A Conceptual Framework of Control, Learn, and Knowledge for Computer Power Management

Ria Candrawati[1], Nor Laily Hashim[2], MassudiMahmuddin[3], and Hammuzamer Irwan Hamzah[4]

[1]Universiti Utara Malaysia (UUM), riacandrawati@yahoo.com
[2]Universiti Utara Malaysia (UUM), laily@uum.edu.my
[3]Universiti Utara Malaysia (UUM), ady@uum.edu.my
[4]Universiti Utara Malaysia (UUM), zamer@uum.edu.my

ABSTRACT

This conceptual paper observes the human inactivity in computer power management and discovers that; the efficiency of the computer power management (CPM)can be achieved by the eligibility of the human inactivity period. This period reduces the efficiency of CPM. This study examines the self-adaptation(SA) and the knowledge repository (KR)concepts, to model the framework of a new approach in computer power management. The essential elements and features from theseconceptswere adapted and applied as a techniqueto a new implementation of CLK-CPM. As a result, this study has proposed a modelof thetheoretical framework and demonstratesit through its conceptual framework for the technique.

Keywords: Self-adaptive, knowledge repository, knowledge management, control loop, and reinforcement learning.

I INTRODUCTION

World population has increased as well as computer users and consequently, the power that was used has become one of the contributors to today's climate change. The needs to increase power management have become essential for many reasons;such as when the condition that causes additional power and higher bill, and what more, when the power state need to make the transition from active to idle and suddenly a user need to activate the state in a short idle period(Irani, Shukla, & Gupta, 2003).On the other hand, the power saving functions that are supplied by computer manufacturers are being used sufficiently; it causes the functionality of the power saving function to save power,is low, in other word, efficiency in computer power management (Hirao, Miyamoto, Hasegawa, & Harada, 2005).In addition, the power controlalways needs to be increased, to reduce power consumption and cooling process;moreover, reliability and compliance, with the environmental standards(Khargharia, Hariri, & Yousif, 2008).

Here, this study finds that, the human inactivity period reduces the efficiency in the CPM. Hence, this paper models a theoretical framework and demonstrates this framework with a conceptual framework technique. Therefore, this paper presents the background of this study, the adjustment of essential elements from the self-adaptation and the knowledge repository concepts, and finally, establishes the theoretical framework for CLK-CPM, together with the demonstration of the conceptual framework.

II BACKGROUND

In this section, this paper presents the background of the CPM, and how the problem for the human inactivity period happens. Due to this problem, several techniques for measuring power consumption forpersonal computers (PCs) have been proposed. These techniques can be categorized into different methods; hardware, software, minimizing the CPU usage, and also by the implementation of the algorithm based methods(Gupta & Singh, 2012).

First, hardware based approach, is an approach which is made by using an external hardware for monitoring activities of the system; and all devices that are attached to the computer. This approach can be made by using simulation, measurement-based estimation, and direct measurement.

Simulation is used to estimate the power consumption by analyzing low level models of the system. Although these methods are proved to be accurate(Trummer et al., 2009) yet, there are severallimitation found, such as:

- Simplescalar simulation would be slow and always requires a detailed design, which is not suitable for early stage exploration(Kim, Flautner, Blaauw, & Mudge, 2004).
- SoftWatt is unable to produce cycle-by-cycle estimates(Gurumurthi, Sivasubramaniam, Irwin, Vijaykrishnan, &Kandemir, 2002).
- FAST simulators also model full systems and can potentially predict power for such complete systems (Sunwoo, Al-Sukhni, Holt, & Chiou, 2007), but it is very time consuming (Trummer et al., 2009).

Secondly, software based approach is divided into three methods, which are predictive, stochastic and time-out process. All these three methods are known as Dynamic Power Management (DPM)(Trummer et al., 2009). DPM tries to achieve energy efficient computation by selectively turning off system components when they are idle. Furthermore, those three methods still have limitations, such as:

- Timeout approach is wasting on power waiting for the timeout to expire.
- Predictive approach cannot deal with general system models where multiple incoming requests can be queued before processing when the device is in off or sleep state.
- Stochastic approach is model dependent, memory and computation expensive when deal with non-stationary request.

