STATUS OF THESIS

Title of thesis: PC Power Management Based on Application Idle Time and Power Transitioning Event Profiling

I CHAN PISETH

hereby allow my thesis to be placed in the information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☒ Non-confidential

If the thesis is confidential, please state the reason:

____

The contents of the thesis will remain confidential for __________ years.

Remarks on disclosure:

____

Endorsed by

Alan Oxley
Professor
Computer & Information Sciences
Universiti Teknologi PETRONAS

Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
No.48B st, 376, BoeungKenkangIII, Chamkarmon,Phnom Penh, Cambodia.	Prof. Dr. Alan Oxley
Date: 11/7/13	Date: 11/7/13

UNIVERSITI TEKNOLOGI PETRONAS

PC POWER MANAGEMENT BASED ON APPLICATION IDLE TIME AND POWER TRANSITIONING EVENT PROFILING

by

CHAN PISETH

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Alan Oxley
Professor
Computer & Information Sciences
Universiti Teknologi PETRONAS

Signature: ______________________

Main Supervisor: Prof. Dr. Alan Oxley

Signature: ______________________

Head of Department: Dr. Jafreezal Bin Jaafar

Dr. Jafreezal Bin Jaafar
Head
Dept. of Computer & Information Sciences
Universiti Teknologi PETRONAS

Date: ______________________

PC POWER MANAGEMENT BASED ON APPLICATION IDLE TIME AND POWER TRANSITIONING EVENT PROFILING

by

CHAN PISETH

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SRI ISKANDAR

PERAK

JULY 2013

DECLARATION OF THESIS

Title of thesis	PC Power Management Based on Application Idle Time and Power Transitioning Event Profiling

I CHAN PISETH

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Alan Oxley
Professor
Computer & Information Sciences
Universiti Teknologi PETRONAS

Signature of Author

Signature of Supervisor

Permanent address:

No.48B, st, 376, BoeungKenkang III, Chamkarmon,PhnomPenh, Cambodia.

Date: 11/7/13

Name of Supervisor

Prof. Dr. Alan Oxley

Date: 11/7/13

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Dr. Alan Oxley and co-supervisor Dr. Low Tan Jung, for their continuous support throughout my research. Anytime I needed their guidance and motivation, they were always there for me.Secondly, I would like to thank my parents, my brothers, and relatives for their motivation, care, and love.

I also want to thank Mr. Ruslan Idris and Information Technology Media Service Department's staffs for their help in accelerating the process of requesting labs for conducting the experiment. Furthermore, I would like to extend my gratitude to the lecturers of Computer and Information Sciences. It was a pleasure working with them. I knew I could rely on them when I needed help or advice.

Lastly, I want to thank all my friends and co-workers who I have met during my time in Universiti Teknologi PETRONAS. They helped to make my workday more enjoyable.

ABSTRACT

Green Information Technology (IT) has recently become one of the main focuses in research and practice. Its primary goal is to reduce or optimize power consumption, and to address the issue of large power wastage in many organizations. Computer power management is part of the green IT that can help to save computer power consumption. Consequently, it is helping organizations to increase profits and to reduce environmental impact. This research contributes to the green IT by proposing a mechanism to efficiently manage the computer power consumption through a computer power management application. The objective of this research is to explore factors surrounding computers, which may affect its power consumption. This research was accomplished by means of interview, observation, and activity monitoring in the Universiti Teknologi PETRONAS (UTP) campus.

Two major components, the power transitioning event monitoring component and the application idle time monitoring component are proposed to enhance the PC's built-in power management tool, which is traditionally based on the static timeout setting. Automatic monitoring services are deployed to log power transitioning events and application idle time in the PCs. The data thus collected are used to draw meaningful conclusion on user's behavior to set the timeout threshold for shutting down the computer. All monitoring tasks and timeout threshold setting generation are done dynamically as a background processing job. The user's foreground processes or applications will not be affected to avoid irritating the user. A prototype application software based on the proposed components was implemented in a practical computer lab installed with 40 desktops. Data were collected over a period of three months. The results collected from this lab revealed that an average of 74 percent of the computer's idle time which consuming power wastefully was reduced, with an average of 85.8 percent of user idleness prediction accuracy, and a low level of user irritation.

ABSTRAK

Teknologi Maklumat Hijau (TM) baru-baru ini menjadi salah satu fokus utama dalam penyelidikan dan amalan. Matlamat utamanya adalah untuk mengurangkan atau mengoptimumkan penggunaan kuasa elektrik, dan menangani isu pembaziran besar kuasa elektrik dalam banyak organisasi. Pengurusan kuasa elektrik komputer adalah sebahagian daripada TM hijau yang boleh membantu untuk menjimatkan penggunaan kuasa elektrik komputer. Oleh itu, ia membantu organisasi untuk meningkatkan keuntungan dan mengurangkan kesan alam sekitar. Kajian ini menyumbang kepada TM hijau dengan mencadangkan mekanisme untuk menguruskan penggunaan kuasa elektrik komputer dengan cekap melalui aplikasi pengurusan kuasa elektrik komputer. Objektif kajian ini adalah untuk meneroka faktor-faktor sekitar komputer yang boleh menjejaskan penggunaan kuasa elektrik. Kajian ini telah dicapai melalui temu bual, pemerhatian, dan pemantauan aktiviti dalam kampus UniversitiTeknologi PETRONAS (UTP).

Dua komponen utama adalah, komponen pemantauan peralihan kuasa dan aplikasi komponen pemantauan masa terbiar yang dicadangkan untuk meningkatkan PC terbina dalam alat pengurusan kuasa elektrik, yang secara tradisinya berdasarkan penetapan masa tamat statik. Perkhidmatan pemantauan automatik digunakan untuk merekod peristiwa peralihan kuasa elektrik dan aplikasi masa terbiar di PC. Data-data itu dikumpul dan digunakan untuk membuat kesimpulan yang bermakna ke atas tingkah laku pengguna untuk menetapkan ambang masa tamat untuk menutup komputer. Semua tugas-tugas pemantauan dan penetapan ambang masa tamat dilakukan secara dinamik sebagai tugas pemprosesan latar belakang. Proses latar depan pengguna atau aplikasi tidak akan terjejas untuk mengelakkan menjengkelkan pengguna. Satu prototaip perisian aplikasi berdasarkan komponen yang dicadangkan dilaksanakan di makmal komputer yang dipasang dengan 40 desktop. Data telah dikumpulkan dalam tempoh tiga bulan. Keputusan diambil dari makmal ini

mendedahkan bahawa purata 74 peratus daripada masa terbiar komputer yang membazirkan penggunakan kuasa elektrik telah dikurangkan, dengan purata 85.8 peratus daripada ketepatan ramalan pengguna dengan komputer terbiar dan tahap kejengkelan pengguna yang rendah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACPI	Advanced Configuration and Power Interface
APM	Advance Power Management
API	Application Programming Interface
BIOS	Basic Input Output System
BOINC	Berkeley Open Infrastructure for Network Computing
CMA	Cumulative Moving Average
CO2	Carbon Dioxide
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DPMaitp	Display Power Management based on Application Idle Time Profiling
DPMS	Display Power Management Signaling
EMA	Exponential Moving Average
GPU	Graphical Processing Unit
HID	Human Interface Device
IT	Information Technology
MA	Moving Average
PMaitptep	Power Management based on Application Idle Time and Power Transitioning Event Profiling
RAM	Random Access Memory
SMA	Simple Moving Average
SONAR	Sound Navigation and Ranging
WMA	Weighted Moving Average
WOL	Wake-On-LAN

CHAPTER 1

INTRODUCTION

1.1. Background

Fueled by population and economic growth, large amount of natural resources have been extensively extracted. In today's advance world, electricity cannot be detached from people. However, it is not self-generated and it is not free. Nuclear, fossil fuel, and natural gas are the main resources to generate the electricity. Some resources are unsustainable and some give negative impact to the environment. For instance, the burning of fossil fuels such as oil and coal, which were reported by British Petroleum (BP) statistical review of world energy report still to be the biggest energy supply for the world in 2013, emits large amounts of carbon dioxide (CO_2) to the atmosphere [1], [2].

CO_2 is a potential element causing global warming. In general, global warming causes the temperature to rise, melting the ice cap in the Arctic and Antarctic resulting in the flooding at the coastal regions [3]. On top of that, it has a catastrophic impact on the global hydrological cycle which could lead to more drought, hurricanes, and storm [4], [5]. Even though there is a positive sign in some of the most advanced and developed nations such as United States on the efforts to limit her emission, some articles have reported that the efforts have failed. Looking at another side of the world, a rapid emission is booming in developing countries, where is a new target for manufacturers from developed countries because of cheaper labor and full of natural resources [6].

IT infrastructure has been known to be one of the main energy consumers, estimated to consume between 2 percent to 10 percent of worldwide energy consumption [7]. Ever increasing use of electronic equipment in many organizations escalates the power consumption. In this modern world, power consumption in the organizations is becoming one of the main emphases because most of the tasks to be accomplished required assistance from electronic devices such as handheld devices and computers. Thus, this is where green information technology comes into place.

The term "Green Information Technology" has been referred to as the practice of reducing adverse environmental effects by using information technology. It is also known as a method to use technology more effectively and efficiently. Green information technology has recently become one of the main focuses on research and practice, with its primary goal of reducing power consumption as a result of the large amount of power consumption and unintended wastage in the organizations.

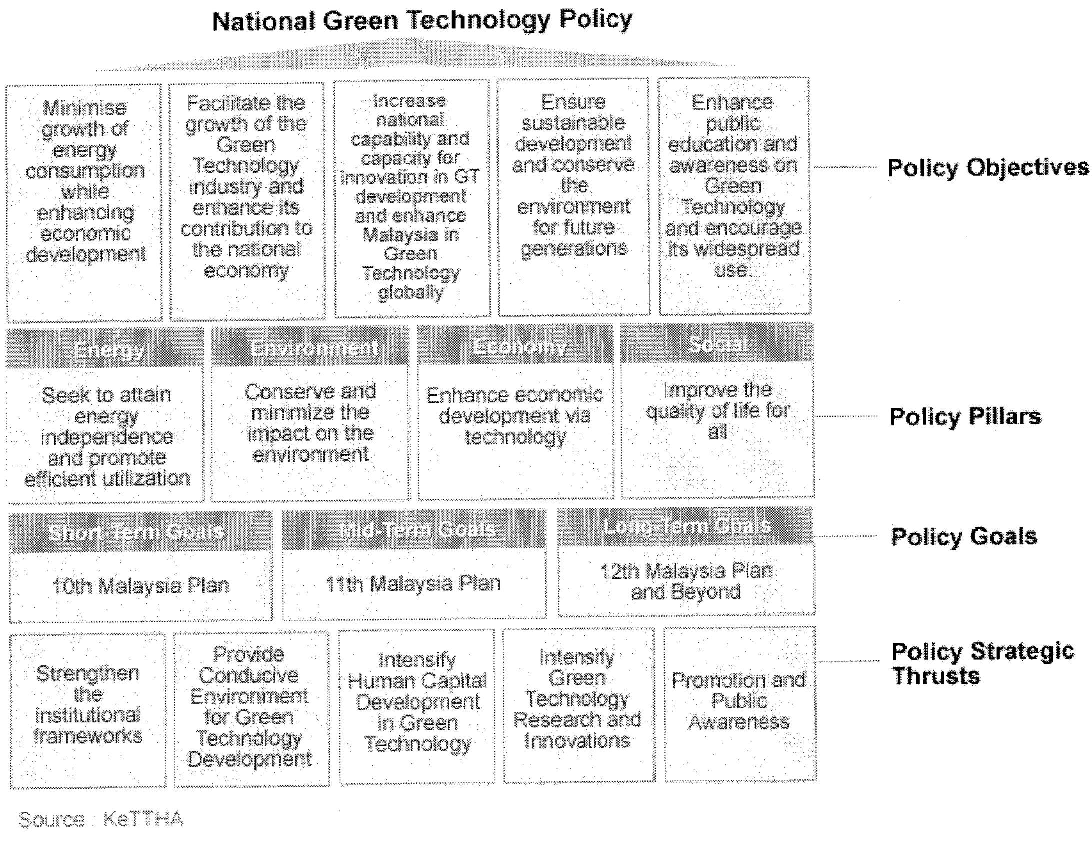


Figure 1.1. Malaysian national green technology policy [8].

Malaysia is one of the nations considering going for green technology to lower down her carbon footprint and to reduce the amount of power consumption for the whole country. For instance, The National Green Technology Policy was launched by the Honorable prime minister of Malaysia on 24 July 2009 [9]. It focuses on four pillars: energy, social, economic, and environment along with a clear objective, goal, and strategic thrusts as shown in Figure 1.1. Furthermore, six main sectors are mentioned in Malaysia's green technology roadmap: energy, building, transportation, water and solid waste management, manufacturer, and information technology [10].

At the fifteenth conference of parties (COP15) in Copenhagen, the honorable Dato Sri Mohd Najib Tun Abdul Razak, prime minister of Malaysia, announced the Malaysia's proposal to reduce its CO_2 emission up to 40% by the year 2020 as compared to its 2005 level [11]. Figure 1.2 illustrates the statistics of sources to generate power and demand of power throughout the year from 2000 to 2010 in Malaysia. We can see that the demand for electricity doubled in 2010 as compared to in 2000. Malaysia needs extra effort and considers every opportunity to keep her carbon emission low to achieve the 2020 goal. By going green, Malaysia is expected to gain some benefits in term of economy, reserving non-renewable resources, and preserving the environment [12].

Computers have been used throughout numerous organizations ever since their prices become affordable. Moreover, computers help to accelerate the work in many organizations. Unfortunately, the downsides of computer are that they consume power and generate heat. The Gartner group reported that computers and their monitors being left powering on without actual usage consumes more electricity than servers in the organization [13]. Likewise, apart from lighting, computers and its monitor are considered the top power consumer in an organization [14]. Thus, power will be continuously wasted if it is not well managed and controlled.

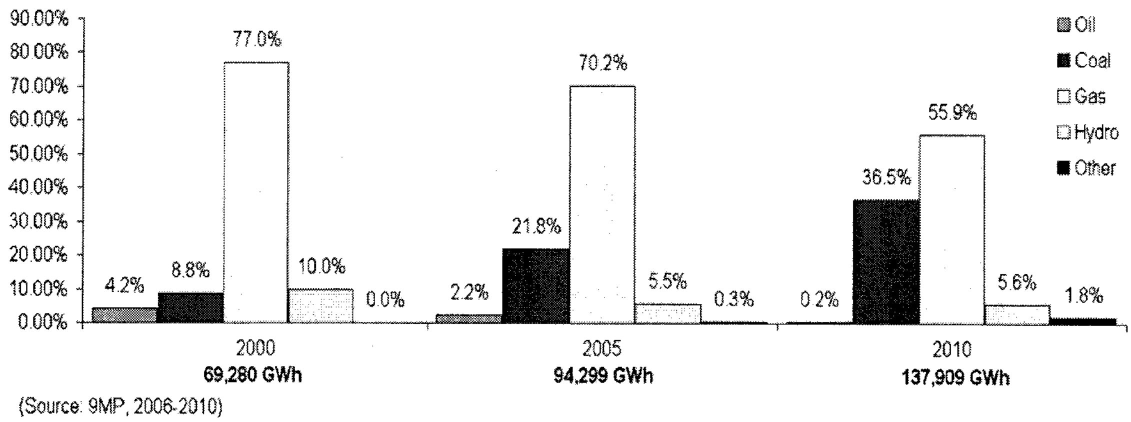


Figure 1.2. Fuel mix in total electricity generation in Malaysia, 2000–2010. [12]

Consequently these reasons have led to the idea of power management to manage and control computer's power state. Power management can be done at several levels of the computer system: the circuit level, the component level, and the operating system level.

Operating systems (OS) nowadays are developed with built-in power management features and tools allowing users to choose or customize power schemes for their computers. The power scheme controls and schedules the computer's power state. For instance, a user could configure a computer to turn off the monitor screen, turn off the hard disks, and switch to standby/hibernation state after it is left idle for a specific amount of time.

A default power scheme was introduced in Windows based operating system as highlighted in Table 1.1. On battery mode, the computer's monitor will be turned off after the computer has been idle for 5 minutes. Furthermore, if it still does not receive any interaction from the user for another 10 minutes, it will switch to a lower power state which is usually the standby state, but on rare occasion is the hibernation state. On the other hand, on plugging-in mode, the timeout for each setting is increased from 5 minutes to 10 minutes and from 15 minutes to 1 hour for "Turn off the monitor" and "System Standby" (or hibernate) setting, respectively. There is one other rarely used option: "Turn off hard disks".

Table 1.1. Window-based OS default power scheme sets for a laptop

	Idle Time	
	Battery mode	Plugged in mode
Turn off the monitor:	5 min	10 min
System Standby/ System Hibernates	15 min	1 hour

Large amount of funds has been allocated for energy usage in a campus. For instance, in the U.S, $100,000 worth of energy is consumed by a 50,000 square foot higher-education building each year. On top of that, the computer is the top four energy consumer in the campus consuming approximately 8% of the total power consumption as shown in Figure 1.3 [15].

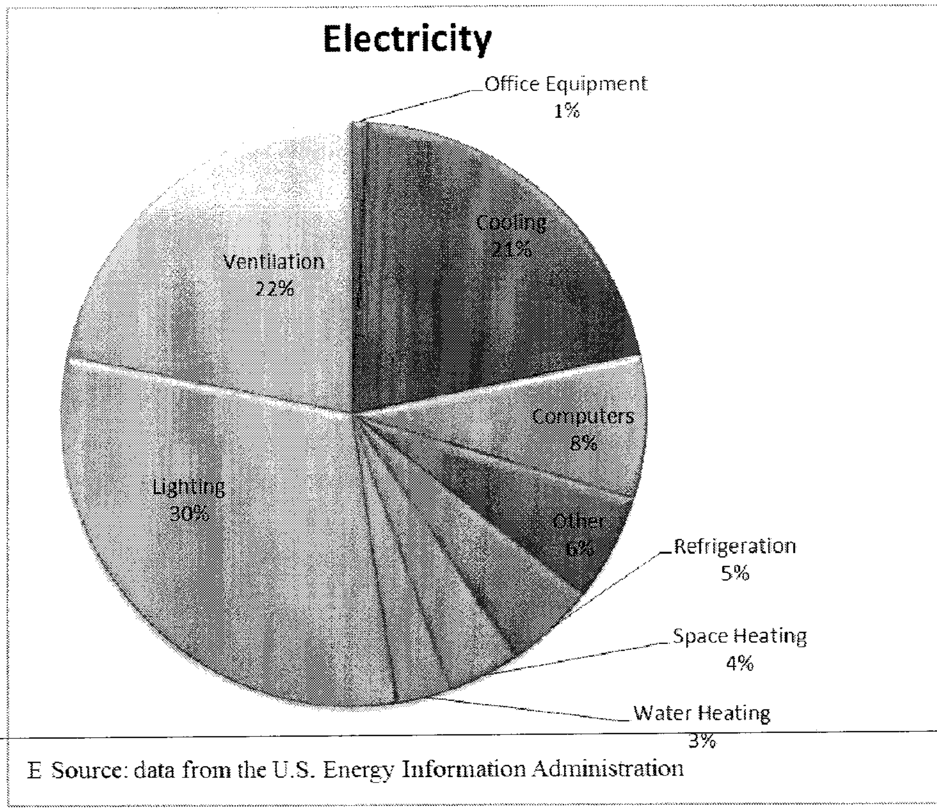


Figure 1.3. Energy consumption in U.S educational facilities by end users [15].

