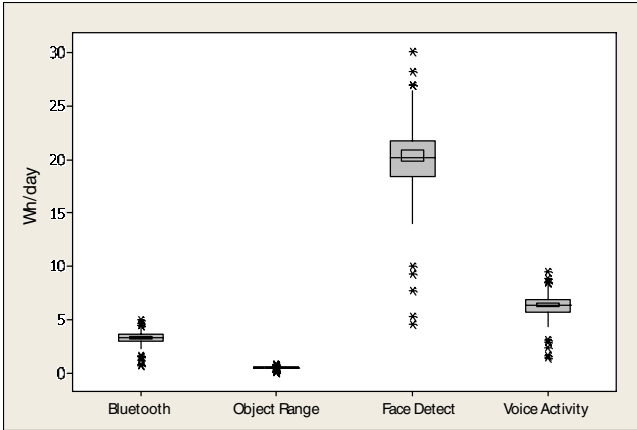


Fig. 5. Estimated sensor energy consumption per day

Fig. 5 shows the estimated energy consumption per day for each of the sensors. This is calculated as the average power of the sensor (at the sample rate) times the number of hours for each of the user's days. The median energy consumption per day for the sensors is BT 3.4Wh, OR 0.6Wh, FD 20.2Wh and VA 6.4Wh.

FD and VA consume a significant amount compared to the potential savings from the SWOB policy for the PC and display (LightUse - 59Wh and 78Wh, HeavyUse - 13Wh and 22Wh). It may be possible to decrease the sensor energy consumption by employing some form of power management for the sensors, but this is not explored here.

We next examine if using additional sensors and more sophisticated Bayesian CAPM policies can improve on the SWOB policy for both LightUse and HeavyUse users.

5.3 Comparison of Additional Sensors

The comparison is based on energy consumption and user-perceived performance, which is evaluated in terms of the number of false power downs (FPDs) that occurred in a day and the number of manual power ups (MPUs) that the user had to invoke per day.

Energy Consumption. Fig. 6 shows the extra energy consumed per day for each of the policies from the oracle for both LightUse and HeavyUse of the display unit. The plots highlight the SWOB policy performing well (close to the oracle) for HeavyUse users and significantly worse for LightUse users.

The policies closest to the oracle for LightUse are IT-OR 40Wh, IT-BT-OR 46Wh and threshold-5 46Wh (median 17.3% to 19.5% from oracle). For HeavyUse the lowest consuming policies are IT-OR 21Wh, IT-BT-OR 21Wh, SWOB 25Wh, IT-BT 25Wh and threshold-5 30Wh (median 8.9% to 13.1% from oracle).

User-Perceived Performance. Table 2 details the user-perceived performance of the CAPM policies in terms of median FPDs per day and median MPUs per day for all LightUse and HeavyUse users. The policies are ordered in ascending order of MPUs.

Table 2. User-perceived performance of policies in terms of median FPDs and MPUs per day for LightUse and HeavyUse users

Policies	LightUse		HeavyUse	
	FPDs	MPUs	FPDs	MPUs
SWOB	0	0	0	0
IT-BT	0.5	2	0	0.25
IT-BT-OR	0.75	4.75	0.5	1
IT-BT-FD	1.25	6.75	0.5	2
IT-BT-OR-FD	0.75	7	0.75	2.75
IT-BT-OR-FD-VA	0.75	7	0.75	2.75
IT-OR	0.25	8	0.25	7