Thirdly, minimizing the CPU usage is a systematic process of CPU usage metering and stated that the operating system manages all resources and keeps track of each process's resource consumption(Trummer et al., 2009). This method also updates the CPU time utilization of the process at every timer interrupt. This process covers CPU and its entire device, but it is not covering memory, chipset and bus controller.

Lastly, algorithm based approach, is a various proposed algorithms, to reduce the power consumption of a computer system, such as:

- Back-Off Algorithm(Das & Das, 2011) for networkapplication.
- Power Nap Algorithm (Meisner, Gold, &Wenisch, 2009) enables the system development onlow power state.
- Soft Watt algorithm (Gurumurthi et al., 2002) applied for measuring power consumption of a PC.

This study furthers the investigation by seeking the information of human inactivity, which increase the computer power usage, at time that which are described above. The following sub-section presents the human inactivity.

A. Human Inactivity

However, for hardware and software techniques, they are still adequately designed to handle the power wastage problem. Furthermore, the workload of a complex system is unpredictable, and there is alack of ability to sense human inactivity periods for computer system.Therefore, this situation could reduce the power consumption automatically without the need to set a fixed time for changing the computer state.

In addition, Gupta and Singh (2012) suggest future research direction, to develop effective solutions for minimizing power consumption of a computer system, by changing the OS's power schemes; using some intelligent schemes, to sense human inactivity. But, the available options in the power scheme are based on the time-out approach, and this approach is not sufficient to minimize power consumption. Here, this study finds that, the human inactivity periods can be controlled, learned, and the knowledge can be stored for future response.

As for that reason, this study attempts to implement the future research above by applying a new model of adapting human control loops into power management system. Control loops execute an automated process to collect the details it needs from the system, and then analyze those details to determine if something needs to be changed, continue with the plan, or sequence of actions, and perform those actions (IBM, 2005). Besides that, this study also incorporates reinforcement learning algorithm, by learning a new power control policy dynamically at runtime from the information it receives (Shen, Tan, Lu, Wu, & Qiu, 2013).

III SELF-ADAPTATION CONCEPT

Self-adaptation is an approach that able to adapt to changes in the execution environment and internal dynamics; such as response to failure, variability in available resources or changing priorities to continue to achieve their goals (Weyns et al., 2013). This approach is effective to overcome the complexity of current software system and response to uncertainty changes in conditions or requirements.

A. Control Loop

Self-adaptation comes with feedback control loops as a generic mechanism for keeping the system running by monitoring the current state, analyzing the information to detect the failure occur then determine how to solve the failure and execute the decision. Nevertheless, control loops is the core of self-adaptive as these loops adapt system behavior to keep objectives controlled based on either regulatory control, disturbance rejection or optimization requirements (Müller, Kienle, & Stege, 2009).The control loops consist of four activities: Monitor, Analyze, Plan and Execute (MAPE).

In 2003, Kephart and Chessintroduced the first architecture for self-adaptive that exposes the feedback control loops that perform identification functional components and interfaces for decomposing and managing the feedback loop called an autonomic element as in Fig. 1. This autonomic element is collaboration between (1) autonomic manager with feedback loops as the core activity,(2) managed element that is connected by the (3) sensor and effector that facilitate the collaboration and data and control integration among autonomic element. Over the years, self-adaptive system has been studied in different research areas of software engineering and also in many different perspectives of research communities (Brun et al., 2009).