Unlike in the computer in the working place, in campus or school the usage of computers is scheduled according to a timetable which is normally less than 8 hours a day. Without a proper control and management, large amount of energy will be wasted because users often forget or ignore to power their machines off after using.

Some organizations have come up with policies and regulations to control computer power consumption by turning off computers after working hours. For UTP's IT computer labs, a remote shutdown batch file is executed manually every weekday after 5pm to turn off computers on the network. This operation has some drawbacks: it does not know whether the client computers are active; and it may disturb users who are still working with their computers.

Products which were introduced by [13], [16], [17] allow users to control and configure their computer power schemes remotely. However, for computers disconnected from the network, remote control is not possible.

1.2. Problem Statement

The usage of existing power management schemes is still limited. For instance, according to [18], [19], which looked at the usage of power management and the monitoring of office equipment turn-off rates, the results of the study showed that more than half of the computers (54%) were always left on at night, wasting energy. The main reason for this is the lack of user's awareness of the existing power management tool, with other reasons such as limited technical skill and the ignorance of the users.

Just by looking at a couple of computers, there is hardly any difference in the amount of power consumed by computers with the computer power management feature enabled and those with it disabled. However, consider that each computer can save one kWh per year by using a power management feature. With the organization equipped with thousands of computers, there will be thousands of kWh of power can be saved. Moreover, the amount of CO_2 emissions to the atmosphere will be reduced accordingly. According to Energy Star, a program that provides labeling for energy-efficient equipment, the amount of energy estimated to have been saved in America alone by using the Energy Star function has increased almost triple from 59 billion kWh in 2000 to 213 billion kWh in 2010 [20]. It is believed that the amount of saving will continue to increase if more and more people start to realize the benefit of the program.

People's knowledge of the benefit of saving energy is still affected by their false assumptions. They take for granted that the resources generating power will never be used up and so unintentionally wasting power. For instance, lights, air conditioners, computers, and others are always left on even though there is no one using them for work.

With a preliminary result from an observation in an early stage of this research in a few computer labs in the practical computer labs shows that there are less than 25 percent of the computers utilizing the power management setting. Among the power managed computers, we notice that the most utilized features was "turn off display" associated with the timeout range from 10 minutes to 1 hour. Moreover, we observe a computer lab after the practical session ended; we notice that 25 out of 35 computers (desktop and its display screen) were still on presumably there was nobody using them and it was inactive more than 10 minutes or 1 hour idle since last used. This promotes a problem of wasteful power consumption, even though a remote shutdown batch file are executed after the office hour to shut down all computers because during the time computer sit idle, they also consume power.

Power management mechanism, which was used by built-in power management tool, still solely depends on static idle timeout which must be enabled and preset by users. For instance, the PCs' power transitions from an active state to "turn off display" state after 10 minutes of inactivity. Although it helps to reduce PC power consumption, the amount of wasted power while waiting for idle time to exceed is still considered much.

1.3. Research Question

The research questions for this thesis are:

- "What are the factors contributing to excessive computer power consumption?
- Can computer power event and application idle time be used as a parameter to help reducing computer power consumption?
- How to exploit the power event and idle time of the computer, by keeping its complexity low, for the benefit of reducing computer power wastage?

1.4. Objectives

This study would be mainly on the factors and issues surrounding the computer power management, e.g. what are the factors that lead to people ignoring the possibilities of reducing the computer power consumption, and what are the motivating factors that make people willing to conserve the energy through computer power management tools or utilities.

Based on these studies, a framework for looking into two novel computer power management components which incorporate the power transitioning event and application idle time that exploit the user's activity history to dynamically choose and generate a suitable timeout schedule to switch the computer to a low power state and a prototype application for evaluating the proposed components are proposed. This is expected to help reduce idle time or waiting time for the computer which depends on a static timeout to switch to a low power state.

The main objectives of this research project are:

a) To explore factors and issues that influence computer power consumption and monitor user activities and computer usage patterns in the practical computer lab and working environment.

b) To improve effectiveness of the built-in power management tool in OS by proposing a power event and an idle time profiling components to extend the built-in power management functionality and to reduce the waiting time to switch the computers to a lower power state.

1.5. Scope and Constraints

The power management mechanism proposed in this research includes the combination of a monitoring tool and a dynamic timeout-based power management scheduler. The tool implemented based on a proposed model or framework that utilizes the user's activities captured by the proposed monitoring tool. Two environments are selected for implementing this idea, which are a practical computer lab and an office located in Universiti Teknologi PETRONAS (UTP).

The constraints of this research are:

a) The proposed power management tool is written in C# and running on Windows based operating system. It has been tested on Windows XP, Windows Vista, and Windows 7. Some functions, which are available for laptop, are not supported by the desktop because of hardware compatibility.

b) The proposed application depends on history data stored in the log file to generate the timeout. Thus, we assume that the application has the permission to write and override the log files in a particular folder in the computer.

1.6. Contribution of the Research

Upon completion, this research will contribute to Green Information Technology specifically in the domain of Energy Conservation and Energy Efficiency. The potential contributions are listed as follows:

a) Filling a knowledge gap in the area of computer power management by looking at its characteristics and possible solutions (using power transition and application idle time profiling) to minimize power wastage.

b) Identifying the new parameters (power transition event and application idle time) that can be used to reduce the waiting time of the computer being idle without any usage.

c) Development of a new method which utilizes the dynamic/adaptive time-out based computer power management which utilizes the proposed components.

The work in this thesis has resulted in the following publication and exhibitions:

a) Chan Piseth, Alan Oxley, and Low Tan Jung, "Dynamic PC Power State Transitioning based on a Monitoring Service," in International Conference on Computer & Information Science, 2012, pp. 237–241.

b) Display Power Management based on Application Idle Time Profiling (DPMaitp), exhibited at the Science and Engineering Design Exhibition 30 (SEDX30) in August 2012, Universiti Teknologi PETRONAS. (Silver Medal)

c) Display Power Management based on Application Idle Time Profiling (DPMaitp), exhibited at the 24thInternational Invention, Innovation & Technology Exhibition from 9 to 11 May 2013, Kuala Lumpur Convention Center (Silver Medal)

1.7. Organization of the Thesis

This thesis is organized into six chapters with a brief description of each chapter as follows:

Chapter 1 presents a brief overview on green information technology from the world and Malaysia perspective, computers power consumption in an organization, and an introduction to computer power management, and how green information technology relates to computer power management.

Chapter 2 highlights some motivations to reduce power consumption from various methods, and the factors contributing to computer power management issue. Moreover, power management solutions, algorithms, tools and systems, developed in the past, are also discussed as part of related work.

Chapter 3 discusses the research methodology for this research work. We will have a closer look at all activities performed in each phase of this research work.

Chapter 4 highlights the development methodology for two proposed power management components, a data extraction tool, and a proposed power management tool. On top of that, their architecture, system flow, and functionalities will be discussed in detail.

Chapter 5 presents the experimental result in each phase of the research work, and the outcomes of the prototype tool implemented in a practical computer lab and in volunteer users' computer.

Chapter 6 concludes the thesis and some suggestions for future works are presented.

CHAPTER 2

LITERATURE REVIEW

2.1. Motivations and Best Practices

To successfully accomplish energy saving, users and organizations often require guidance, motivations, and best practices from both internal and external sources. The energy pyramid depicted in Figure 2.1 has been applied throughout the energy conservation community with the goal to provide simple ways or practices for saving the energy and using a new form of energy which has a lower environmental effect. It is also an excellent tool to visualize and to define the priority of action to save the energy both at home and in an organization.

The popularity of the action is assigned from the bottom to the top. The reason behind this assignment is that the actions in the bottom level are easier to achieve as compared to others above it because of the nature of the implementation complexity. On top of that, the higher the pyramid level, the higher the cost will be to implement the action [21].

The renewable energy or energy resource sits on the top of the pyramid, in which it has the lowest popularity. This action will be necessary for a long run in the future as it produces clean energy with less environmental effect. Most of the renewable resources such as solar power, wind power, tidal power, wave power, and hydroelectricity power involved a complex set up and require a big investment to implement the project. Therefore, energy efficiency and energy conservation, sitting on the lower level of the pyramid, are giving most promising opportunities and results with low implementing cost and easier implementation.

The next level of the pyramid is energy efficiency. It can be seen in the form of purchasing and using high energy efficient products and equipments. For example, electronic products like air-conditioner, computers, monitors, freezer, refrigerators, washers, and dryers which are certified by energy efficient program or institution such as Energy Star are considered as a high energy-efficient product which consumes less power while providing the same service with the same performance.

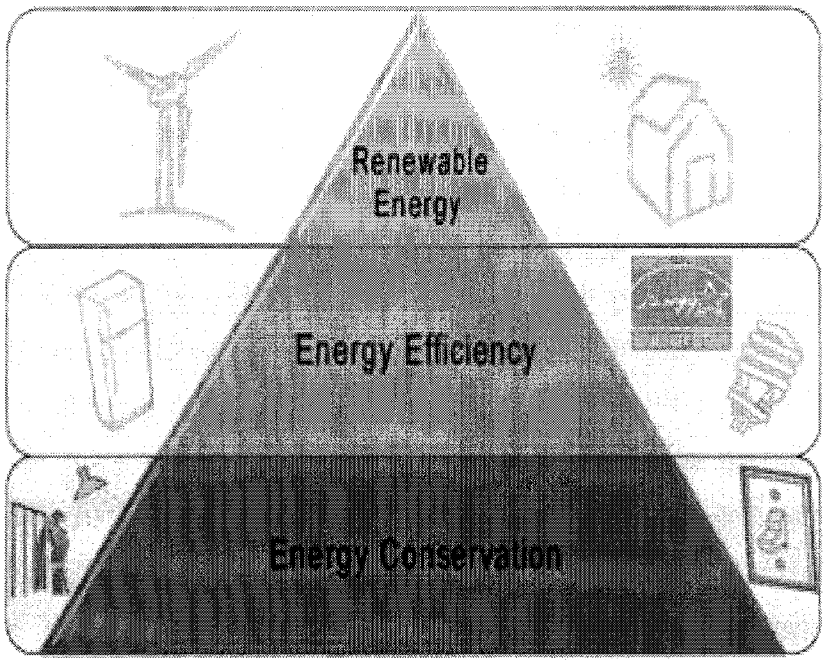


Figure 2.1. The energy pyramid graphic [22]

Energy conservation is the lowest level in the energy pyramid. This action depends on people's habit, behavior, and practices to reduce the energy consumption in homes and in organizations. There are good examples of some of the best practices to help reduce the energy consumption. For instance, some basic practices such as turning off the electronic devices when not in use, and configuring computer power management to go to power saving mode after some period of inactivity. For electrical equipment such as laptops and mobile devices, they consumed energy if they are left in active mode when not in use. The energy is consumed wastefully and this will also shorten their battery life. Thus, it will be wise if all equipment is configured to go to standby or sleep mode as soon as possible. In general, the electrical equipment still consumes a small amount of energy when it is completely off if it is still plugged into the power socket outlet. Therefore, a more effective way to save energy is to unplug the electrical equipment after using or to use power strips to cut off the power source.

2.1.1. Going Green with IT

Information technology plays two roles in the energy pyramid. One of the roles, generally, can be seen as the main power consumers. However, it also acts as a catalyst for saving power consumption which is known as green IT.

Two important concepts have emerged in the field of green IT: using IT to green the organization and the greening of IT facilities in the organization. In general, for using IT to green the organization, it can appear in many forms: visualizing power consumption and carbon footprint so that people can see the performance of their organization which lead to better management of the power consumption in the organization; and facilitating greener work practices such as teleconferencing, online learning, and remote working in the organization.

Alternatively, greening of IT facilities in the organization tends to reduce the carbon footprint by optimizing power usage of the IT facilities. These have been applied in many areas such as optimizing the usage of server through server consolidation and Virtualization which helps reduce the need of always-on machines, providing cloud service such as cloud storage which allows the users to access their files remotely on the cloud, and reducing the power consumption of computers in the workstation by using computer power management application or software.

One of the best examples of using IT to green the organization is the use of social networking websites or Internet portal site in supporting reduction of individual energy consumption. With the increasing usage of the social network sites with billions of users, it has become one of the main targets and opportunities for ecological and environmental information sharing, leveraging, and disseminating. Even though social networks are considered a double-edged sword, it is worth to give it a try. For instance, Jennifer et al. (2007, 2010) proposed an approach that integrates feedback about ecological footprint data into existing social networking sites and Internet portal sites to improve people's awareness on their ecological footprint for motivating people to be more energy conservative [23], [24]. Besides allowing sharing the information on a personal wall, the social network site — Facebook, in particular, allows users to create advertisements, groups and pages that allow other users to join and access in public. Hence, information could be spread to a wider range of audience who is a potential energy consumer. Information such as personal environmental footprint saving, goals, and success story of saving power should be depicted on the individual profile and page to capture people's attention and to enhance their knowledge and awareness on how their involvement and IT can cooperate with each other to green the organization.

Similarly, Tiffany Holmes (2007) proposed a concept called Eco-visualization which combines art work and technology to help reduce energy consumption [3]. An animation of dynamic energy loads at the National Center for Supercomputing Application (NCSA) has been shown as part of the artwork called "7000 oaks and counting" which is publicly accessible through a website. The goal of the artwork is to bring the hidden information within the building infrastructure available and visible to public with the belief that it could change people's behavior towards energy conservation. Therefore, the concept of social network and eco-visualization could be combined as a platform for social data exchange to promote the awareness of the energy conservation by making it publicly accessible anytime and anywhere.

Recently, the booming of the technologies has brought along new opportunities for people to improve their living lifestyle as well as reducing the carbon footprint of their organization. As depicted in Figure 2.2, people have been given access to a wide range of communication through interconnected network which is known as the Internet. As a result, some tasks such as working, interviewing, meeting, and learning that required people to travel around using various means of transportation such as car, train, plane, and ferry have been replaced with IT technologies. For instance, teleconferencing can be set up for the meeting and interview instead of travelling a long distance to the actual destination. Students are opened to a new gateway which allows them to study online through an online portal and streaming media.

Furthermore, people begin to be familiar with a new culture of working from home or nearby café instead of travelling a long distance to the office. Therefore, the usage of transportation can be greatly reduced consequently reducing power consumption and the environmental pollutants. Even though, IT has bypassed the transportation layer, we know that computer in particular still consume power.

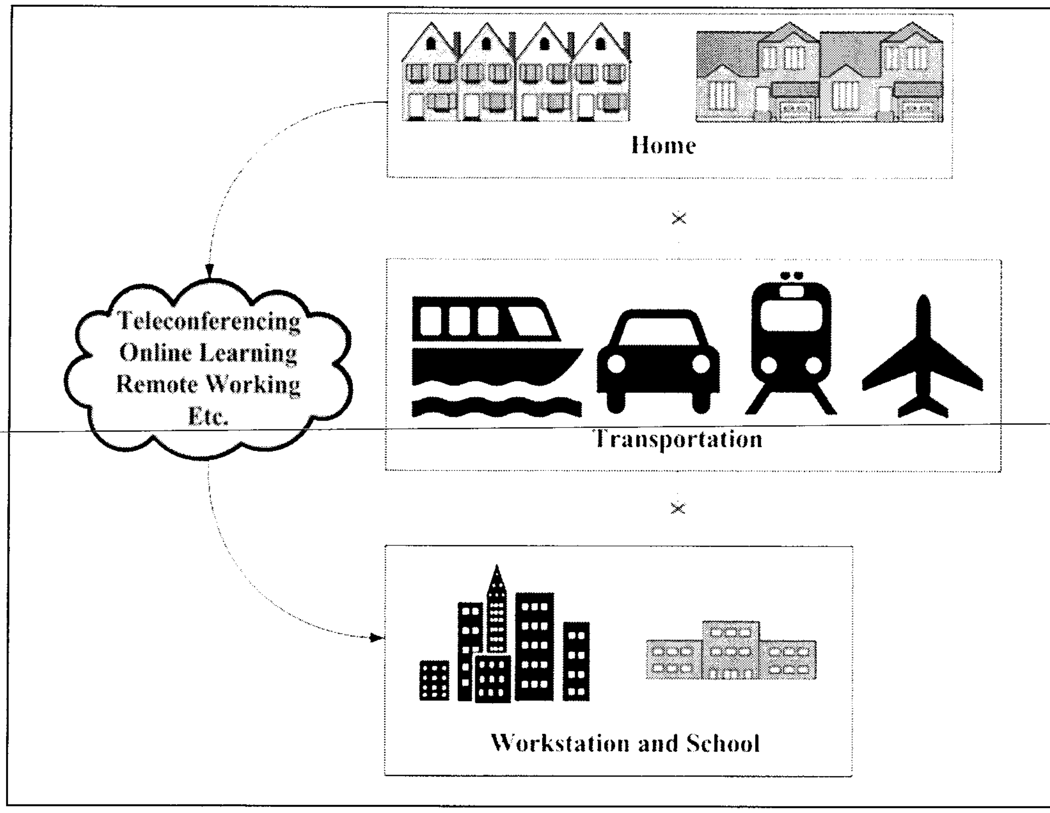
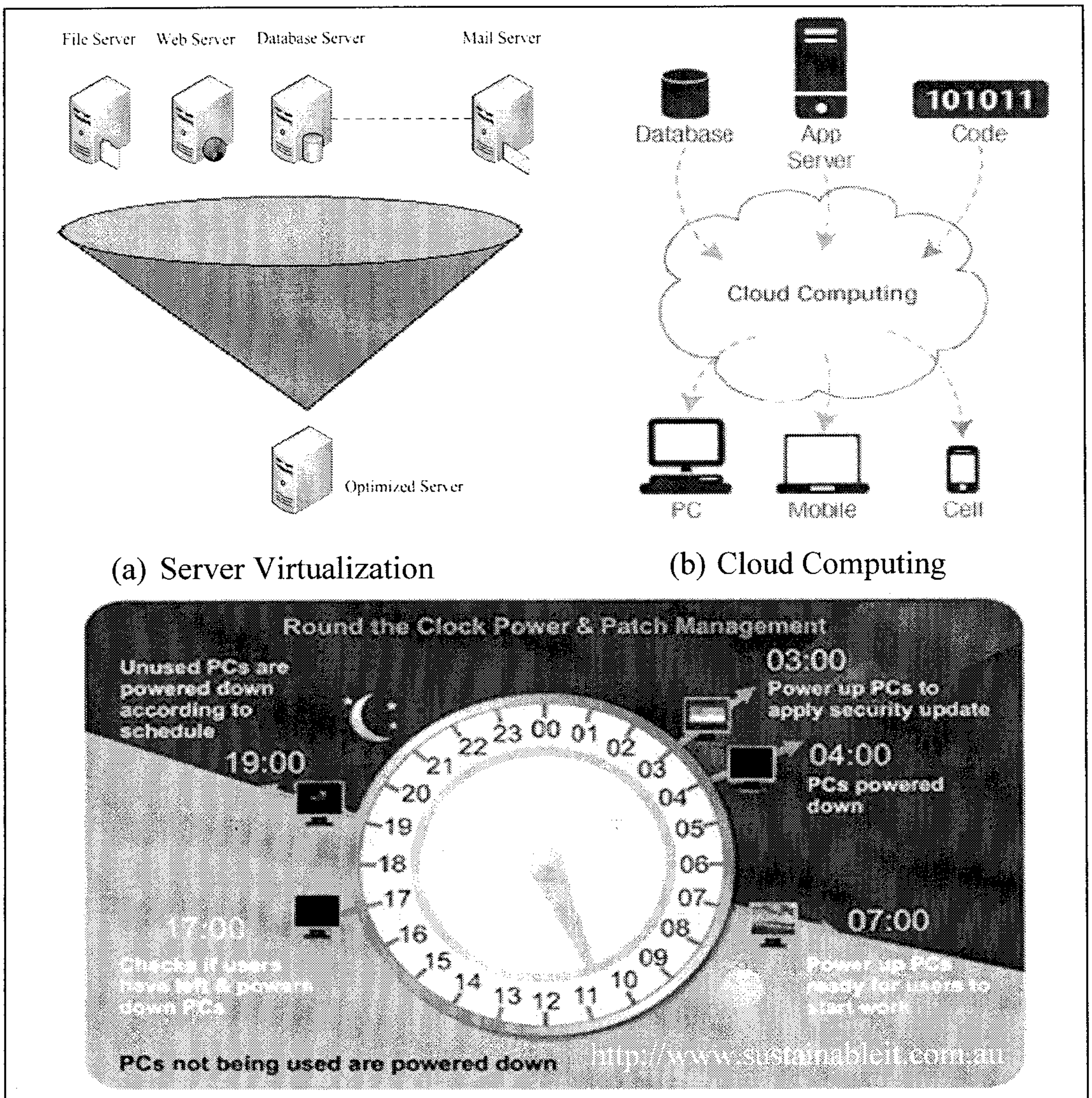


Figure 2.2. How IT connects to the workplace and school by passing transportation layer.