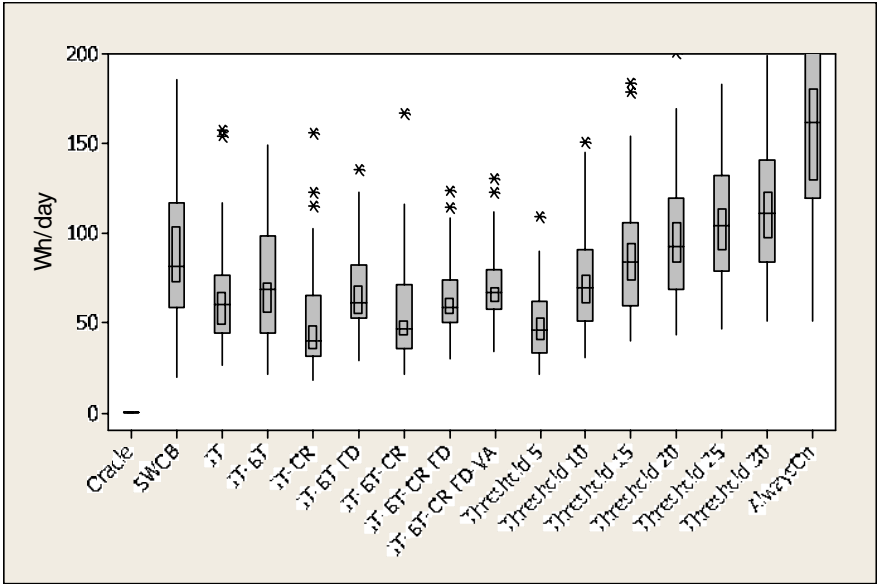
The best CAPM policy in terms of manual power ups for LightUse users is SWOB with median 0, then IT-BT (median 2) and IT-BT-OR (median 4.75) next. Since the power-up policy for all BN policies was limited to only powering up on BT, the more times the device was powered down when the user was in the vicinity, the more MPUs were incurred. The MPUs are less for HeavyUse users. This intuitively makes sense as HeavyUse users do not allow the device to power down as often, therefore requiring less power ups (SWOB median 0, IT-BT median 0.25, IT-BT-OR median 1).

The SWOB policy also performs best in terms of false power downs with no FPDs for both LightUse and HeavyUse users. The BN policies for LightUse are similar with median of 0.5 to 1.25 per day and range of 0 to 3 FPDs per day. There is no significant difference for HeavyUse users with median 0.25 to 0.75 and range of 0 to 5.

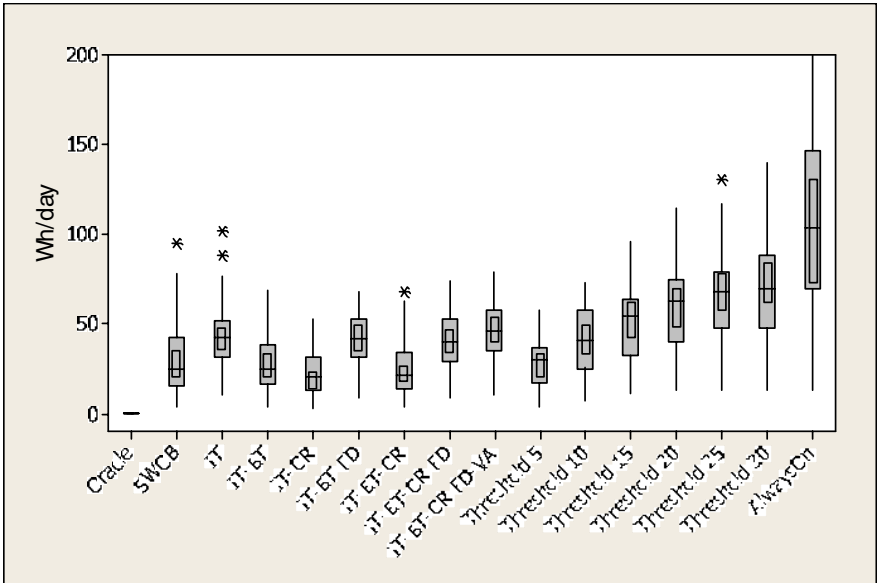
5.4 Summary

From these results for the display, it is clear that the best policy for HeavyUse users is the SWOB policy performing well both energy wise and user-perceived performance wise. On average the energy consumption is 9.0% from the oracle per day, there are median 0 FPDs and median 0 MPUs, resulting in very good user-perceived performance. The BN CAPM policies did not save significantly more energy and caused additional FPDs and MPUs.

For LightUse users the SWOB energy consumption is not as good and varies considerably across the users. It is equivalent to the threshold 15 to 20 policies and is on average within 31.1% of oracle. To do better energy wise, it seems we must accept some performance penalties in terms of false power downs and



(a) Light Use



(b) Heavy Use

Fig. 6. Extra energy consumption per day from the oracle (including sensor energy)

manual power ups. Of the BN policies, the IT-BT-OR policy is one of the lowest energy policies with the least MPUs and FPDs.