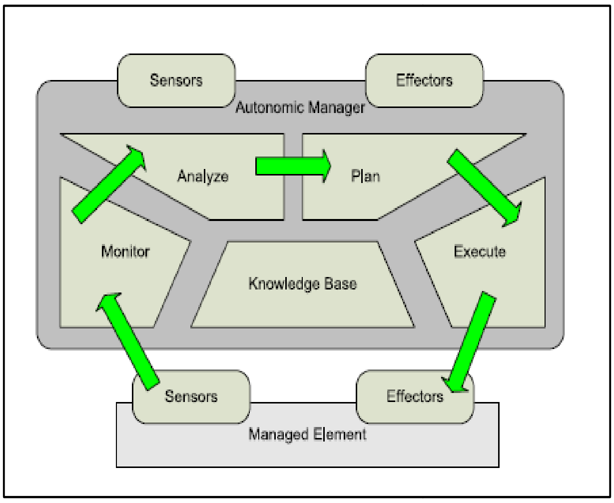

Fig. 1. Autonomic Element.

Hence, this study adopts the autonomic element solution, as the control loops function, to be an element for developing CLK-CPM. Besides that, to strengthen the decision making of the autonomic element, reinforcement learning concept is proposed to support CLK-CPM. The following sub-section provides the explanations for the reinforcement learning information.

1) Reinforcement Learning

This study examines the elements of reinforcement learning and adapts them in the CLK-CPM. Power management is a prediction problem. A power management policy is a procedure that takes decisions upon the state of operation of system components and on the state of the system itself (Benini, Bogliolo, & Micheli, 2000). A power management policy also known as a standard that needs to decide when to perform operation state transition and which transition should be performed (Jiang, Xi, & Yin, 2010). There are three well-known policies in power management, which are:

- Time-out stochastic (Lu & Micheli, 2001).
- Predictive (Hwang & Wu, 2000).
- Stochastic ((Tan & Qiu, 2008), (Qiu, Tan, & Wu, 2007), (Simunic, Benini, Glynn, & De Micheli, 2001)).

Moreover, Dhiman and Rosing(2006), also introduced learning algorithm to make power management decision at runtime, but this learningapproach cannot overcome the uncertainty condition. However, the reinforcement learning appears to counter the problem by learning a power control policy dynamically at runtime from the information it receives. It retrieves the system stateand adjusting the action when this state is re-visited next time, based on the reward/penalty received. This approach is appropriate for the scenario that involves dynamically changing task with time and user (Barto & Dietterich, 2004).

Here, this study finds that the reinforcement learning algorithm provides a solution to the problem of the uncertainty of human in loops. Therefore, this study plans, and models the appropriate framework for this issue to be explained. As for more information for searching the elements to develop CLK-CPM, this study examines the uncertainty in self-adaptation concept. The following sub-section presents the information for the uncertainty in self-adaptation.

B. Uncertainty in Self-Adaptation

However, uncertainty is a well-known challenge that the self-adaptive need to address. An uncertainty happens in every facet of adaptation and reason behind this problem is the controller and the system are loosely coupled and then introducing numerous sources of uncertainty (Cheng & Garlan, 2007). Esfahani and Malek(2013)categories uncertainty based on the relation to the spectrum of uncertainty, which are:

- Uncertainty due to the lack of knowledge related to Simplifying assumptions, Model Drift, Human in the loop, Objectives, Decentralization, and Cyber-physical.
- Uncertainty related to Noise, Parameters over time, and Context is due to the variability.

On the other hand, for further explanation, Fig. 2 summarizes existing methods for dealing the uncertainty challenges.

	Simplifying assumption	Model drift	Noise	Parameters over time	Human in loop	Objectives	Decentralization	Context	Cyber-physical system
Rainbow (Garlan et al., 2004)	√		√						
RELAX (Whittle et al., 2009)						√			
FLAGS (Baresi et al., 2010)						√			
FUSION (Elkhodary et al. 2010)	√	√						√	
ADC (Narayan and Satyanarayan, 2003)				√					
RESIST (Cooray et al., 2010)	√	√	√	√				√	
POISED (Eshafani et al., 2011)	√		√			√			

Fig. 2. Existing Approach for Uncertainty in Self-Adaptation.