The issue on how we can use and manage those technologies to be more energy efficient is of great interest. Examples which use the techniques such as Virtualization, cloud computing and computer power management are illustrated in Figure 2.3 for greening the IT facilities in an organization. By optimizing the usage of computers and servers through virtualization, the power wastage can be decreased because the underutilized computers and servers will be powered off and they will be consolidated into a particular machine.

The contemporary computers provide a high end performance with high capacity hardware resources. Some personal computers' performance and hardware resources are even better than some of the server computers available. Unfortunately, those resources are not used to their fullest potential. Apart from performance, computers are designed with different architecture and run on different operating systems. Programmers and developers should find a way to implement their software and applications to be portable and supporting all architectures, platforms, and environment. Thus, virtualization is known to be a good solution.



(a) Server Virtualization

(b) Cloud Computing

(c) Power management

Figure 2.3. Examples of greening of IT facilities in an organization

Virtualization is the concept of sharing resources such as CPU, RAM, GPU, storage, and network among applications and services. The term "virtualization" always leads people to assume that it is either desktop or application virtualization because these are the most commonly used applications. The desktop Virtualization is defined as a concept or software technology which isolates a PC desktop environment from users who are used to access it and allows to access a new environmental [25]. For application Virtualization, it is defined as techniques to encapsulate the running applications from the underlying operating system on which they are executed making them to be more protected, more flexible, and easier to manage [26].

Apart from desktop and application Virtualization, there are other types of Virtualizations such as network, storage, and server Virtualization [27]. For server virtualization, a single computer or server takes advantage of its under-utilized resources to share among virtual servers running on different operating systems (Figure 2.3 (a)). It converts and allocates its available hardware resources into multiple resources dynamically for each virtual server. Virtual servers are capable of running without interfering with each other even though they are running on a single computer or server.

Cloud computing and services illustrated in Figure 2.3 (b) provide users with a flexible way to work anytime and anyplace. It is an alternative to having in-house servers. In addition, users can scale their work anytime without having to think about upgrading their computers or server hardware and software because the service provider will handle it transparently. Thus, it is no doubt that the power consumption of the organization will be reduced. It does not mean that by having the services hosted on the cloud it does not consume any power. Instead, the power usage will be optimized.

In general, computers need electricity to power on, and they will keep consuming power even though there is no one using them. That is the main reason why computer power management was introduced. The power management handles the power consumption of all components in the computer system. It switches the system to low power mode when the system is idle as shown in Figure 2.3 (c). Computer power management will be further discussed in the next session.

2.2. Computer Power Management Overview

The power consumption bill has become a main issue in many organizations because of the increasing usage of computers. It involved both money and environmental issue. Most of the sources used to generate power are unsustainable and cause negative effects to the environment. For example, petroleum, coal, and natural gas used to generate electricity are emitting carbon dioxide (CO_2) causing greenhouse effect which contributes to the increasing of global temperature. In some countries, electricity usage is limited by quota in the organizations, thus it must be used effectively and efficiently.

Power consumption varies from computer to computer. It depends on the manufacturers' specification because hardware and software are chosen and installed by them. However, time can be used as one of the main parameters to determine how much power ($Power_{total}$) is consumed as given by Equation (2.1). The more time a computer is used the more power being consumed. Thus, we can save the amount of power wastage if we can reduce the system waiting time by switching it to a low power state sooner as shown in Figure 2.4.

$$Power_{total} = \sum_{i}^{n} Power_{state(i)} * Time \qquad (2.1)$$

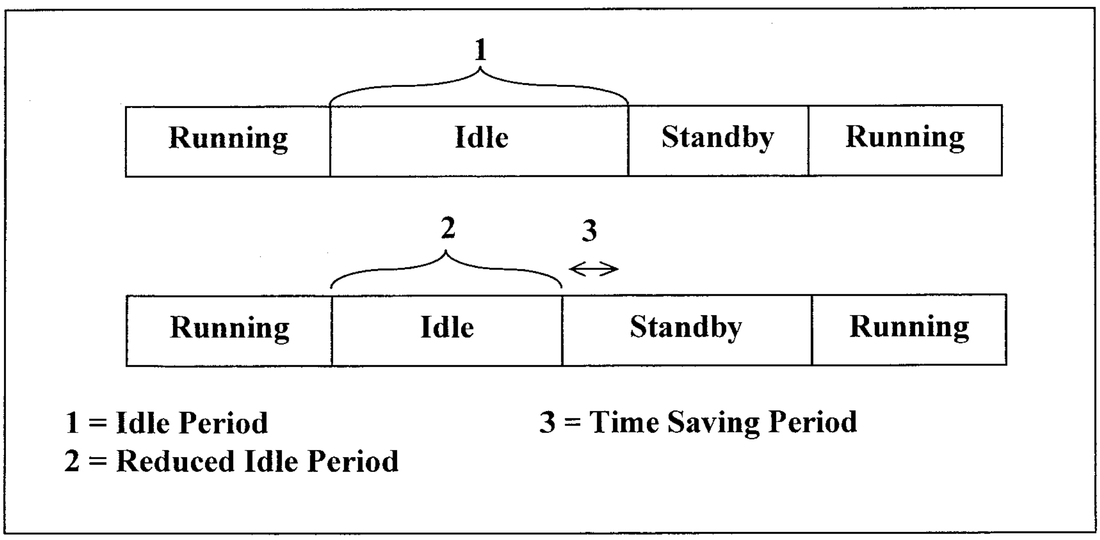


Figure 2.4. Comparison between two idle periods

A computer is an assemblage of many electronic components: power supply unit, motherboard, CPU, RAM, hard disk, graphic card, etc. Each of these requires electricity to power up. On power up, they consume energy and generate heat; consequently, they require cooling to prevent overheating. Thus, how can this be monitored and managed? Power management has been introduced to deal with this problem.

There are two main power management components for the computer: hardware components and software components. For the hardware component, its specifications and interfaces are provided by the manufacturer to allow the operating system to control over each device's power state through the Basic Input Output System (BIOS). In some cases, the hardware is able to manage its power state. For the software component, it is a set of utilities normally bundled with the operating system to control power event transitions based on prescribed timeout [27]. Two important terms closely related to the power management are Advanced Power Management (APM), and Advanced Configuration and Power Interface (ACPI) [28].

Table 2.1. APM power states and description [27], [29], [30]

State	Description
Full on	The system is working normally and all devices are on as there is no power managed.
Enable or Ready	The computer is fully powered up and ready for use. However, some of the hardware might be under a low power state or off state.
Standby	The computer is not working until receiving hardware interrupt from either user activities or operating system command to resume the operation. It is quickly resumed. The CPU clock is slowed or stopped.
Suspended/ Hibernation	The computer appears to be off. The computer's CPU clock is stopped and the computer is running in the lowest power. Computer resumes its state only if is receiving signals by power button, timer alarm, or etc. The system context is saved in RAM or hard drive (hibernation). For storing in hard drive, The previous state or system context is preserved in hiberfile in the hard drive to prevent from unforeseen power shortage.
Off	The computer is completely powered off.

The APM standard was developed by Intel, IBM, and Microsoft in 1992. Windows 3.11 was the first operating system to take advantage of it. Later, Windows 95 was designed similarly [3]. According to [29], APM supports five power states as shown in Table 2.1.

Working with power management using APM, an APM-aware driver must be installed for each device to enable communication with BIOS. Devices will be put into, that is, switched to, a low power mode when BIOS receives an interrupt from the driver after an idle period has been exceeded.

The architectural issues associated with APM include:

- Communication issues between APM and operating system—the computer's BIOS might issue the command to switch the system to a low power state without informing the operating system which could cause a loss of data in RAM and system context. In the same way, the operating system might power-managed itself without going through APM.

- Devices detection—APM will only detect the supported APM devices in the motherboard. A new add-on device will not be detected and monitored by APM. This will appear to the system as unsupported APM devices so there is no power managed for that particular device. As APM unable to monitor the activity from the external devices, the system still appears to be idle even though that device is running.

- Network connectivity—APM does not support network connectivity monitoring. This causes the system transitions to a low power state after the system timeout exceeded even though the computer's network traffic is still high because of downloading activity.

- Setting/Configuration—APM setting must be changed through BIOS. There is no direct interface for changing the setting in the operating system. On top of that, computer need to be rebooted each time the setting is altered.

The ACPI standard, which is a standard for operating systems, was designed by Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba in 1996 as a replacement for APM. ACPI provides more abilities that are flexible. One of the main abilities is giving the operating system direct control over the power management configuration of the whole system. It was introduced in Windows 98 by Microsoft [27], [31].

According to [3], for ACPI architecture, all power management and administration tasks are gathered into one place that works as an interface between the operating system and hardware. There are two main components in ACPI architecture: hardware components and software components. Power requirements and status are supplied with hardware devices and at the same time software logic for controlling power is supplied by the operating system.

Table 2.2. ACPI power states and description. [27], [31], [32]

Categories of States	Description
Global	It is the power management's highest level of abstraction. There are four global states G0 (Working state), G1 (Sleeping state), G2 (Soft Off), G3 (Mechanical Off)
Device	It is a power management state of devices. There are four device states D0 to D3.
CPU	It is a power management state of the CPU. The CPU states normally vary from manufacturer from C0 to Cn.
Legacy	It is a power state where the power management decisions are made by the platform hardware/firmware shipped with the system. This state is mainly for the system that only support a legacy OS that does not support the OS-direct power management.

Different states were supported which are mentioned in ACPI documentation shown in Table 2.2. The ACPI global state is divided into four levels G0 to G3. The following is the description of each level [31]:

- G0 is the normal working state. This is a power state where all operations and applications are running normally. It consumes the most power among all global states. However, individual peripheral devices and components are power-managed independently. It is based on the activities on each device and component.

- G1 is sleeping state which normally appears to user as being turned off. This state is invoked when the system is idle for a period of time. It only consumes a few watts of power to keep the RAM or hard disk running so that it is able to store the system context. The system recovery time depends on which sleep state the system enters. There are six sleeping states from S0 to S5 specified in ACPI documentation.

- G2 is soft off state. The whole system is powered off but there is still a small amount of power drawn through the power supply unit to keep some devices alive. For instance, network card maintains some power for Wake-on-LAN operation. There is no application running and the system context is not retained. Reboot is required for the operating system. The system recovery time is long. It is not safe to remove or disassemble any of the components.

- G3 is a mechanical off state. The system is completely powered off as there is no power supplied to the power supply unit. It takes the longest time to boot to G0 state. It is safe to remove or replace any devices and components from and to the computer.

Sleep state is further divided into six sleeping states from S0 to S5. These states describe how deep the system sleeps. The power consumption varies based on the depth of the sleep. The description of each state is as follows [31]:

- S0 is a normal working state. It resides in G0.

- S1 is a sleeping state with low wake latency. The system context is maintained for both hardware and operating system.

- S2 is a sleep state which offers improved power saving mode over S1. The CPU and system cache are not maintained. It takes a bit longer to recover from sleep but it consumes less power.

- S3 is a sleep state where CPU, system cache, chipset, and peripherals are powered down except RAM which goes to a low power state with self-refreshing state.

- S4 is known as hibernation. The platform's setting and system context are maintained in the hard drive. All devices including RAM are turned off.

- S5 is G2 soft off state. It is similar to S4 except that it doesn't maintain any system context. Reboot is required for the computer.

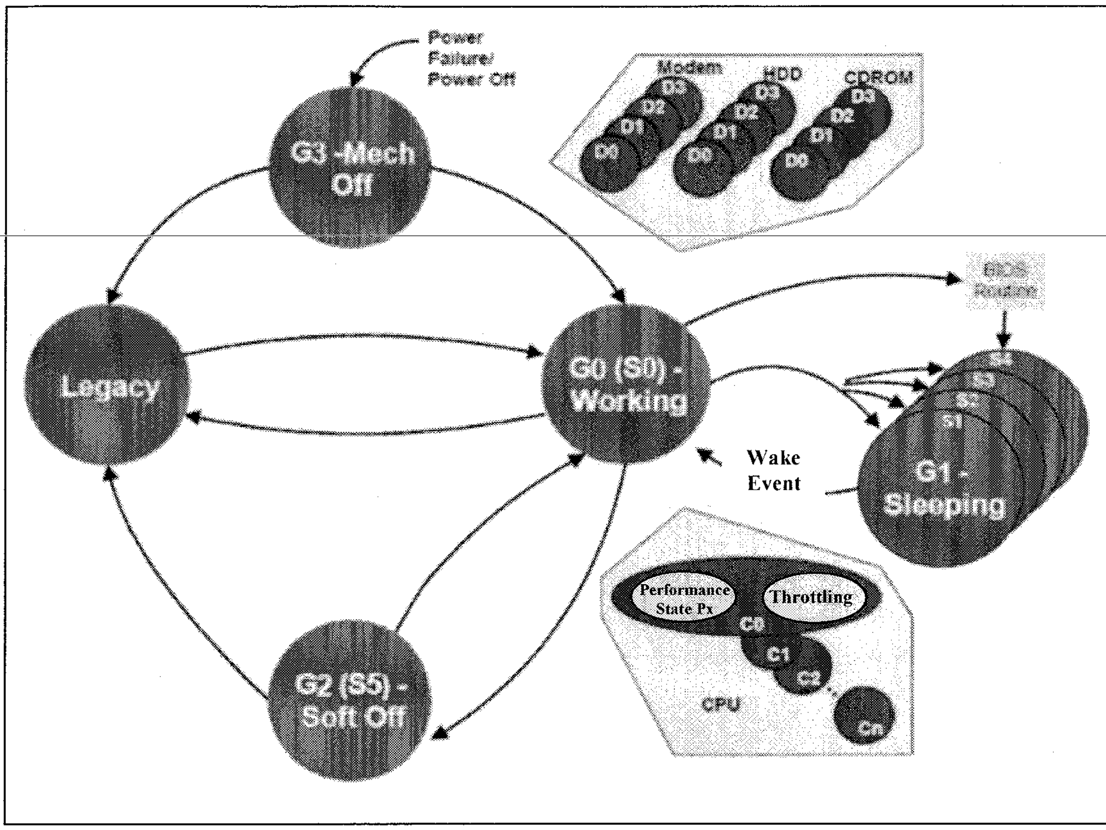


Figure 2.5. Global system power states and transitions [31].

Figure 2.5 shows how the ACPI compliance computers switch between Legacy, Working, Sleep, and MechOff states.

Computers that supported legacy BIOS power management boot from Legacy state and transition to Working state when powered on. Otherwise, they boot directly from Mechanical Off state to working state for unsupported legacy state computers [31].

Working state consumes the highest power but it depends on their resource usage over time. In this state, users are working with applications and services running on the computer. The individual application or service requests different resources causing the computer to wake and associate their devices and hardware to respective requests. An individual device and hardware switched to a low power state whenever there is no task associated with them and return to normal working state with a little bit of latency. The latency depends on the level of sleeping state—the deeper the state the longer is the latency.

The computer will transit from the working state to low power states when the whole system becomes inactive after a preset timeout is exceeded. The sleep state is determined by preset power management setting. In the sleep state, if it receives the signal or interaction from the user, it will return to Working state. The latency depends on the level of the sleep state.

In the case of power failure while it is in a Working state or Sleep state which doesn't save the system context, the computers will automatically boot up from Mechanical Off state in the next power on to the Working state.

Two categories of power management applications well-known to most of the users are standalone and network controlled power management. Standalone power management normally refers to a power management utility which does not communicate over a network. This utility is normally bundled with modern computers. On the other hand, network controlled power management refers to a utility which performs power management over a network. Remote shutdown and Wake-On-LAN utilities are two main technologies that are used alongside with scheduling application to control the power management feature over a network.

Computer power consumption can be cut down by enabling its built-in power management capability. However, researchers and manufacturers introduce other approaches and methods, and products that can help to save more power consumption in the organization. For instance, social networks have been used to improve people's awareness of their ecological footprint at home or organization for motivating people to be more energy conservative [23].

Some commercial products such as Verdiem [17], 1E watchman [16], and Triumfant [13] (all enterprise class power management tools) have been introduced and they work as centralized power management tools to remotely monitor, shutdown, schedule, or impose a power scheme, on the client computers in an organization's network. They also help users to visualize the estimated amount of energy saved by utilizing the provided functions.