The overall pattern of results for power management of the PC are similar to that of the display. For HeavyUse the SWOB policy is again the clear choice with both low delta energy consumption 13Wh (5.7% from oracle) and no FPDs and few MPUs. The fact that the break-even time is longer means the policies achieve percentages closer to the oracle policy. The number of standby periods that were less than the break-even time were median 4 per day. This would be a concern for the lifetime decay of this PC.

For LightUse of the PC the IT-BT-OR policy has one of the lowest delta energy consumptions 30Wh (12% from oracle) with the least FPDs and MPUs (FPDs median 0.5, MPUs median 3.75 per day). The number of MPUs is significant as the resume time of 7 seconds will cause significant user annoyance. The SWOB policy delta energy consumption for LightUse is 59Wh (22.1% from oracle). This appears to be the best overall policy due to it having no false power downs and very few manual power ups (median 0).

In all cases the face detection sensor consumes significantly more energy than the object range detection sensor and does not provide significantly better information. The voice activity sensor information does not improve the IT-BT-OR-FD policy at all as the quality of its information for determining the context NOT USING is very weak.

6 Conclusions

Development of effective context-aware power management has an essential part to play in reducing overall building energy consumption. Achieving close to optimal performance is a hard problem as it requires accurate prediction of the user's future intent. Furthermore, performance varies significantly with user behaviour and there is a balance whereby adding more sensors to improve context inference can increase overall energy consumption.

The user study of desktop PCs shows that for HeavyUse users, a policy based on idle time and user presence (Bluetooth tag) performs well both energy and user-perceived performance wise. For LightUse users, a policy based on idle time, user presence and near presence (object range detection) performs well for the display, but may incur too many manual power ups for the PC, which has a long resume time. Therefore, for LightUse users of devices with long resume times it may be necessary to accept increased energy consumption to maintain user-perceived performance. To improve on this case, better sensor information is needed which has a low energy cost.

Finally, there is a concern of reducing device lifetime due to the policies not being able to predict long enough idle periods. Future work will look at including time-of-day information to reduce the number of short standby periods that are less than the break-even time (e.g., 5 minutes for the PC). If this is not sufficient, remote sensing may be necessary to detect user location throughout the building in order to predict these longer idle periods.

References

1. E.U.: Towards a european strategy for the security of energy supply (2000), http://europa.eu.int/comm/energy_transport/en/lpi_lv_en1.html
2. Benini, L., Bogliolo, A., Micheli, G.D.: A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 8(3), 299–316 (2000)
3. Douglass, F., Krishnan, P., Marsh, B.: Thwarting the power-hungry disk. In: *USENIX Winter*, pp. 292–306 (1994)
4. Tetri, E.: Profitability of switching off fluorescent lamps: Take-a-break. In: *RIGHT LIGHT 4*, vol. 1, pp. 113–116 (1997)
5. Harris, C., Cahill, V.: Exploiting user behaviour for context-aware power management. In: *International Conference On Wireless and Mobile Computing, Networking and Communications*, pp. 122–130. IEEE Computer Society Press, Los Alamitos (2005)
6. Mozer, M.: Adaptive house project, <http://www.cs.colorado.edu/~mozer/nnh/>
7. Mozer, M.: 12. In: *Smart environments: Technologies, protocols, and applications*, pp. 273–294. J. Wiley and Sons, Chichester (2004)
8. Oliver, N., Horvitz, E., Garg, A.: Layered representations for human activity recognition. In: *Fourth IEEE Int. Conf. on Multimodal Interfaces*, pp. 3–8. IEEE Computer Society Press, Los Alamitos (2002)
9. Oliver, N., Horvitz, E.: S-seer: Selective perception in a multimodal office activity recognition system. In: *Multimodal Interaction and Related Machine Learning Algorithms*, pp. 122–135 (2004)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition* (2001)
11. Rabiner, L., Juang, B.H.: *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)
12. Dietterich, T.: Statistical tests for comparing supervised classification learning algorithms. Technical report, Department of Computer Science, Oregon State University (1996)
13. Spiegelhalter, D.J., Lauritzen, S.L.: Sequential updating of conditional probabilities on directed graphical structures. *Networks* 20, 579–605 (1990)