In this study, the uncertainty of human in loops problem is investigatedby proposing a new model for adaptation of human in loop process. As this problem consistent with our motivation domain problem which is human inactivity,. Therefore, this study seeks the relation of the human in aloop from the self-adaptation concept, against the human inactivity problemto acquire the eligibility in achieving the efficiency of CPM. The following sub-section provides the categories of uncertainty in self-adaptation for further explanations.

1) Categories of Uncertainty in Self-Adaptation

There are many different sources for uncertainty; some sources come from internal system and another from the external system. This source of uncertainty has different characteristic, andEsfahani and Malek(2013)capture the uncertainty categories according to loose coupling between Meta level, user, base level and environment. Then, uncertainty categories discuss in the following:

a) Simplifying Assumption

This type of uncertainty occurs when the modeling abstractions become an inaccurate representation of the system and when an analytical model quantifying the system's response time may account for the dominant factors, such as execution time of components, and ignore others, such as the transmission delay difference between TCP and UDP. One of the reasons for inaccuracy is that sometimes the assumptions underlying the model are not held at runtime.

2) Model Drift

This uncertainty happens when the models that over time become wrong and do not represent the base-level subsystem correctly. If the base-level subsystem fails to enforce this change, the models used for reasoning by the meta-level subsystem will become inconsistent representation of the actual base-level subsystem.

3) Noise

This source due to sensor monitoring the available network bandwidth may return slightly different number every time a sample is collected, even if the real value of the bandwidth is fixed.

4) Parameters Over Time

The real changes in the monitored event cause this source of uncertainty. This category need to consider the behavior of the system in the future operation or the Adaptivity of this system cannot proceed as desired.

5) Human in Loop

Human behavior is inherently uncertain and it is turned creates uncertainty in the software system. Moreover, this uncertainty partially caused by a paradigm shift from software systems used merely as data processing entities deployed on isolated servers to becoming ubiquitous and engaging the users in their daily activities, yet new breeds of software usually depend on correct human behavior

6) Objectives

This source of uncertainty id reversing version of human in loop uncertainty which related to human's dependency on software. Eliciting user's preferences in terms of utility functions often have difficulty expressing their preferences and expectations using mathematical functions. Thus, the overall accuracy of such preferences remains subjective, making the analysis based on them prone to uncertainty.

7) Decentralization

The self-adaptive control creates a decentralized system, where the knowledge is scattered among the self-organization units comprising, and it is decentralized among different entities, which makes the system liable to the uncertainty.

8) Context

This uncertainty is representable in portable and embedded computing devices. In general, the performance of these software systems heavily depends on availability of the resources and it is subject to change as the context of execution changes. In contrast, sources of complexity introduced by the growing class of mobile and pervasive software, which are innately dynamic and unpredictable, and here the uncertainty occur when self-adaptive control is expected to detect the change in the context and adapt to behave appropriately

9) Cyber-Physical System

When computation continues to become cheaper and more widespread, software and physical spaces become increasingly intertwined yet tightly integrated, at this moment the physical world itself become an inherently uncertain. Uncertainty caused by the effect of the physical world on the software is a subset of context, which was described in the previous source. However, software can also affect the physical world, and this interaction can also host uncertainty.

IV KNOWLEDGE REPOSITORY CONCEPT

Knowledge repository is designed to store organized knowledge for future reuse (Gray &Durcikova, 2006)in order to promote deliberate knowledge sharing and reuse (Gray, 2001), to enhance the transfer of best practices, improve adaptation, and promote innovation through quicker access to new knowledge (Gray and Meister, 2004; Majchrzak et al., 2003).

Here, this study intends to adapt knowledge repository as a shared knowledge in proposed model to be a helpfulness approach ofdecision making process. The characteristic of this shared knowledge are to construct an organized memory of application activities captured, hardware's usage status and modeand to capture the explicit and exchanged activities of thecontrol loop.