In [7] instead of using the commercial products, the author presented "Polisave" which utilizes existing technologies such as Wake-on-LAN, Hibernation, and Web services to deliver a web-based architecture which allows users to easily turn on/off the PC remotely. However, the proposed solutions depend on user control and interaction. In other words, users still can ignore to turn off their PC or ignore to schedule and set their computer's power management settings.

2.3. Display Power Management Overview

The computer display (the display screen unit) power management has a more successful story than computer power management as it is more reliable and gives a much promising result [33]. Monitor screen depends on the computer, the initiator, to issue the power managing command such as dimming the screen or switching to sleep state. This is because the monitor screen does not know when to issue the power managing action, as it does not receive the information about the computer's activities and states directly.

Display power management signaling (DPMS) is used for controlling most of the computer monitors. To work properly, both the monitor and computer must support DPMS. Since the monitor screen is directly connected to the video card, DPMS signal must pass through the video card. Some video cards have the ability to issue the DPMS signal directly to the monitor through some software. After entering the first low power state, the screen will be able to handle the succeeding low power state through its internal timer. Some monitors supported the dimming state allowing them to lower their brightness for some energy saving, but the saving is small as compared to sleep state and standby state. The monitor display leaves the low power state after receiving a signal from the computer when it becomes active.

In most cases, DPMS normally support four power states which are active, standby, sleep, and off state. Besides these four states, there are other two alternative power levels. The two alternative power levels are normally controlled by the screen saver setting. They appear the same as a blank screen to the user. However, they have different recovery time.

The BIOS timer or the video card special software allows the computers to initiate the power management signaling and command to the monitor screen. The timeout for the timer is preset by the users either through a BIOS setting panel or power management software. When the idle time of each computer system exceeds the timeout, the DPMS signal will be sent to the monitor screen to indicate that it should go to the low power mode. The current image on the screen will be redisplayed when the monitor screen receives the DPMS activated signal normally caused by keyboard and mouse event.

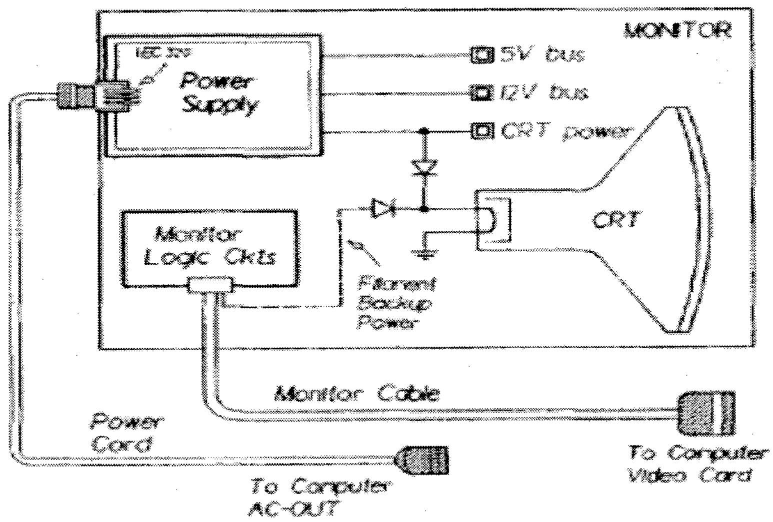


Figure 2.6. A monitor power management system based on power-source switching scheme [34].

Apart from being power-managed with DPMS, monitor has two alternative methods: blanked screens and computer power-source switching. However, the usage of these methods has faded over time even though they show a promising saving in an early implementation stage. For blanked screen method, the monitors are handled by the screen saver or specific software that comes along with the video card to make the screen blank when the computers become inactive for a preset time duration.

For computer power-source switching method, a solid state relay located within the computer's power source controls screen's power source. Specifically, the monitor screen depends on the computer's power source. In this case, if the computer enters a low power state which involves shutting down the computer's power source, the power supplied to the monitor is cut off. Therefore, the monitor screen only supports power-on and power-off mode in this method.

2.4. Barrier for Power Management

Power management utilities were developed with the aim to reduce computers' and their monitor's power consumption when they are not actively in use. The operating system, software, CPU, monitor, network cards, video cards, and peripherals are directly managed by power management system [35]. Power management feature is enabled by default in up-to-date operating systems, but users tend to ignore its importance. What are the barriers and ignorance to the high usage of the computer power management?

2.4.1. PC's Response Time

One of the most important aspects influencing the ignorance of users towards using power management is the response time. PCs need some time to boot and resume from shutdown, and standby or hibernation state. Typically, it takes 20 to 40 seconds, but sometimes longer, for a PC to resume operation. It is because some components such as hard disks and RAM need to reload their system context. Standby and hibernation states are introduced to address the problem of long booting time. However, these new states still need quite an amount of time for the system to fully resume to the working state.

The users occasionally keep their computer power on during break time rather than switching to a low power state or turning it off because they want the convenience of computers to be available at any time. They do not like to wait for the computer to load or boot each time they need to use it. It is because these users do not really know how much power was wasted by leaving their computers on. According to [19], 60% of desktop computers were left running after office hours, only 4% of which were in a low power state. This shows the end user ignorance level is high and it should not be overlooked.

The long response time always causes irritation to users. The longer time to wait for computer to resume to working state the higher the irritating level will be. To reduce this irritating level, the time taken for power state switching should be as low as possible.

2.4.2. Computer Worn Out

The users believe that a frequent power state transitioning could lead to faulty components such as hard disk, RAM, CPU, and screen. It would be true if the computers were old-fashioned or out-of-date [36]. For instance, the old generation hard disks did not automatically park their heads when computers are shut down so the hard disks could be harmed if computers are frequently on and off.

There are some research identified that it would require on-off cycling of every five minutes for computer to reduce its lifespan [35]. On top of that, through better design, a modern hard disk has largely overcome this problem. Based on the Rocky Mountain Institute's report, today's new and improved computers are designed to withstand 40,000 on-off cycles before failure. This number is high which is unlikely to exceed, even if the computers are used from 5 to 7 years [37–41]. Therefore, turning it off will reduce unnecessary heat from the CPU and other components resulting in saving power and preventing components from being faulty.

2.4.3. Network Connectivity

Computers in the organization need to be maintained and updated that require a persistent connection to make sure they stay healthy and up-to-date. The network connectivity and internet are the source for daily data backup, software updates, and patches; the connection cannot be maintained in a low power state such as standby and hibernation mode. This leads users to ignore the usage of power management features.

Power management has much more functionality to offer for users to solve such problem. With a correct setting, computers do not always go to a low power state such as standby and hibernation state immediately. The power management allows users to impose an alternative power scheme so that computer's network connectivity is maintained. Users could just configure a power scheme which only controls the computer display. In this case, the computer display will be turned off after some period of inactivity instead of being set to go to standby or hibernation state. With this replacement, some amount of power can be saved because one of the power-hungry components is power-managed.

As more and more computers are connected in networks, working remotely from anywhere to access data from individual machine is getting more popular. As most of the users think that by enabling power management, it will interrupt the connection to their computer. Thus, they keep their computers on most of the times.

In fact, most of the users do not know that to remotely access computers when they are in a low power state is not a problem anymore with the advancement of software and hardware technologies. It requires an extra effort to wake the individual machine up first before it can be accessed remotely. For instance, one of the popular technologies is Wake-On-LAN (WOL) which is supported by most modern network cards. Those network cards have the ability to handle requests from other computers to wake up the individual computer within the network. The request is sent through a magic package. Alternatively, some recent hardware technologies such as Intel vPro also offer similar functionality which doesn't rely on the WOL standard [41].

2.4.4. Higher Power Consumption

Computers on/off transition consumes more power than computers in idle state due to power surges during cycling on and off [41]. It really consumes more power during the transitioning period. However, if we compare between the power consumed during the computer's inactive period and the power consumed during the computer on/off period combined with the power in low power state, inactive computers period consumes power far greater than the power consumed during the on/off period combined with the power consumed in the low power state. It is because of the power surges period is short and the power consumed in the low power state is far less than the power consumed in inactive period. As referred to Table 2.3 for two minutes of inactivity, the computer would consume (2.17 – 0.36 + 0.09) = 1.72Wh higher as compared to switching the computer to the low power state.

Table 2.3. An example of the power consumption for a computer.

State	Power (watts)	Duration (seconds)	Power Consumption (watt hours)
Inactive	65	120	2.17
Transition	85	15	0.36
Low Power	3	105	0.09

2.4.5. Software and Hardware Upgrades

Computer hardware upgraded such as CPU, motherboard, and graphic card will affect the power management capabilities. Likewise, operating system software and some application upgrades could alter the characteristic of power management and sometimes it is disabled or becomes unsupported. Thus, a thorough review of the upgrading feature for ungraded software and hardware should be taken to determine if the changes interfere with power management.

2.4.6. Screen Saver

It is always confusion between display power management and screen saver. They are two different things but they have a close relationship. The screen saver is used to consider as part of display power management. The screen saver is first introduced to help minimize the power consumption of cathode ray tube (CRT) computer display by preventing the permanent etching of a pattern to be displayed on the monitor. A blank screen or a constantly moving image is also used in the screen saver [42].

In fact, the screen saver could sometimes increase the power consumption of the computer system as much as twice the power consumes in the "working" state because displaying complex graphic requires more computation which consumes more power. For instance, graphic-intensive screen saver such as 3D graphic with animation requires additional computational power from the CPU and GPU. Furthermore, it occasionally prevents the computer and its display from going to a low power state [43]. In this case, when a display timeout is set to be less than a screen saver timeout, the display will resume from a low power state to an active state displaying the screen saver after system idle time exceeding the screen saver timeout.

2.5. Existing Techniques for Display Power Management

The concept of power management has been introduced for quite some time already. It is by default pre-installed in the OS to look at user idleness and to determine whether to switch the computer and its display to a low power state. Some third party power management tools and applications can be installed to extend the traditional function of the built-in power management. Some applications are standalone applications handling individual computer power management. Some applications are network controlled handling a group of computer's power management of the whole organization's network-wide.

New methods, mechanism, and algorithms have been proposed to address the problem of the power management in various computer system levels such as in circuitry level, component level, device level, and operating system level (application and process).

2.5.1. Human Interface Device Based Power Management

Human interface device (HID) based power management is the most widely used power management policies. Most of its applications are developed on the application level of the operating system. It has been bundled with almost all of the popular OS such as Windows, Ubuntu (Linux), Mac OS, etc. On top of that, it has also implemented in the portable and smart devices such as tablet, e-reader, mobile phone and smart phone.

For this type of power management, computers switch to a low power state after a prescribed timeout exceeds the system idle time and there is no user interaction or activity. User interaction or activity is determined by either a keyboard or a mouse event [44]. System idle time will be reset and the system and display are powered up after receiving user interaction or activity.

This policy solely depends on a parameter—timeout which is static and rigid. As shown in Figure 2.4, there is quite an amount of unnecessary duration waiting for the computer's monitor screen to switch to standby state. If this waiting time can be exploited, there is room for saving the power consumption. For instance, if a CRT screen with 5-minute timeout, 75 Watts in active mode, and 3.3 Watts in off mode, can be switched to standby mode a bit earlier than its default static timeout by 3 minutes; (75 Watts x 3 / 60 – 3.3 Watts x 3 / 60) = 3.585 Watts can be saved for that particular occurrence.

In some case, a large amount of power is consumed in the inactive period waiting for command from power management to switch the computers to a low power state. It is because user tends to set a long period timeout and even ignore to set one. Thus, if a dynamic timeout setting could be employed, administrators just need to enable the feature once and let the system do the rest of the work.

In recent years, a research interest has switched from hardware, OS, and network level power management to application and process level power management. There are quite a number of literatures discussing about various methods of power management in process and application level to reduce power consumption by identifying user presence, attentiveness, and behavior. These methods capture sensing data from sonar [45], camera [46–48], and other sensors. With the application of event and log analysis, rules and policies are generated for power management.

2.5.2. Sonar based Power Management

Sound Navigation and Ranging (Sonar) is a sound wave system which is well known because of its wide range of applications to locate and navigate the position of an object under the water. For instance, Sonar has been applied in robotics for obstacle avoidance or detection, and range measurement. It allows the robot to sense the surrounding area and to move accordingly.

In [44], [45], the author describes a technique which utilizes an inaudible recorded echo sound ranging from 15 to 20kHz emitted from a device's built-in speaker to detect the user's presence. For user activity study, the author considers five user attention states: active (interacting with keyboard and mouse), passively engaged (reading material on the screen), disengaged (sitting in front of the computer, but not facing it), distant (staying nearby computer, but not in front of it), and absent (user has left the area). An office environment was set up for the experiment with five activities mapping to each user attention state. Each activity was recorded for four minutes.

By applying this technique, 81% of overall display power consumed when the computer is idle can be saved as compared to 51% power saving using HID timeout policy [44]. Even though this technique gives a promising result, there are a few main weaknesses. The author assumes that the environment is static but in practical the environment can change anytime. Moreover, ping sound generated by the system might be interfered by surrounding sounds such as the music sound. Most importantly, in some cases, the display goes off without any notice while the user still attends to it. For instance, in the situation that the user does a presentation, turning off the display will irritate the user and audiences. Thus, there is room for improvement in term of reducing user irritation level because of false detection and user involvement.

2.5.3. Camera based Power Management

A prototype system called FaceOff was introduced as an alternative solution for display power management. This system utilizes the camera to sense user intention. It solely depends on face detection module which analyzes a large central area of skin color in the image captured every second [47]. The proposed technique saves an average of 29.5% of power higher than the saving which is given by implementing the default power management setting which is the HID-based power management.

A similar system with additional policies and enhanced detection algorithms is implemented in [46]. If the user switches attention away from the display, the display's backlight will dim down to a lower level. If the lowest level reached, the display will enter standby mode. User presence detection and Eye-gaze detection are used to determine the presence of the user in front of the screen and user attention, respectively. For user presence detection, skin-color pixel is the main parameter which is used to determine the user presence similar to what FaceOff system offered. The number of skin-color pixels in the largest foreground region of the input image frame is counted and comparing between the previous frame (f_t) and the current frame (f_{t+1}). The thresholds to determine the skin-color are set with some conditions: $0.55<R<0.85$, $1.15<R/G<1.19$, $1.15<R/B<1.5$, and $0.6<R/G/B<1.8$ (R=Red, B=Blue, and G=Green)[49]. After determining the skin color, the system verifies if the user's face is within minimum selection range with the condition of 25% of skin colored pixels presented. Therefore, if all conditions are qualified, the user is assumed to be present.

For eyes-gazed detection, the author uses the eye-tracking algorithm [50] that applies some optimization introduced in [51], [52] specifically for tracking computer user. It scans through the image captured during user presence detection phase to locate the between-the-eyes region of human face using the six segment rectangles technique that satisfies the bright-dark relations [46]. Comparing to ACPI in active mode, this technology consumes a little bit more of power because extra hardware is used. If looking at total power consumption, this technology saves 36% of the total power of the display screen in the test. However, there are a few key limitations of this technology: the user must be within an arm-long distance (50-70cm) from the computer display screen; if multiple users are presented in front of the screen, the problem will be arisen as the system would not know which user focuses on the screen. When using the camera for detection, lighting always becomes a main topic to be discussed and improved; and the computational overhead is caused by the user presence and eyes-tracking detection.

In [48], a display power management (DPM) system with more precise detection level is presented. The proposed system helps to reduce the user's irritation level by implementing four states for user presence detection: away, interactive, attentive, and inattentive; and a bridge state between attentive state and inattentive state called weakly attentive state. It is introduced to reduce the fault detection thus reducing user irritation. By implementing this power management scheme, it helps to reduce the average system-wide power consumption additionally 5%. Moreover, comparing with the HID timeout based approaches, the proposed scheme saves up to 13% more power.

2.6. Understanding User Behavior and Usage Pattern

Computers and their user have a close relationship. They are interacting with each other like employer and employee. Computers do what the users want and at the same time. Some requests and actions are captured and recorded, and some are just broadcasted in term of messages or events to inform the devices and components in the system. This information can become an important resource for a future reference and to understand the user behavior.

In [53], the author introduces a prototype system which helps to identify suspicious users in a web environment. This system utilized the real data logged when the end user interacts with the web environment. These data are stored in two log files: firewall log and web log in a centralized server. A correlation analysis between the firewall and web log is performed. A decision tree is developed based on the correlated information which helps users in taking a proper decision.

A novel method for analyzing computer usage logs which implemented in Window-based OS is proposed in [54]. The working patterns and behaviors of employees in a company are identified through two levels of concept: task summarization using HMM, and user behavior comparison using kernel principle component analysis based on a graph kernel.

These concepts have directed a way of utilizing the information associated with the computer's power transitioning event and the application idle time event to understand the usage activities and usage pattern of both users and their computer. By understanding the usage pattern and activities of users and their computer, it is hoped that the current power management technology's functionality can be extended. Therefore, more power can be saved with low impact to the user.

2.7. Moving Averages Market Forecast Algorithm

Moving average (MA) is a statistical technique which is used as a tool to analyze time-series data in some data point, and to predict the future data based on a subset of the recent actual data with a specific interval [55], [56]. This is one of the most successful techniques to identify trends of the past data and to gain profit from it.

Moving averages techniques have also been used in many areas for prediction including sensory network, signal processing, and image processing [57]. However, they have been widely applied in financial and market forecasting field.

In terms of finance, this technique allows us to smooth out fluctuated data not only in stock prices but also in commodity prices, and in foreign exchange rates. It gives us a better view of the general direction of the series we are studying [58]. When seeing the movement of the price in a more simplified form, we can determine what action should be taken much faster and easier.

A successive average price is generated from the average of prices in a constant period [59]. For instance, a 25-day moving average of a stock price shows us: the average price for days 1 to 25, the average price for days 2 to 26, etc.

The constant period of the moving averages can be assigned by any length of time but it depends on the goal we set whether we want a short term or a long term analysis. The shorter the time frame the more sensitive the movement is. The fast line is used to explain the short term moving averages which have a quicker response to current action because it uses less days in average [59]. Therefore, a large change in the value on any days will have a large impact on the short term moving average [58].

In his research work, we have a look at two commonly used moving averages techniques: simple moving average (SMA), and cumulative moving average (CMA). The data in these techniques are given the same equal weight to all data in the observation. Besides these two techniques, there are other techniques such weighted moving average (WMA) and exponential moving average (EMA) that assigned the weight to the most recent data. However, they are not in our scope of study in this research.

Simple moving average which sometimes referred to as the arithmetic moving average is the most commonly used moving average technique. To calculate SMA, a set of data with constant period is selected. The data in the set are added then divided by the number of data in that selected period as shown in Equation (2.2).

$$SMA_n = \frac{1}{n}\sum_{i=1}^{n} Data_i \qquad (2.2)$$

To calculate a successive data of the SMA, another equation is employed as shown in Equation (2.3). As suggested by the equation, the oldest data are replaced with a new data in the calculation of the successive SMA.

$$SMA_{new} = SMA_{old} - \frac{Data_{oldest}}{n} + \frac{Data_{newest}}{n} \qquad (2.3)$$

A cumulative moving average is similar to SMA in the sense that it gives an equal weight to all the data in the observation, but to calculate the CMA, all data both old and current data are considered. It is the average of all data points in the observation. When a new data is obtained, a successive data point of CMA is equal to the previous cumulative average plus the difference between the latest data and the previous average divided by the number of points received so far as shown in Equation (2.4) [55].

$$CMA_{i+1} = \frac{Data_{i+1} + i * CMA_i}{i+1} = CMA_i + \frac{Data_{i+1} - CMA_i}{i+1} \tag{2.4}$$

In our proposed work, SMA and CMA method are tested to identify the application idleness for each computer in the experimental environment. The idle time history data of each application are retrieved. SMA and CMA method are applied to predict the ideal event for the idle time historical data. For SMA method, three constant periods are tested which are fifteen constant period, twenty constant period, and thirty constant period. On top of that, the twenty constant period is implemented in the proposed power management application to generate the timeout to control the computer power states. The reason of choosing SMA and CMA for the proposed work is because we want to keep the computational complexity as low as possible to reduce the computational overhead which may lead to additional power consumption. It is also hoped that it will give a good prediction result.

CHAPTER 3

RESEARCH METHODOLOGY

This research consists of two methodologies: research methodology and development methodology. This chapter describes the research methodology that was used to achieve the objectives of the research work.

The research methodology for this research work is illustrated in Figure 3.1. It is divided into four major phases: project initiation, experimental setup for the power transitioning event, experiment setup for application idle time, and evaluation phase. The details of each phase are presented in the following sections.

3.1. Project Initiation

This phase marks the beginning of the research. Three main activities are performed in this phase: literature search, request permission, and identify problems.

3.1.1. Literature Search

A thorough study on the core components and overviews of power management were carried out. A literature search is performed to browse for information related to following questions:

- Why Green IT is introduced to the IT infrastructure?
- What is computer power management and its connection with green IT?
- What are the general problems and barriers in implementing computer power management and how can these problems be solved?

- What are the motivating factors for people to realize the importance of their contribution and involvement in reducing power consumption?

- What are the related works in this research field and what are the opportunities that can be used to improve and enhance the current built-in power management application?

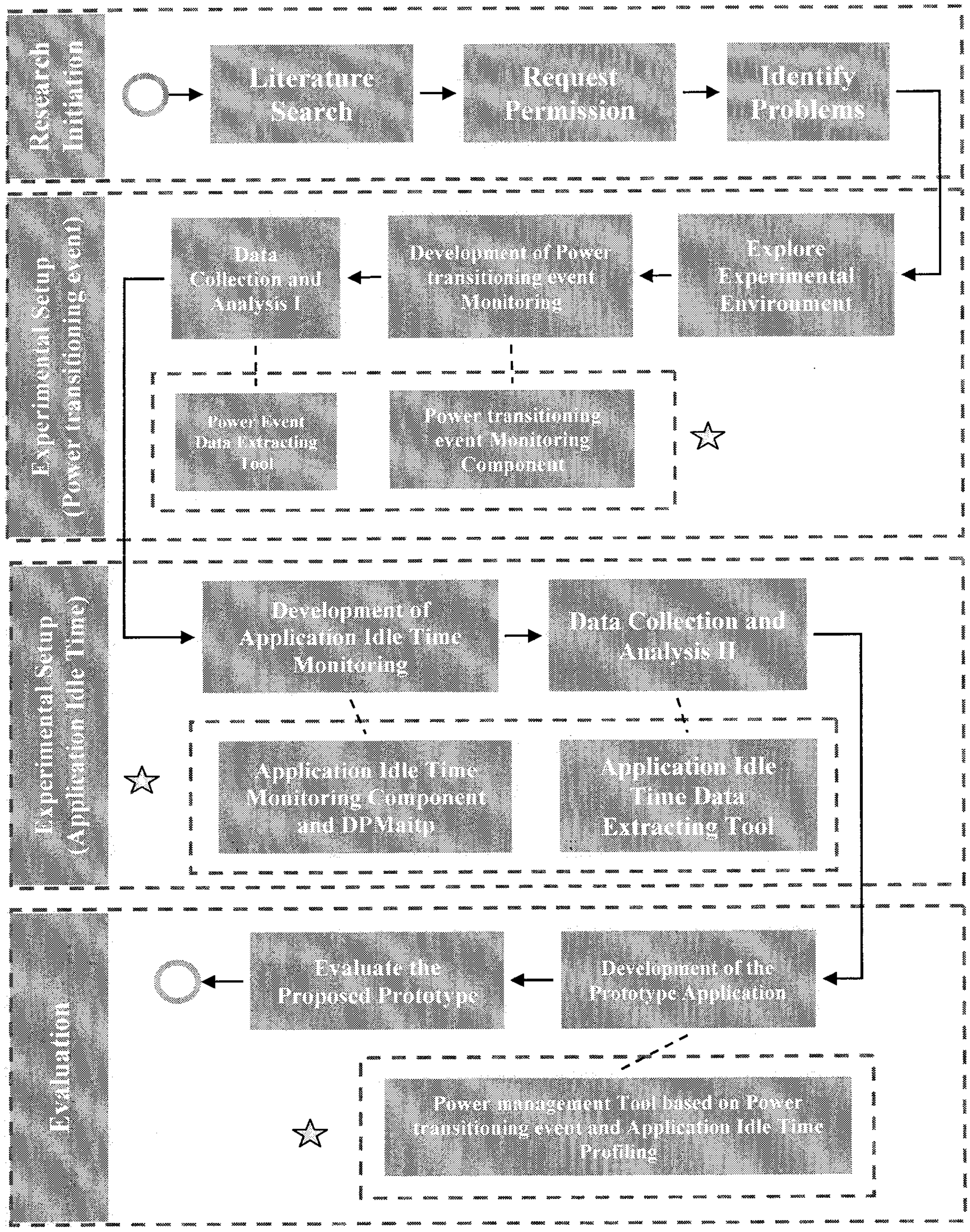


Figure 3.1. Research activities for the research work.

3.1.2. Request Permission

A request for a permission to set up the experiment in the labs was submitted to the Information Technology Media Service (ITMS) Center of Universiti Teknologi PETRONAS. The request was approved with some limitations. This is because computers in some labs are not under the university management, that is, under third party ownership. They are outsourced and managed by external parties. Therefore, the authorization is divided into two permission levels: observation and experiment level.

For observation level, the computers in the lab are only allowed to access and to be inspected but are not allowed to install any software or to make any changes to the computer setting or configuration.

Three labs were permitted to be used as set-up for the experiment level with full authority. However, only a practical computer lab with 40 PCs located in block two of Universiti Teknologi PETRONAS (UTP) can be used because of some technical problems associated with the built-in software called "steady state" in other computer labs. This problem causes the state of the computer to resume to an earlier state if users shut down their computer without saving its latest state. Moreover, this system software deletes some important information such as power on/off event and standby event that are stored in the Windows event log which is the important information for this research. These data are supposed to be used for analyzing the computer usage in the lab.

Timetables of the experimental computer lab are collected and kept for future references. These timetables are used for comparing between the duration of lab session and the real usage of the computers in the lab because by depending on the log files alone, it is hard to identify the reliable time frame whereby computers are fully-utilized.

The computers in the experimental lab are arranged with the layout depicted in Figure 3.2. A main computer for lectures or instructors is located in front at the center of the room along with four rows and three columns of student's computer desks. Some of the desks accommodate three computers and some accommodate four computers. Each computer is uniquely labelled from the number starting at 1 to 40.

The arrangement of the computers in the lab holds important information that can help to pinpoint the locations of the computers which have been accessed frequently. Therefore, the data provided by those computers are more reliable.

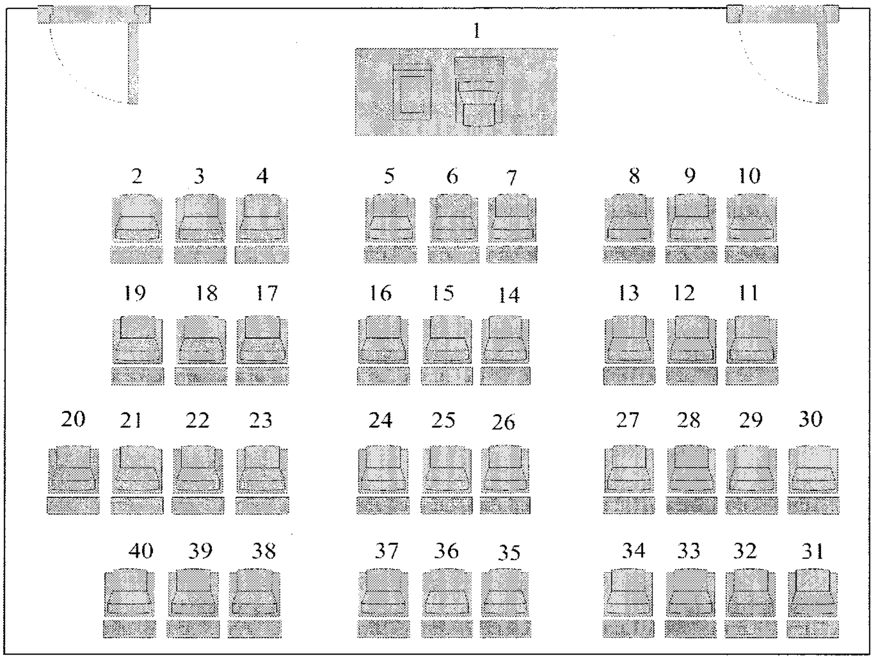


Figure 3.2. The arrangement of the computers in the experimental lab.

3.1.3. Identify Problems

In this activity, the utilization of computers and the power management tool in the labs were looked into. The shutdown policies implemented by the lab technicians were also being considered.

For utilization of computer and the power management tool, the problems were identified through observation and inspection in the practical computer labs. Two different situations were taken into consideration: at-the-end-of-the-practice session and after-the-office hour. The reason of choosing these two situations is because university computer labs are used for student experiment works and they are utilized only in some specific time based on the timetable's schedule. Moreover, there is a break time in between each lab session. Therefore, it covers the break time for both at-the-end-of-the-practice session and after-the-office hour.

Two shutdown policies are implemented for the computers in the lab: manual and automatic shutdown policy. The manual shutdown policy requires the technician to go to each computer lab to manually power off each machine after the operating hour. For the automatic shutdown policy, remote shutdown batch file and built-in shutdown scheduling are used. The remote shutdown batch file is issued by technician to force to shut down multiple machines at 5pm without allowing any users to cancel the action. The content of the remote shutdown batch file is listed in Appendix A. Likewise, a shutdown scheduling is configured on some of the computers in some labs. It issues the shutdown command automatically at exactly 5pm every day. In addition, Appendix A presents how to set up a simple shutdown scheduler for Windows XP with a reference to set up for later versions of Window-based OS.

By implementing the above policies, it was noticed that some problems still occurred. For instance, some users are still using the computers while the shutdown command is executed. This forces those users to save their work as quickly as possible before the computer is automatically turned off. Even though users could resume their work by rebooting their computer, it will take some time before the computer resumes to its working state. Thus, it will irritate the user.

In addition, if users use their computer after the operating hour, and forget or neglect to turn off, then computers without power management feature enabled will be left powering on leading to power waste for the whole night. It is because there are no policies to handle the computers after the operating hour. Moreover, the two policies introduced earlier are only applied for the-after-operating hour situation but not for the after-the-lab session ended situation. Thus, more power will be wasted if the computers are on after the lab session ended.

To have a clearer understanding of the effectiveness of current policies, we conduct a further observation on the user's computer usage behavior through power transitioning events. These events give us a better view of how often and how long an individual computer has been used. At the next session, we look at how the power transitioning event data are obtained from the experimental lab.

3.2. Experimental Setup for Power transitioning event

In this phase, a thorough assessment of the experimental lab is performed along with the development of power transitioning event monitoring component that generate power transitioning event data storing in the log files. These log files store vital information for analyzing the computer utilization and user's usage activities.

3.2.1. Explore Experimental Environment

Even though the environment has been revisited several times for the observation and inspection of computer utilization and their power management tool utilization, the detailed information about individual computer is still unknown such as operating system and hardware specification, and power specification.

The interview was conducted with lab technicians who handle particular labs for getting detailed information on each computer and how they manage all computers. The installation of operating system and software are done using image file which includes most of the frequently used software and applications such as word processing, power point, development tools, browsers, database applications, etc. Windows XP professional service pack 3 is the main operating system and the hardware specification is listed in Table 3.1. Moreover, there are also a few computer labs installed with a Linux based OS and the Windows-based OS as a dual boot machine. However, Windows-based environment is the primary focus of the experiment for this research because the utilization of the lab installed with Linux based OS is limited.

Table 3.1. Hardware specification of the computers in the experimental lab.

Desktop	CRT
• **Branch: COMPAQ EVO D51C** • **CPU: Intel ® Pentium 4–2.4 GHz** • **RAM: 1GB—1.5GB**	• **Branch: COMPAQ V7550 Color Monitor** • **Manufacture Year: 2003** • **AC 100-240V ~60/50Hz 1.4A**

The actual environment was later explored to verify the information provided by the lab technicians by identifying the computer's operating system and hardware specifications. The operating system specification is collected from computers in the lab which are in the same subnet using a free software called Network Mapper (NMAP). On the other hand, to determine the hardware specification of the computers, a manual reviewing is performed by writing down the specification from a plate attached to each computer. Most of the information from our inspection matches the information given by the lab technicians.

The hardware specification of computers in the experimental lab with full authority mentioned in Section 3.1.2 is shown in Table 3.1. RAM is the only component with different specification varying from 1 to 1.5 Gigabytes (GB).

Table 3.2. Power Meter ENER007 Specification

Description	
Power Meter (ENER007)	Voltage: 240V AC 50Hz Maximum current: 13A Maximum load: 3120W Typical Power Consumption: < 0.5W
Measuring Range	Voltage: 200-276V AC, +/-0.5% Current: 0.005-16A, +/-0.5% Power: 0.1-3680W, +/-0.5% Accumulative electricity usage: 0-9999.9 kWh Frequency: 45-65Hz Clock accuracy +/- 1 minute per month
Main Key Features (Display information)	Current clock in 12 or 24 hour mode Voltage in volts Frequency in Hz Current being consumed in amp Power being consumed in watts Accumulated total electricity consumed in kilowatt-hours (kWh) Total on time Total costed on time Carbon footprint in Kg of CO_2

After confirming the software and hardware specifications of computers, the power measurement of some computers was done using a power meter ENER007 from Energenie with its specification stated in Table 3.2. This is to understand how the power consumption is influenced by different computer power states and by their display screen power modes.

Among all information provided by the ENER007, computer's power and maximum power are the only parameters in which we are interested. Since all computers and their computer's monitor screen have the same specification except RAM, we randomly select five computers for the power measurement and get the final result by averaging all the measured power.

We consider three different states, which are Off, Active, and Low power state for both computers and their screen. The Off state is the state when computers and their screen are powered off, but its power cord is still attached to the power outlet. In this state, there is only a small amount of power consumed by the computers and their display screen. For the Active state, computers as well as their screen are powered on. It involves some working tasks such as word processing, and browsing the Internet. The Low power state is when the computer and its screen switch to a low power state which is a standby state in this case.

In addition to the power measurement of the three states above, we have two extra power measurements for computer's monitor screen: the fully white and black screen. This measurement is performed because we want to know whether the color of the screen will also affect the power consumption of the whole computer system. The result of these measurements will be discussed in Chapter 5.

3.2.2. Development of Power Transitioning Event Monitoring

To understand and to have a better view of the computer usage in the lab, a power transitioning event monitoring component is designed and implemented. This component is a service application which is running on the system level collecting information such as power transitioning events and their time stamp. These information is stored in the log files.

This service is designed based on the computers in the experimental environment. C#.Net framework is used for coding the service application because the computers in the lab are running on Window-based OS. On top of that, C#.Net framework eases the process of the development because of its rich application programming interface (API).

Basically, the information related to the power transitioning event can also be retrieved from the Windows event log, but a few settings are required to be enabled. However, the information given by the event log is limited: new parameters cannot be added to the log. In contrast, by logging the data to the log file, it gives more freedom on data formatting and new parameters can be introduced. Therefore, only necessary information needs to be filtered so that the data can be easily translated and analyzed in real-time mode.

A timestamp and power events such as computer's lock, unlock, logon, logoff, resume from standby, standby, shutdown, and startup are written to three different log files: "session_powerevent.txt", "suspend_powerevent.txt", and "shutdown_powerevent.txt" whenever any of the events depicted in Figure 3.1 is triggered by the system.

It is important to split the information about the power transitioning event into three different log files. It accelerates the analyzing process because the amount of data keeps growing which shall slow down the processing and extracting speed of the information in the log files. In general, computers need a longer time to process a big file as compared to a small file because they retrieve the information from file to memory before processing and returning the result back. Therefore, the information in the log file that has already been classified in the same group and split into different log file can be extracted and translated into useful information much easier and faster. Moreover, it helps to reduce the memory burden to store large data when extracting the information from the log file and translating into useful information.

The events are divided into two levels which are the main events and sub-events. Figure 3.1 shows the structure of the events generated by the power transitioning event monitoring service. There are four main events generated namely *OnPowerEvent, OnStart, OnSessionChange, and OnShutdown.* Among all four main events, only *OnPowerEvent*and and *OnSessionChange* contain sub-events.

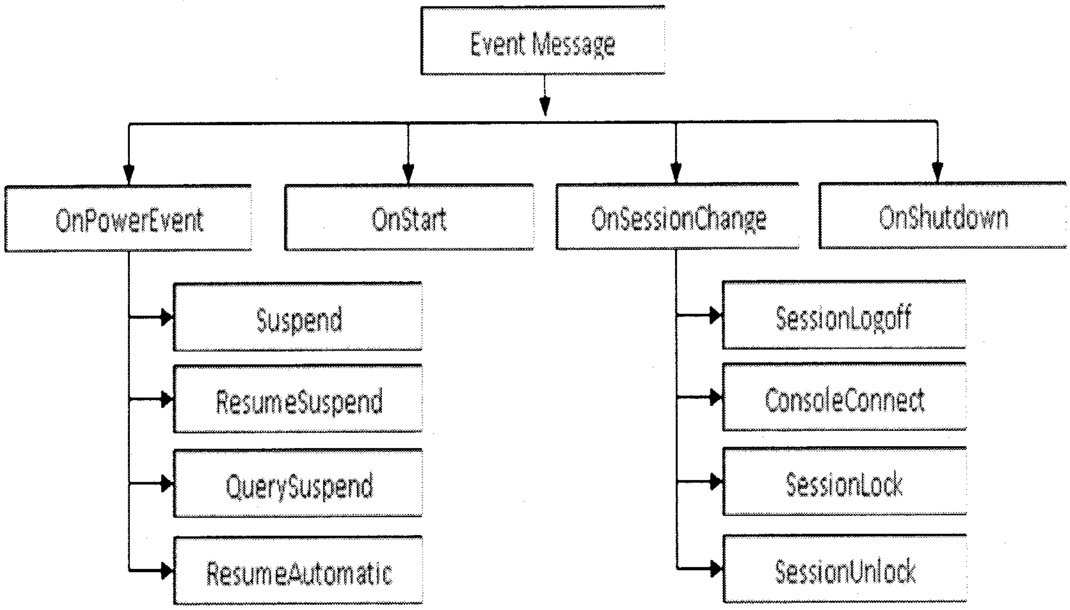


Figure 3.1. Power events logged by power transitioning event monitoring service

OnPowerEvent is a main event that is passed to the computer's system in the form of *PowerBroadcastStatus*message when the power status of computers has changed. In general, this event occurs when the system transitions into suspend mode, when it resumes from suspend mode, or when it receives low battery status in a laptop computer [60].