V PROPOSED CLK-CPM

In this paper, this study analyzes the issues of human inactivity, and discovers that the human inactivity period can be reduced. There are several elements investigated in attaining the features to create a theoretical framework and the conceptual framework due to the solution of this problem. For easier understanding, the equation (1) provides the function of how the CLK-CPM is obtained.

$$CLK\text{-}CPM = ((CL \cup RiL) \cap KR) \qquad (1)$$

This situation is also visualized using a set diagram; Fig. 3 shows the union of CLK-CPM. The following sub-sections present the development of the theoretical framework and the conceptual framework for CLK-CPM.

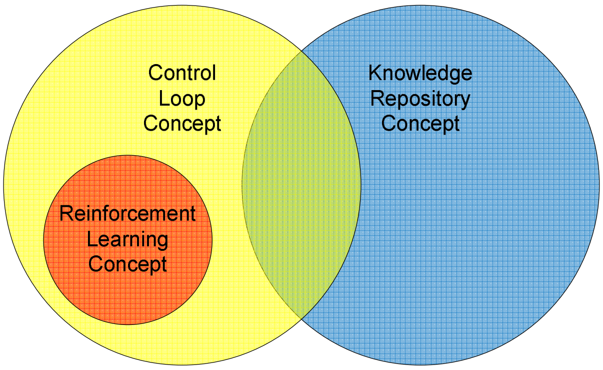

Fig. 3. CLK-CPM in the Domains' Concept.

A. Theoretical Framework

This proposed model is a hybrid of the two main concepts which are self-adaptation and knowledgerepository concepts to be applied in the CLK-CPM that runs the adaptation human control loops. This framework is shown in Fig. 4.

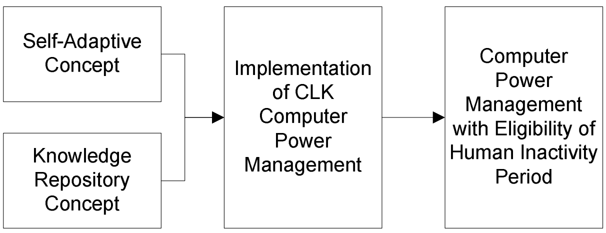

Fig. 4. Theoretical Framework for CLK-CPM.

B. Conceptual Framework

To solve the problem that appears in the motivation domain which is no intelligent control to sense the human inactivity, this study adopts with human in loop control as a solution. This paper introduces the CLK-CPM as the solution to reduce the human inactivity period for computer power management. Fig. 5depicts the conceptual model of CLK-CPM. The CLK-CPM counters the issue by adapting the MAPE-K loop, and attaches with another knowledgerepository outside from the main loop, that stores not only the history of all transactions in the loop, but also all the history of the parameters from the hardware and application logs. This second knowledgerepository is used for decision support. Furthermore, the CLK-CPM collaborates with reinforcement learning, as the processor, or brain, in the planning phase, in the control loop, to produce solutions for any symptoms that occur.

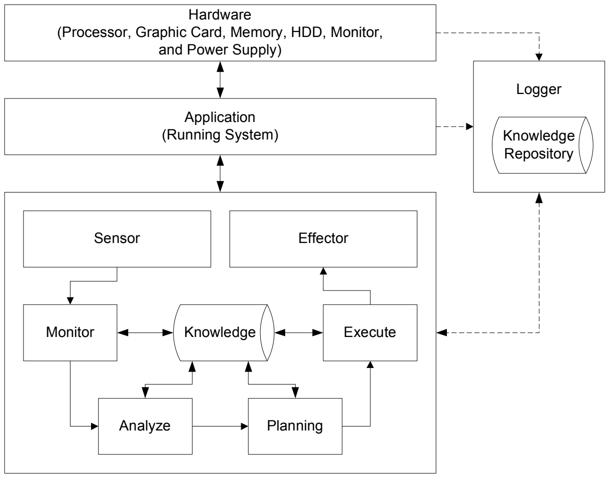

Fig. 5. Conceptual Framework for CLK-CPM.