PowerBroadcastStatus message contains one of the four sub-main events: *Suspend, ResumeSuspend, QuerySuspend, and ResumeAutomatic* which will be written into "session_powerevent.txt" log file. On top of that, the *PowerBroadcastStatus* message also carries other event messages but they are filtered out to reduce the burden of storing unnecessary data on a single log file. It is because the file size affects the speed of retrieving and translating the information for analysis so the size of the file shall be relatively small.

QuerySuspend event is broadcasted to request the permission from all applications and drivers to switch the computer to suspend mode [61]. The applications and drivers that are ready for suspension will reply to the request by returning a true message to the system. After the system receives the agreement from all applications and drivers, suspend event will be sent to acknowledge the request and to inform the system in the form of a message that carries a true value before the suspend operation takes place.

ResumeAutomatic event is raised when computers wake up automatically from suspended mode after receiving a wake up signals from the system [61]. On top of that, after broadcasting *ResumeAutomatic* event, if the system receives any action from the user, *ResumeSuspended* event will be broadcasted to inform all applications and drivers that they can resume their normal operation and they are ready to handle any works.

*OnStart*event is a main event which is attached to services. This event will be raised to inform the system that the services have been started and they are ready to handle any activities. In contrast, *OnShutdown* event is issued when the services receive a shutdown request from the system. The system terminates all services and processes preparing to shut down. Moreover, the message is sent to inform the running services that it is going to the shutdown mode. These two events are written in "shutdown_powerevent.txt".

OnSessionChange event is a main event that notifies the system when the terminal server session broadcasts the change event to inform the applications [62]. The description of this event is carried by the *SessionChangeDescription* which contains four sub-main event messages: *SessionLock, SessionUnlock, SessionLogoff, and Session Logon.*These events are logged in "session_powerevent.txt". As the name implies, *SessionLock and SessionUnlock* is related to events which are used for notifying the system when the users lock or unlock their computers. For *SessionLogon* and *SessionLogoff*, they are broadcasted when logon and logoff action are issued respectively.

The content of "suspend_powerevent.txt" and "session_powerevent.txt" log file contains three fields which are the date and time when the event happens, the message of the main event, and the message of the sub-main event. For the content of the "shutdown_powerevent.txt" log file, it only contains two fields— the first two fields specified in the above two log files. The sample content of the three log files is shown in Table 3.3.

Table 3.3. The format and content of the power transitioning event log files.

Log file's Name	Date and Time	Main Event	Sub-main Event
shutdown_powerevent.txt	8/1/2012 8:42:00 8/1/2012 14:57:00	(OnStart) (OnShutdown)	
session_powerevent.txt	8/4/2012 16:41:00 8/4/2012 17:57:00 7/30/2012 8:38:00 7/30/2012 9:23:00	(OnSessionChange) (OnSessionChange) (OnSessionChange) (OnSessionChange)	SessionLock SessionUnlock SessionLogon SessionLogoff
suspend_powerevent.txt	11/7/2012 17:04:00 11/7/2012 17:05:00 11/7/2012 17:05:00 11/26/2012 10:46:00	(OnPowerEvent) (OnPowerEvent) (OnPowerEvent) (OnPowerEvent)	QuerySuspend ResumeAutomatic ResumeSuspend Suspend

In the designing and implementing of power transitioning event monitoring activity, we delivered a software package called power transitioning event service. This application was deployed on all the computers in the experimental lab from the administrative computer using its MSI package and a trail version of remote software deployment tool called EMCO remote deployment [63]. This application runs in the background, captures and writes the power transitioning event messages as depicted in Figure 3.1 to the log files with the content shown in Table 3.3. Moreover, it will filter the unnecessary events such as power status change event and power setting change event to keep the file size relatively small.

3.2.3. Data Collection and Analysis I

The data generated by the power transitioning event service collected and stored in three different log files are raw data. Moreover, all log files are located in individual computer. Thus, data collection and translation process are needed to extract the important information for further analysis.

In the initial stage of the experiment, data from the power event are collected and analyzed for two weeks to investigate the utilization of the experimental lab to see if the chosen lab gives a favorable result and to identify the location of computers that are frequently used. Furthermore, exploring other problems associated with power management implementation in that lab was also done. The period for logging these data is extended substantially afterward.

To collect the log files, a shared folder is created in a master computer. It is the machine used for storing all log files from the computers in the experimental lab. A batch file containing the script to copy log files from individual computer to the shared folder is also created. This batch file is put in the shared folder location. The master computer remotely executes it with administrative right using the PSEXEC command. The detail usage of the batch file and the PSEXEC application are given in Appendix A.

An information extracting tool is designed to extract and to translate the raw data in the log files into important information. This tool allows the user to browse for the location of the log file and to select which log file to be analyzed. The content of the extracted data depends on which log file the users selected to be extracted because each log file contains different information. The format of the extracted data is shown in Table 3.4. The result of the analysis can be exported to excel file and kept for future reference and analysis.

The extracted data comprises of information related to how long an individual computer has been used, how long it transitions to a low power state, and how often the users access their computer. The extracted data are grouped into usage category of the day of the week. Then, the computer's utilization is critically compared among all computers in the lab.

The contents of "session_powerevent.txt" log file are manually analyzed by importing them to excel format. The Session Lock and SessionLogOff message are investigated to see how often users leave their workstation and in which period of time.

Table 3.4. The format of extracting data from respective log files.

Log file's Name				
shutdown_power event.txt	Date	Start Time	End Time	On Time
suspend_powerev ent.txt	Date	Suspend	ResumeSuspend	Suspend Duration

3.3. Experimental Setup for Application Idle Time

In general, idle time is defined as the total time a computer or device has been powered on, but has not been used [64]. However, the term application idle time is used throughout this research which has a related meaning of idle time. It refers to the period when the application is in focus or in the foreground, but there is no user interaction or activity.

After setting up the experiment to explore the power transitioning event, another experiment on application idle time was conducted. In this experiment, two main packages are delivered: application idle time monitoring component and application idle time data extracting tool.

The development of the application idle time monitoring component, and the process of collecting and analyzing of the generated data from the application idle time monitoring component are discussed in the following sections.

3.3.1. Development of Application Idle Time Monitoring Component

The application idle time monitoring component is written in C#.Net. Apart from the power transitioning event component which is a service application, the application idle time monitoring component is a user level application running in the background collecting individual computer idle time along with a time stamp, application name, and application title as shown in Table 3.5. These data are written into three log files "appIdlelog.txt", "defaultIdleLog.txt", and "penaltyIdleLog.txt".

Table 3.5. The content of the application idle time log files.

Log file's Name	Date and Time	Idle Time (sec)	Application Title	Application Name
appIdlelog.txt	11/28/2012 13:40:07	34	Chapter 1.docx - Microsoft Word	WINWORD
defaultIdleLog.txt	11/27/2012 17:37:26	40	Khmer Sport Fans - Mozilla Firefox	firefox
penaltyIdleLog.txt	11/28/2012 13:48:12	55	Chapter 1.docx - Microsoft Word	WINWORD

Windows-based OS does not broadcast any idle events or store the idle time of the running applications or processes. Only system idle time is stored and can be accessed. To get the application idle time, a combination of the system idle time and the foreground application detailed information is used. A timer with one second interval is implemented to capture the system idle time and the foreground application when the user becomes inactive for a period of time, exceeding the static threshold of twenty seconds. In general, on user level the application on the computer is either a background process application or a graphical user interface (GUI) process application. The background process runs transparently to users. It is similar to service application but it runs in user level. For a GUI process, it is attached to a GUI for displaying application contents to the user or for getting the feedback from the user.

Application idle time monitoring component contains two main processes: a background process and a graphical user interface process. The background process runs in the background monitoring and logging the foreground application detailed information when the foreground application becomes idle. The foreground application is the application that the user actively interacts with or the application that is currently in focus. For the GUI process, it is attached to a form designed to display the information of the last idle foreground application such as the process name, process title, and idle time as shown in Figure 3.2. The information displayed on the form changes dynamically when the background process detects a new foreground application event being logged by the application idle time monitoring process.

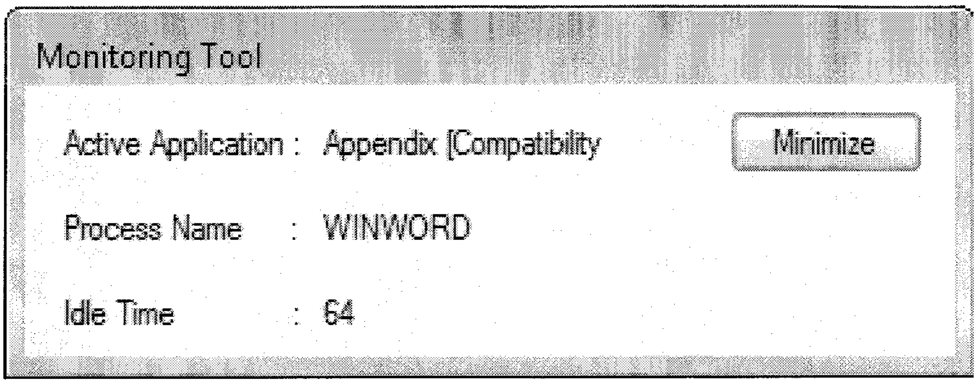


Figure 3.2. The GUI displays information about the latest idle application.

3.3.2. Development of Display Power Management based on Application Idle Time Profiling (DPMaitp)

A display power management based on application idle time profiling (DPMaitp) utilizing the application idle time monitoring component is implemented. It is used to control the computer display screen power state. It handles when to switch the display screen to a low power state.

DPMaitp has three power states: Active state, Screen Dimming state, and Screen Off/Standby state as shown in Figure 3.3. It is similar to the traditional display power management policy which is bundled with most of nowadays PCs. However, the proposed work provides additional feature such an application idle time profiling mechanism and a method to learn from false user interaction to enhance the traditional power management application in term of power consumption and user irritation.

DPMaitp has two main components which are the application idle time monitoring component and the action enforcer component. The application idle time monitoring component monitors the idle foreground application and logs that activity to the log file. In general, Windows based OS only provides the system idle time. Thus, to get the idle time for a particular application, a system idle time and a foreground application are combined to generate a unique application idle time.

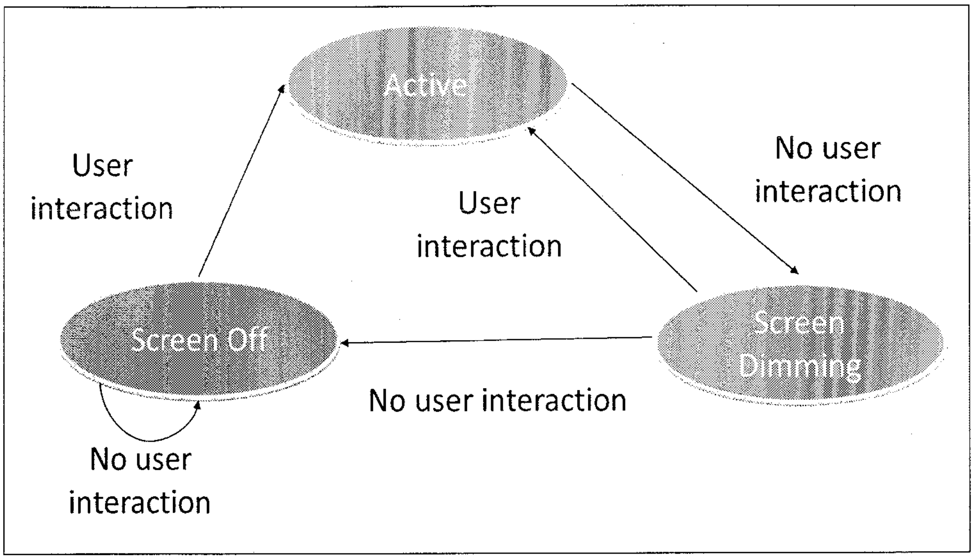


Figure 3.3. State transition diagram of display power management based on application idle time profiling.

System idle time can be calculated by using system uptime and system last input ticks counter as shown in Equation (3.1). The OS overrides the system last input ticks counter when the PC receives any user interaction. The system idle tick counter can be captured through platform invocation services (P/Invoke) by implementing "GetLastInputInfo" function in the Windows API ("user32.dll"). This function has a parameter referencing to a data structure called "LASTINPUTINFO" which contains the size of the structure and the tick count when receiving last input [65].

$$SysIdleTime = (SysUpTime - LastInputTick) / 1000 \quad (3.1)$$

Active or foreground application title and name are captured when the system becomes idle. Three functions are needed to be implemented from the API ("user32.dll"): "GetWindowThreadProcessID" "GetForegroundWindow", and "GetWindowText".

Three log files are generated by application idle time profiling component: default application idle time log, penalty log, and application idle time profiling log. Only application idle time profiling log is used for generating display's timeout setting while the other logs are used for performance and prediction accuracy evaluation.

Table 3.6 shows the content of the log file generated by DPMaitp. It includes four main fields: logged time, idle duration, application title, and application name.

The idle duration information will be analyzed to generate the timeout for the application using the simple moving average. The timeout for each application is written into a setting file. The content of the setting file is shown in Table 3.6.

Table 3.6. Content of application idle time setting file

	Description	Example
Head	[ApplicationProcessName]	[firefox]
Body	Last Application Name	appName= mozilla firefox
	Total Idle Time	appTotalIdleTime=35159
	Count/Occurrence	count=343
	Average Idle Time	averageIdleTime=102
	Timeout	timeout=51.4
	Penalty	penalty=0

3.3.3. Data Collection and Analysis II

In this data collection activity, the same method is used as in the data collection and analysis I activity. The log files, generated by the application idle time monitoring process, are copied to the shared folder on the master computer. The batch file for copying the log files created in the early activity is edited to append the script to allow the computer to copy its log files to the shared folder. Then, the PSEXEC application is executed remotely from the master computer to run the script in the batch file.

The idle detailed information from each application written in the application idle time log files is scattered around based on the time stamp when the event occurs. Two methods are used for extracting the information from log files: manual extraction and automatic extraction.

For manual extraction, the log files generated in each computer are exported to excel file for filtering and analyzing the information. These extracted data can be used for analyzing the computer idle activities, and the unusual or unexpected idleness of the application.

For automatic extraction, a data extracting tool is designed. This tool extends the functionality of the data extracting tool which is implemented in the early activity to extract information about the power transitioning events. The new functionality allows this tool to group applications with the same process name. This tool counts the occurrence, and calculates the total idle time and the average idle time of each process. The extracted information is displayed on the form which is run on the GUI process. Moreover, it can be exported to an excel format file. These extracted data can be used to identify groups of frequently used applications, and frequency of their usage. The content of the extracted information from the three log files generated by the data extracting tool is shown in Table 3.7. It contains four fields such as application's name, total idle time, counter (the occurrence of the application's idleness), and average idle time of the application. These three log files have the same contents.

From the computer idle activity analysis, the idle period and the active period of computers in the lab are critically compared. This comparison gives an insight of the real utilization time of the computer. Moreover, this helps us to understand frequency of the user inactivity.

From the unusual idleness of application analysis, the moving average techniques such as three simple moving averages and a cumulative moving average are used to smooth the application idle time data and to detect the unusual the application's idleness. For SMA, a few constant periods are tested such as 15, 20, and 30. This constant periodis selected based on the usage frequency of the application.

Table 3.7 depicts information regarding four applications: devenv (Visual Studio), notepad, AcroRd32 (Acrobat Reader), chome (browser). All applications have different behavior regarding to the frequency of usage and idleness of the application. For instance, notepad which is a plain text editor has the lowest average idle time (36 seconds) and usage (29 times) among all the application in the list. This application has been used less as compared to other applications. Moreover, it shows that the idleness of the application was around 36 seconds which is an usual idle behavior for the text editing application. Therefore, the power management application can act accordingly when the application becomes idle relatively longer than usual.

Table 3.7. Information extracted from the application idle time log files.

Application Name	Total Idle Time (Seconds)	Counter	Average Idle Time (Seconds)
Devenv	**47080**	**628**	**74**
notepad	**1058**	**29**	**36**
:	**:**	**:**	**:**
:	**:**	**:**	**:**
AcroRd32	**3727**	**49**	**76**
Chrome	**215886**	**1710**	**126**

3.4. Evaluation

Two main activities were performed in this phase: development of a prototype application and the evaluation of the research work. The evaluation of the research work depends on the prototype application because the data for evaluation processes are generated by this application.

3.4.1. Development of The Prototype Application

A prototype application is developed, extending DPMaitp and utilizing the proposed components introduced in the earlier research activities as part of its functionality. This application is called PMaitptep which means power management based on application idle time and power transitioning event profiling.

The PMaitptep is used for assessing whether the proposed components can be applied in the real practical environment to detect the user activities and the idleness of the user and the foreground application. Moreover, it extends the functionalities of the built-in power management tool in the OS by improving the power state transitioning decision. The power state transitioning decision of the PMaitptep depends on the dynamic timeout which is generated based on the data in the log files. This is better than a static timeout preset by the user which is offered by the built-in power management tool in the OS.

The dynamic timeout of the application refers to the timeout assigned to each application whereby it is calculated based on the analysis of the historical data generated by the power transitioning event and application idle time monitoring component. Two dynamic timeouts are implemented to control the power states of the system. One of which is used for controlling the display screen power states and another one is used for controlling the computer system power states. When the PMaitptep application is executed, it analyzes the historical data in the log files and generates the two dynamic timeout.

The PMaitptep is developed using C#.Net framework. This application handles four power states: Active State, Screen Dimming State, Screen Standby State, and Computer Standby State as shown in Figure 3.4.

The active state is a state whereby the user is working with the application such as word processors, excel, browser, etc. The system will stay in the Active state as long as users actively interact with their computer through keyboard or mouse.