Each component of the conceptual framework for CLK-CPM is defined in the definitions listed below:

- Hardware component supplies the parameters used while application is running or idle. They are processing speed, memory load, power usage, etc.
- The Application component passes the current mode to the sensor.
- The Sensor captures the current state of application and hardware parameters.
- The Monitoringcomponent measures the parameters and application mode then senses the inactivity symptoms.
- The Analysis component analyzes the value from the Monitoring phase. In this function, all the learning process will be captured in the knowledge part as a support factor for future prediction action.
- The Planning component performs the Q-Learning algorithm using the priorknowledge stored to decide what the optimal stage should maintain.
- The Executing component executes when the planning function is given the task to optimize the power usage when the usage achieve the warning point and also to activate, sleep or idle the computer based on the timeout timer that will determine based on learning history in theknowledge part.
- The Effector componentwill react to the application and hardware then counter the effects of the changes by changing the system and maintaining the environment

- The Knowledgecomponentstores the resources details and also the result of all process from loop progress.
- Shared Knowledge components records the parameters from hardware, mode state from application and execution history of loop to a disk file

With that, this paper establishes the CLK-CPM as a solution to reduce the human inactivity period in the computer power management.

VI CONCLUSION

As a conclusion, this study has analyzed and observed the eligibility of the human inactivity period issues from the dynamic power management domain, especially in the computer system. This study investigates, and adapts the elements from the self-adaptation and the knowledge repository concepts in creating the new theoretical framework for CLK-CPM. Besides that, this paper also demonstrated the implementation of the proposed conceptual framework for the CLK-CPM. As for the future work, this study will try to develop the CLK-CPM and perform effectiveness evaluation.

REFERENCES

Barto, A., & Dietterich, T. (2004). Reinforcement learning and its relationship to supervised learning. *Handbook of Learning and Approximate Dynamic Programming*. Retrieved from http://web.engr.orst.edu/~tgd/publications/Barto-Dietterich-03.pdf

Benini, L., Bogliolo, A., & Micheli, G. De. (2000). A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *8*(3), 299–316. doi:10.1109/92.845896

Brun, Y., Di, G., Serugendo, M., Gacek, C., Giese, H., & Shaw, M. (2009). Engineering Self-Adaptive Systems through Feedback Loops, 48–70.

Cheng, S., & Garlan, D. (2007). Handling uncertainty in autonomic systems. *ASE'07*. Retrieved from http://acme.able.cs.cmu.edu/pubs/uploads/pdf/IWLU07-HandlingUncertainties-pub.pdf

Das, D., & Das, D. (2011). Back-off algorithm based power saving mechanism in a multi-RAT UE. *2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application*, 1–6. doi:10.1109/IMSAA.2011.6156346

Dhiman, G., & Rosing, T. (2006). Dynamic Power Management Using Machine Learning. In *2006 IEEE/ACM International Conference on Computer Aided Design* (pp. 747–754). Ieee. doi:10.1109/ICCAD.2006.320115

Esfahani, N., & Malek, S. (2013). Uncertainty in self-adaptive software systems. *Software Engineering for Self-Adaptive Systems II*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-35813-5_9

Gray, P. H., &Durcikova, A. (2006). The Role of Knowledge Repositories in Technical Support Environments : Speed Versus Learning in User Performance. *Journal of Management Information System*, *22*(3), 159–190.

Gupta, P. K., & Singh, G. (2012). Minimizing Power Consumption by Personal Computers: A Technical Survey. *International Journal of Information Technology and Computer Science*, *4*(10), 57–66. doi:10.5815/ijitcs.2012.10.07

Gurumurthi, S., Sivasubramaniam, A., Irwin, M. J., Vijaykrishnan, N., & Kandemir, M. (2002). Using complete machine simulation for software power estimation: The softwattapproach. *High-Performance Computer Architechture (HPCA'02)*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=995705

Hirao, E., Miyamoto, S., Hasegawa, M., & Harada, H. (2005). Power Consumption Monitoring System for Personal Computers by Analyzing Their Operating States. In *Environmentally Conscious Design and Inverse Manufacturing, 2005. Eco Design 2005. Fourth International Symposium* (pp. 268–272). doi:10.1109/ECODIM.2005.1619220

Hwang, C.-H., & Wu, A. C.-H. (2000). A predictive system shutdown method for energy saving of event-driven computation. *ACM Transactions on Design Automation of Electronic Systems*, *5*(2), 226–241. doi:10.1145/335043.335046

IBM. (2005). *An architectural blueprint for autonomic computing* (pp. 1–34).

Irani, S., Shukla, S., & Gupta, R. (2003). Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Transactions on Embedded Computing Systems*, *2*(3), 325–346. doi:10.1145/860176.860180

Jiang, Q., Xi, H.-S., & Yin, B.-Q. (2010). Adaptive optimisation of timeout policy for dynamic power management based on semi-Markov control processes. *IET Control Theory & Applications*, *4*(10), 1945. doi:10.1049/iet-cta.2009.0467

Kephart, J. O., & Chess, D. M. (2003). The Vision of Autonomic Computing, (January), 41–50.

Khargharia, B., Hariri, S., & Yousif, M. S. (2008). Autonomic power and performance management for computing systems. *Cluster Computing*, *11*(2), 167–181. doi:10.1007/s10586-007-0043-6

Kim, N., Flautner, K., Blaauw, D., & Mudge, T. (2004). Circuit and microarchitectural techniques for reducing cache leakage power. *Very Large Scale Integration (VLSI) Systems*, 167–184. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1266406

Lu, Y.-H., & Micheli, G. De. (2001). Comparing system level power management policies. *IEEE Design & Test of Computers*, *18*(2), 10–19. doi:10.1109/54.914592

Meisner, D., Gold, B. T., & Wenisch, T. F. (2009). PowerNap : Eliminating Server Idle Power. *Architectural Support for Programming Languages and Operating System*, 205–216.

Müller, H., Kienle, H., & Stege, U. (2009). Autonomic computing now you see it, now you don't. *Software Engineering*, 32–54. Retrieved from http://www.nature.com/nature/journal/v435/n7046/full/4351165a.html

Qiu, Q., Tan, Y., & Wu, Q. (2007). Stochastic Modeling and Optimization for Robust Power Management in a Partially Observable System, 779–784.

Shen, H., Tan, Y., Lu, J., Wu, Q., & Qiu, Q. (2013). Achieving autonomous power management using reinforcement learning. *ACM Transactions on Design Automation of Electronic Systems*, *18*(2), 1–32. doi:10.1145/2442087.2442095

Simunic, T., Benini, L., Glynn, P., & De Micheli, G. (2001). Event-driven power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *20*(7), 840–857. doi:10.1109/43.931003

Sunwoo, D., Al-Sukhni, H., Holt, J., & Chiou, D. (2007). Early Models for System-Level Power Estimation. *2007 Eighth International Workshop on Microprocessor Test and Verification*, 8–14. doi:10.1109/MTV.2007.8

Tan, Y., & Qiu, Q. (2008). A Framework of Stochastic Power Management Using Hidden Markov Model, 92–97.

Trummer, C., Kirchsteiger, C., Steger, C., Weiß, R., Pistauer, M., & Dalton, D. (2009). Specification and automated simulation-based verification of power requirements for system-on-chips. *Circuits and Systems and TAISA Conference*, *NEWCAS-TAI*(IEEE North-East Workshop), 1–4. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5290475

Weyns, D., Schmerl, B., Grassi, V., Malek, S., Mirandola, R., Prehofer, C., … Goschka, K. (2013). On patterns for decentralized control in self-adaptive systems, 76–107. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-35813-5