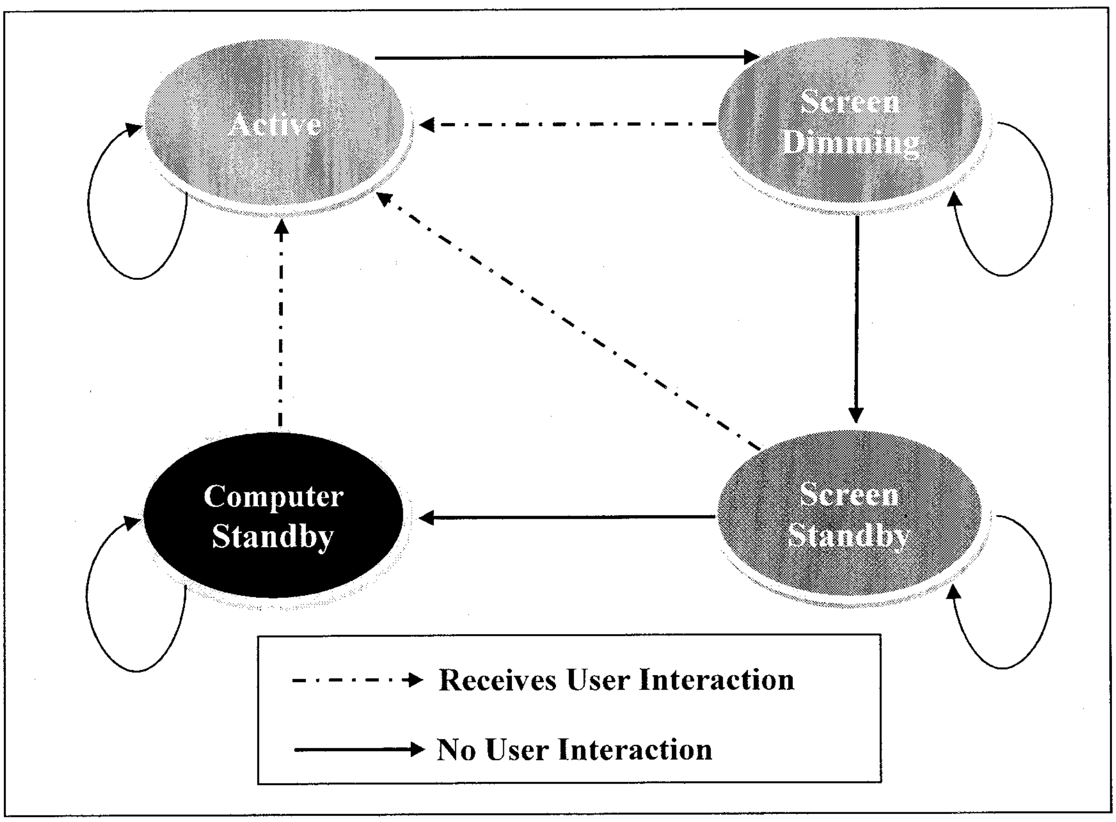


Figure 3.4. Power state transition diagram of PMaitptep

The screen dimming state is the state whereby the display screen of the computer lowers down its brightness to one third of the current brightness. The display screen transitions to this state whenever the computer becomes inactive for a period of time that exceeds the timeout of the foreground application in the setting file. This state only lasts for 10 seconds. This state is used for detecting user attentiveness and for reducing the timeout false prediction. If the computer receives any user interactions such as moving the mouse cursor or typing any character on the keyboard in between the 10-second interval of the screen dimming state, the computer and its display screen power state will return to the active state and the timeout of the foreground application is increased which either double the current application's timeout or the maximum of historical idle times of that particular application.

The screen standby state is the state whereby the display screen switches from the screen dimming state to the screen standby state. In this state, the display screen is turned off but it still consumes a small amount of power. It needs this power to keep its components warm so that it will switch back to the working state quickly without spending much time to warm all components. After this state kicks in, the PMaitptep compares how long the individual computer had been on with the timeout generated by the power transitioning event. If the on-time exceeds the timeout, the computer will switch to the computer standby state.

For GUI process, a form is designed to visualize the dynamic timeout information assigned for each application. The dynamic timeout is recalculated for the application that becomes idle and being logged by the PMaiptep.

3.4.2. Evaluation of the Proposed Work

To evaluate the proposed work, some analysis on the time saving of the prototype power management application, prediction accuracy, and user irritation were performed.

3.4.2.1. Time Saving

Time is the main parameter to determine how much power is consumed by the computer. The longer the time a computer in active mode the more power it consumes. Therefore, time is a good candidate for evaluating how well the power management tool performs.

Time saving analysis is divided into two: desktop time saving and display screen time saving. For desktop time saving, we look at how often the computer transitions to a low power state and how long it stays in that state as compared to the built-in power management tool. For display screen time saving, we look at how often the screen becomes inactive and how long it stays in the low power state. Associated with time saving by the desktop and its display screen, the overall time saving is calculated for all computers in the experimental environment.

3.4.2.2. Prediction Accuracy

In general, the system can produce a good prediction by using a good and complex algorithm. In this research, the moving averages and the false prediction reduction method are introduced to control the power transitioning decision for the computer. These methods need to produce a good prediction result in an acceptable range with minimal interruption to the user.

The simple moving average method is used to identify the unusual idle event and to generate the timeout for applications. The last twenty idle time periods are retrieved from the log file to calculate the predicted timeout for the application. If the idle time period is less than the twenty periods, the timeout is the average of all the past idle time value. The comparison between the predicted timeout and actual system idle time is done by PMaitptep to decide whether to switch the computer and its display to low power states.

For example, a moving average method with five constant periods is implemented and tabulated result of idle times is listed in Table 3.8. The state transition decision depends on the predicted timeout and the actual system idle time. For the third and fourth period, the actual system timeout exceeds the predicted timeout so PMaitptep switches the display screen's power to a low power state. Thus, a new timeout is generated when new idle time is logged to the log file. For the fifth period, the predicted timeout is set to double the previous timeout because the user interacts with the computer during the dimming state. Therefore, a penalty is set for that particular application and a new timeout is generated which is double the previous timeout.

Table 3.8. Example of moving average method with five constant periods to decide when to switch to low power state.

Actual System Idle time	Predicted Timeout	Screen Dimming	Screen Off
35	-	-	-
34	35	No	No
60	35	Yes	Yes
40	36	Yes	No
35	52	No	No
34	35	No	No

3.4.2.3. User Irritation

In general, irritation is the feeling of being annoyed, angry, or impatient because of something that happens repeatedly for a long time [66], [67]. In this research context, user irritation is defined as the events or activities that make the users feel irritated. The user irritation has a close relationship with the prediction accuracy because users become annoyed once the application produces many false predictions. For instance, in the worst case, the screen is turned off while the user is still actively using the computer.

Even though the power management tool helps to reduce power consumption in the organization, if its implementation makes the users feel irritated to some unacceptable level, this will affect their work. Therefore, user irritation level needs to be as low as possible.

CHAPTER 4

DEVELOPMENT METHODOLOGY

This chapter describes the development methodology of the research work. The development process that consists of two proposed components, tool, and applications mentioned in Chapter 3 are discussed. The functionality and the system flowchart of the two proposed components are discussed in detail i.e. the power transitioning event monitoring component and the application idle time monitoring component along with the display power management (based on application idle time profiling). The data extracting tool which is used for extracting the data from the log files generated by the proposed components is also presented in this chapter. This chapter deliberates the functionality and the system flowchart of the power management application that utilizes the proposed components to enhance the built-in computer power management application. The development tools for this research are also discussed in detail.

4.1. Development Process

The Phased Development Methodology is chosen to develop the components, tools, and applications in this research. It consists of three major versions as shown in Figure 4.1. The main reason for choosing this methodology is because of time constraint and the overall system requirement and the concept identified.

Major phases and deliverables:

- Initiating and planning phase

    a. Conduct literature research on the topic of power management.

b. Preparing proposal defense within three months after confirming the research topic.

c. The completed proposal defense report is submitted to the CIS department along with a proposal defense request form for approval.

- Analysis phase

  a. User requirements and system requirements are collected. All the requirements are reviewed critically in each version of the development.

  b. Refinement of the requirements is performed in every version of the system if applicable.

- Design phase

  a. System flowchart for the proposed components, tool, and applications are defined along with its respective pseudocode.

- Implementation phase

  a. Project backup is created and code refinements are enhanced for every version of the system for project maintenance and system roll back.

  b. System testing is performed for every version to determine the usability of the system.

The project is developed via three major versions. In version one, it consists of the development of power transitioning event monitoring component and data extracting tool. In version two, an application idle time monitoring component and a display power management based on application idle time profiling are implemented. The data extracting tool are updated with additional functionalities. In version three, a power management application based on the two components introduced in the earlier version is created. This application is called the Power Management based on Application Idle Time and Power Transitioning Event Profiling (PMaitptep).

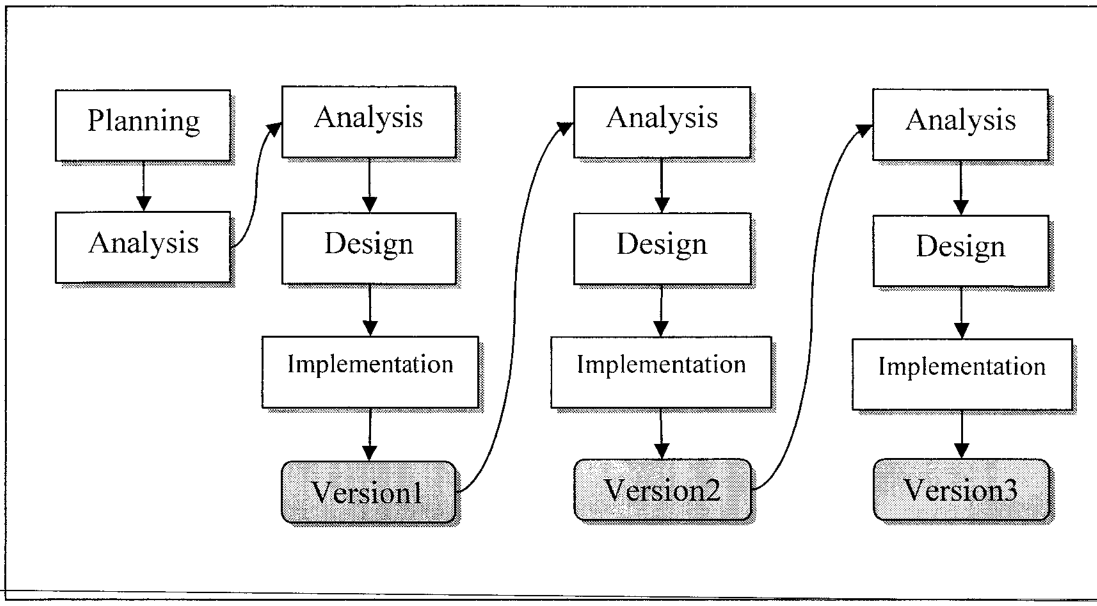


Figure 4.1. Phases Development-based Methodology.

4.2. Power Transitioning Event Monitoring Component

Power transitioning event monitoring component monitors the power event of the computer and records the detailed information about the power event in the log files. The following section shows the system architecture, pseudocode, and system flowchart of the power transitioning event monitoring component. It describes how this component works and the activities flow of the system.

4.2.1. Power Transitioning Event Monitoring Component Architecture

Power transitioning event monitoring component is a service application. Thus, it sits on a level lower than the user level application but it has higher authority. The architecture of the power transitioning event monitoring component is shown in Figure 4.2.

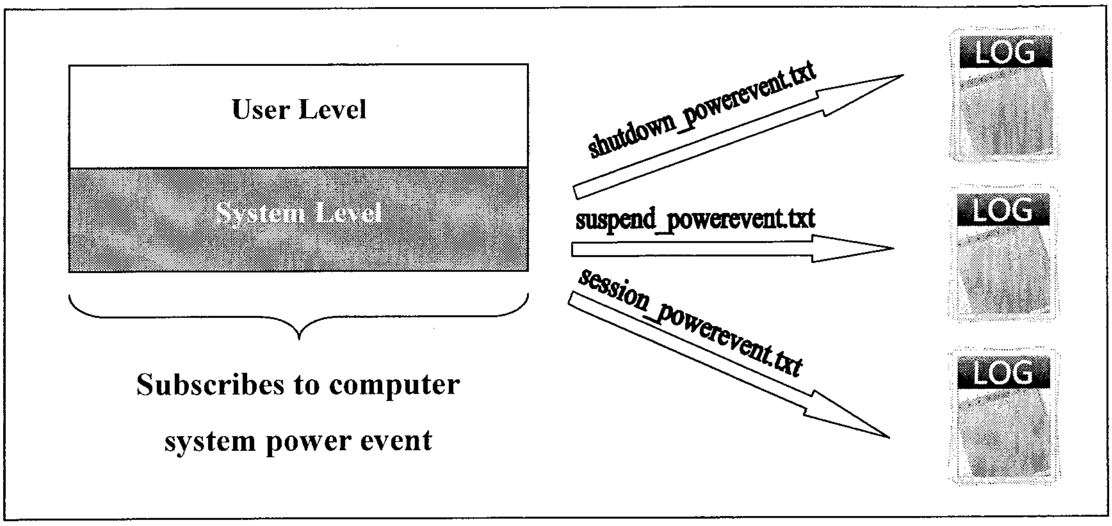


Figure 4.2. The system architecture of power event transitioning event monitoring component.

This component subscribes to computer system power events so that when any power event is broadcasted, it filters and selects the necessary messages such as computer on/off, suspend/resume suspend, lock/unlock, and logon/logoff. Then, it writes the detailed information in the log files accordingly. The component starts when the computer startup and stops when the computer shut down.

4.2.2. Pseudocode and System Flowchart of Power Transitioning Event Monitoring Component

Figure 4.3 and Figure 4.4 show respectively the pseudocode and system flowchart of the power transitioning event monitoring component. These figures show the functionalities of the component and how the component handles the power event messages. Two main functions are created to handle the monitoring and logging processes.

The monitoring function is invoked when the computer system service starts to keep track all the broadcasted power event messages. Those messages are filtered by the if-else condition clause where this clause compares the types of the messages being broadcasted. Then, the logging function is initiated to handle the logging process according to the power event broadcasted by the computer. The monitoring function keeps looping to the beginning of the code to listen and to check the power event messages broadcasted by the computer system until the computer goes to sleep, standby, or shut down mode.

```
Start
1.0 Checks computer power event
2.0 If(service_start or service_stop)
    2.1. Writesthe event to shutdown_powerevent.txt
3.0 Elseif(suspend or resume_suspend)
    3.1. Writesthe event to suspend_powerevent.txt
4.0 Elseif(lock or unlock or logoff or logon)
    4.1. Writesthe event to session_powerevent.txt
5.0 Go to 1.0
End
```

Figure 4.3. The pseudocode of the power transitioning event monitoring component.

The power transitioning event monitoring component has the ability to handle some specific event such as start event, stop event, suspend event, resume suspend event, lock event, unlock event, logon event, and logoff event. Those events are divided into three groups that will be written to three different log files: shutdown_powerevent.txt, suspend_powerevent.txt, and session_powerevent.txt. These log files contain detailed information about the power event which will be used to determine the usage pattern and activities in each computer.

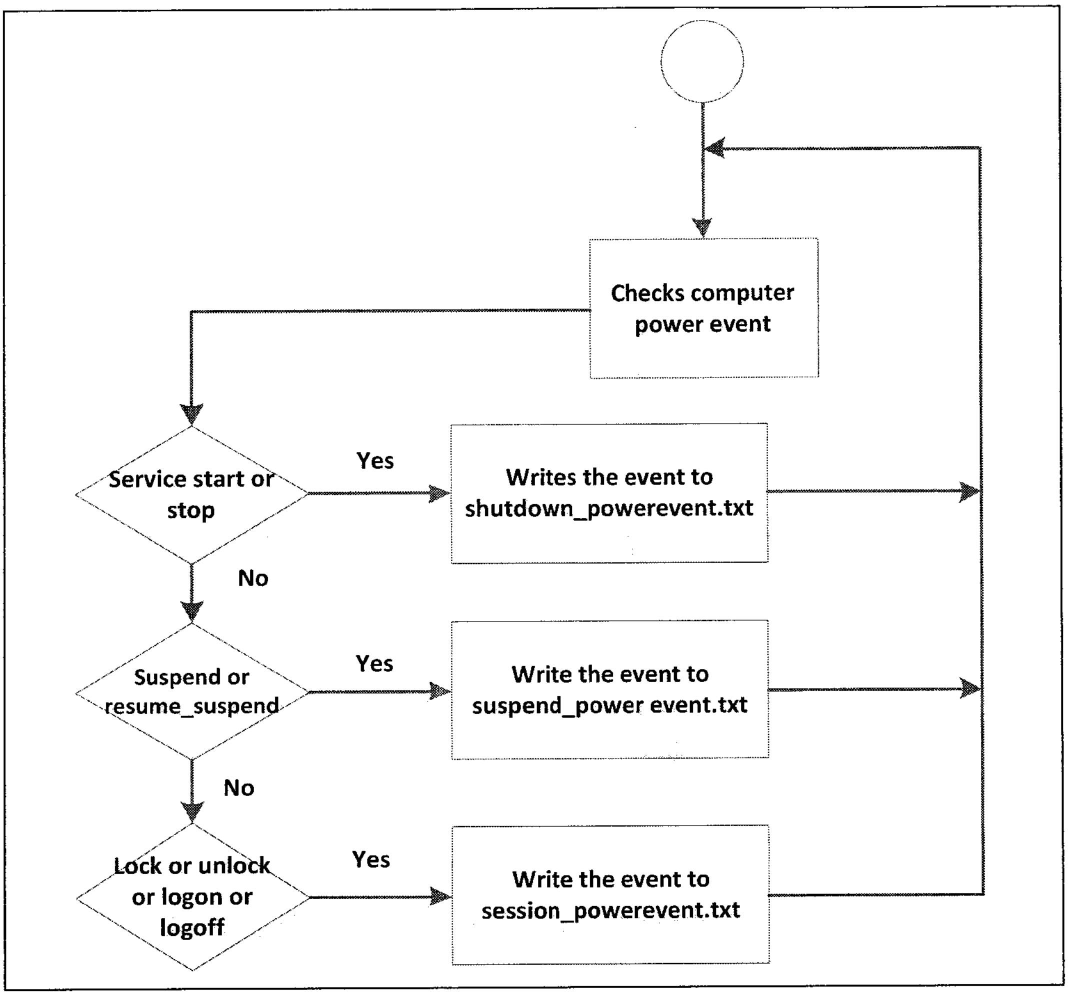


Figure 4.4. The system flowchart of the power transitioning event monitoring component

4.3. Application Idle Time Monitoring Component

Application idle time monitoring component is a part of the background application which monitors the idleness of the foreground application and records the detailed information about the application in the log files. The following section shows the system architecture, pseudocode, and system flowchart of the application idle time monitoring component. They explain how this component works.

4.3.1. Application Idle Time Monitoring Component Architecture

The application idle time monitoring component is a user level application. It has a lower authority as compared to system level application. Each user has a user profile and an application idle time profile as shown in Figure 4.5.

The system architecture of the application idle time monitoring component subscribes to the system idle timer and the foreground application detailed information. It generates a unique application idle time for each application when the computer becomes inactive because there is no user interaction.

Each user profile is attached to an application idle time profile with three log files. The three log files are "penaltyIdleLog.txt", "defaultIdleLog.txt", and "appIdlelog.txt". The idle information about each application is written in these log files.

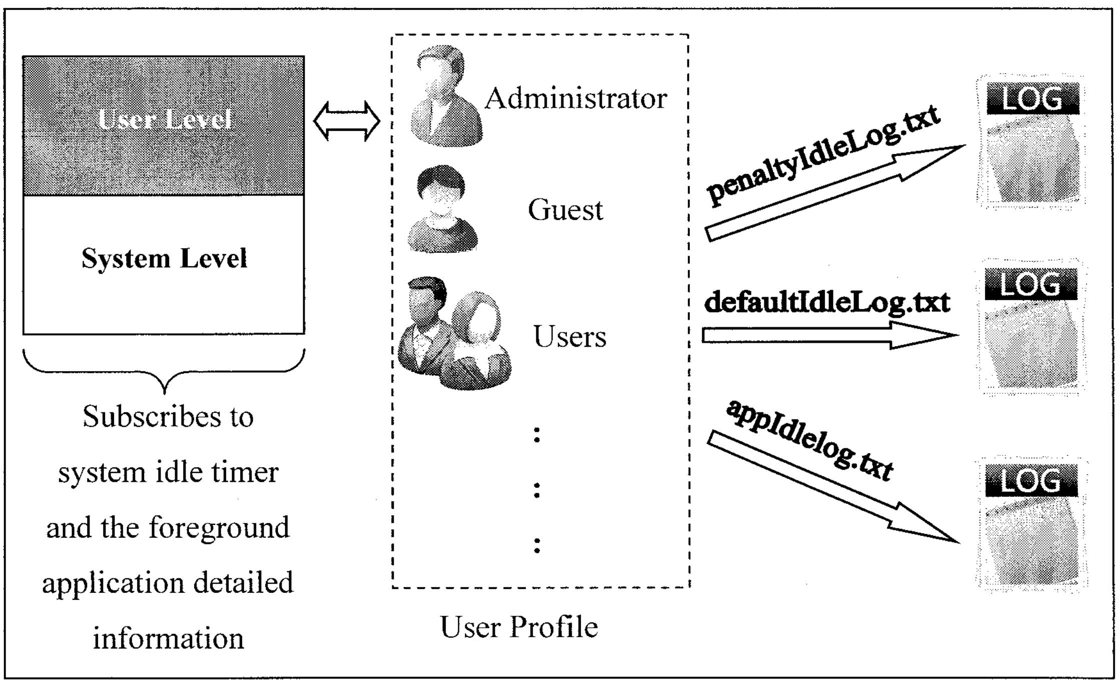


Figure 4.5. The system architecture of the application idle time monitoring component.

4.3.2. Pseudocode and System Flowchart of Application Idle Time Monitoring Component

Figure 4.6 and Figure 4.7 show respectively the pseudocode and the system flowchart of the application idle time monitoring component. These figures show the functionality of it and how the component handles the idle time of the applications that the user has ever used in the computer. There are three main functions are used to create the user profile, and to handle the idle time monitoring and logging process.

The user profile checking function is invoked when the component starts to check the availability of the user profile. A new application idle time profile is created if the current profile is not available in the user profile list. To monitor active or foreground application, an idle application monitoring function is invoked to get the detailed information of the active or foreground application. Then, a comparison between the application idle time and threshold of 20 seconds is used with the usage of nested if-else clause to identify which log files to be written into.

```
Start
1.0 Checks user profile
2.0 If (user_not_available_in_the_list)
    2.1. Creates a new user application idle time profile
3.0 Get active/foreground application detailed information
4.0 Compare active/foreground application idle time with the threshold of 20
    seconds
    4.1. If (idle_time>= threshold)
        4.1.1.Writes to defaultIdleLog.txt
        4.1.2.If (No user_interaction for 10 Seconds)
            4.1.2.1. Writes to appIdlelog.txt
        4.1.3.Else
            4.1.3.1. Writes to penaltyIdleLog.txt
5.0 Go to 3.0
End
```

Figure 4.6. The pseudocode of the application idle time monitoring component.

More importantly, this component has been used in display power management based on application idle time profiling to monitor the idleness of each application. This can ensure that a dynamic timeout for the application can be generated to control the power state of the display. The proposed power management application controls three power states of the display screen of computers in the experimental lab which are active state, screen dimming state, and screen standby state.

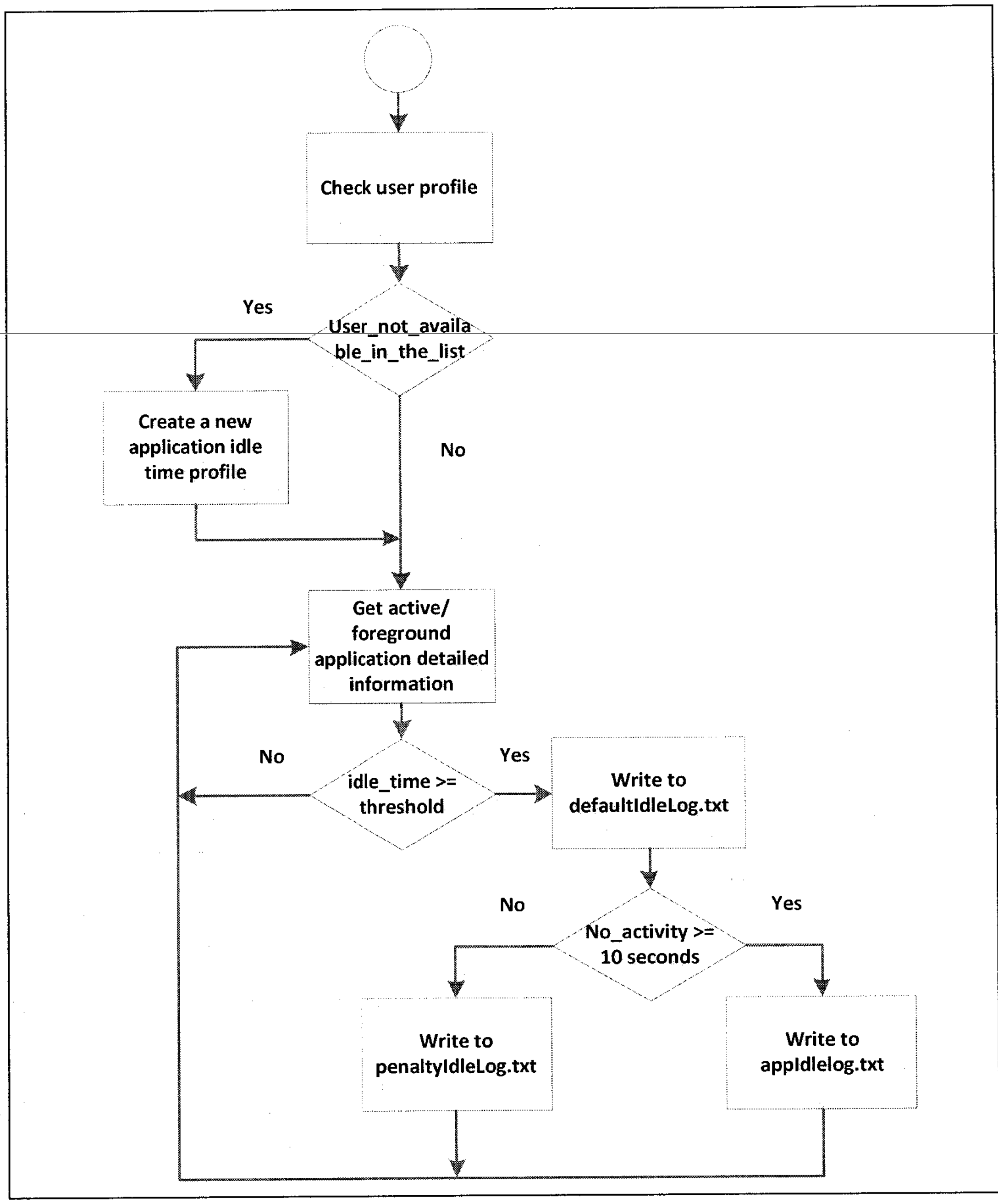


Figure 4.7. The system flowchart of application idle time monitoring component.

4.4. Display Power Management based on Application Idle Time Profiling

A display power management based on application idle time profiling is implemented. It utilizes the application idle time monitoring component as one of the core elements to monitor the idleness of the application that has been used by the user.

4.4.1. DPMaitp System Architecture

DPMaitp has two main components: application idle time profiling component and action enforcer component. All the operations performed by the active applications are controlled by these two components. The application idle time profiling component generates log files that keep the history of the application idle times. This history is utilized to create a dynamic timeout for each application to control the three power states of the display screen. There are three power states which are active state, screen dimming state, and screen standby state.

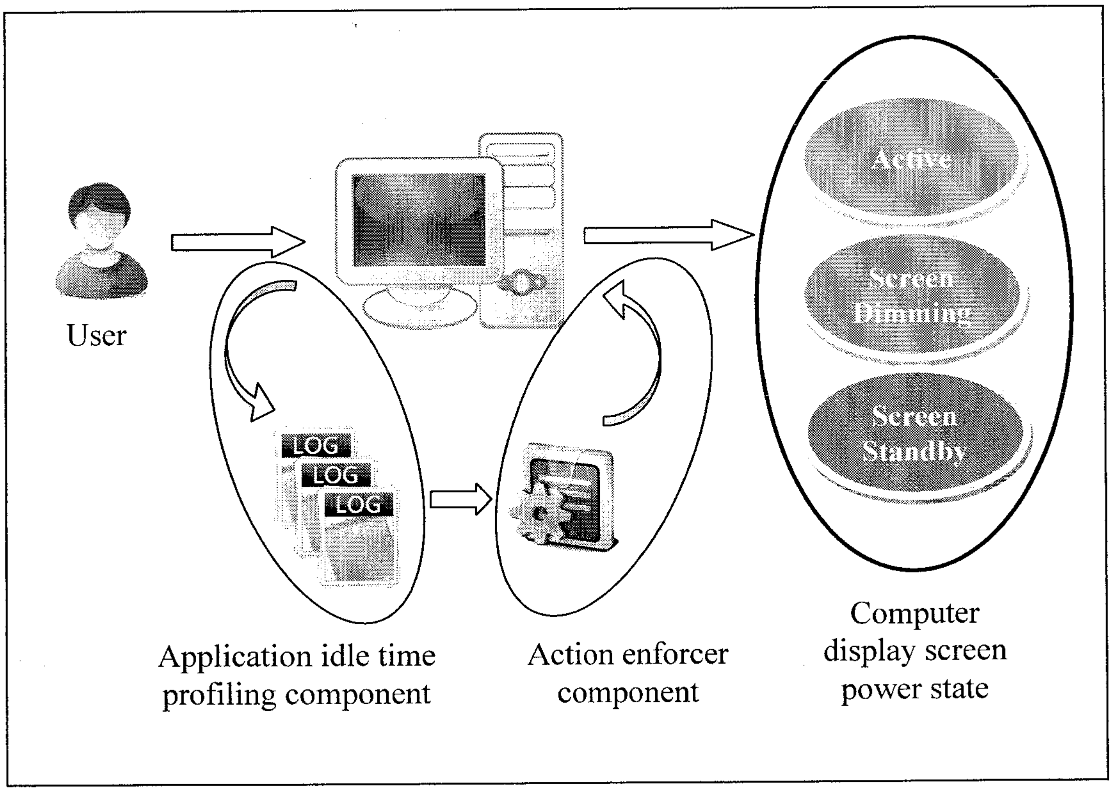


Figure 4.8. The system architecture of display power management based on application idle time profiling.

Application idle time profiling component monitors the idle foreground application and records the activity to the log file. It extends the functionality of the application idle time monitoring component to allow the application idle time profiling component to profile the history of application idle time and generates a unique timeout for each application.

Action enforcer is a main component which checks the foreground application and locates the timeout in the timeout setting file when the computers are idle for a period of time exceeding the default threshold of 20 seconds. It compares the system idle time with the timeout in the setting file for the current foreground application. The screen dimming state will be triggered once the system idle time is equal or bigger than the timeout. A ten-second dimming interval is used to alert the user. Without any user interaction, the display will be switched to a low power state, i.e. the screen off or standby state. Otherwise, a penalty policy will be imposed by increasing the timeout duration.

The penalty policy will select between the maximum value of the maximum value of application idle time duration in application idle time log and two times the last timeout duration in the setting file which has a bigger value. Therefore, the new timeout replaces the old timeout in the setting file.

4.4.2. Pseudocode and System Flowchart of DPMaitp

Figure 4.9 and Figure 4.10 shows respectively the pseudocode and system flow chart of DPMaitp. They depict how the DPMaitp works and the activity flow of the system. There are five main functions used to implement DPMaitp: the application idle time monitoring function, application idle time analyzing function, timeout setting generating function, getting foreground application detailed information function, and power state selection function.

The application idle time monitoring function is invoked when the application idle time profiling component starts. It handles the monitoring process and generates log files. Then, a loop to check the idle time history for each application is executed. This loop analyzes through all elements and generates a timeout for each application listed in the log files using application idle time analyzing function together with the timeout setting generation function. Foreground application detailed information is retrieved through the getting foreground application detailed information function. A comparison between the foreground application idle time and its timeout is performed using a nested if-else clause in the power state selection function.

```
Start
1.0 Start application idle time profiling component
2.0 For each application in idle time profiling log
    2.1. Analyze application idle time
    2.2. Generate timeout/threshold setting for the application and store in
         the setting file
3.0 Get active/foreground application detailed information
4.0 Compare active/foreground application idle time with its
    timeout/threshold listed in setting file
    4.1. If (idle_time >= timeout)
         4.1.1.Dim the display
         4.1.2.If (No user_interaction for 10 Seconds)
                  4.1.2.1.1. Turn off the screen
         4.1.3.Else
                  4.1.3.1.1. Set user interaction penalty
                  4.1.3.1.2. Reset application idle time setting by
                             increasing the timeout
                             (Timeout = Max (2* timeout, MaxIdleTime))
5.0 Go to 3.0
End
```

Figure 4.9. The pseudocode of the display power management based on application idle time profiling.

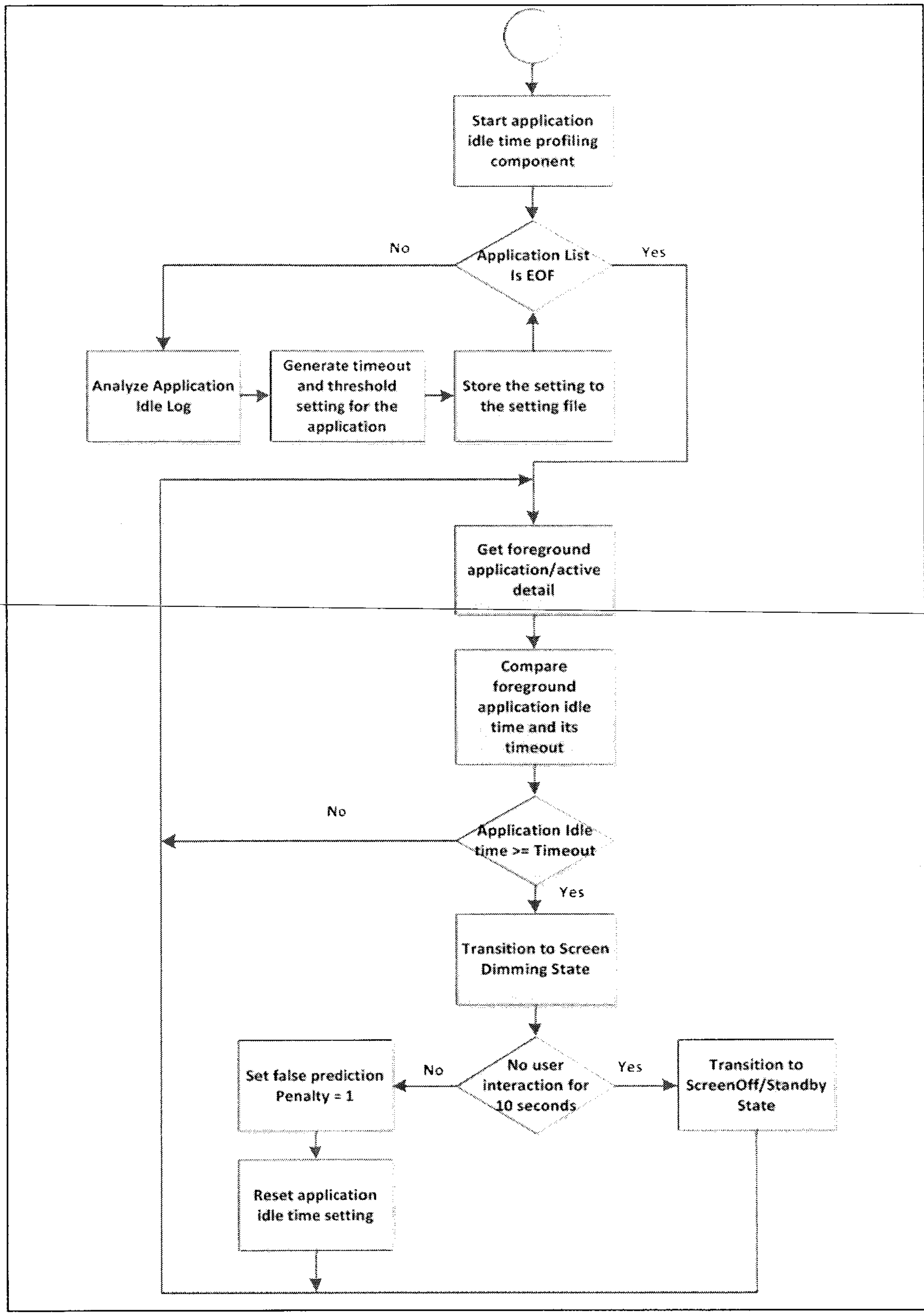


Figure 4.10. The system flowchart of the display power management based on application idle time profiling.

4.5. Data Extracting Tool

Some data from the log files generated by the power transitioning event monitoring component and the application idle time monitoring component are extracted using a data extracting. This tool handles the raw data in the log file and transforms to a useful information such as how long the computer stay in each power state, and the average idle time of each application in the log file.

4.5.1. Data Extracting Tool System Architecture

The extracted information depends on which log file the user selected as shown in Figure 4.11. For data extracted in log files generated by the power transitioning event monitoring component, this tool produces the information related to the utilization of the computer and the duration of computer staying in each power state. Furthermore, the information related to the idleness of each application is produced for the log files generated by the application idle time monitoring component.

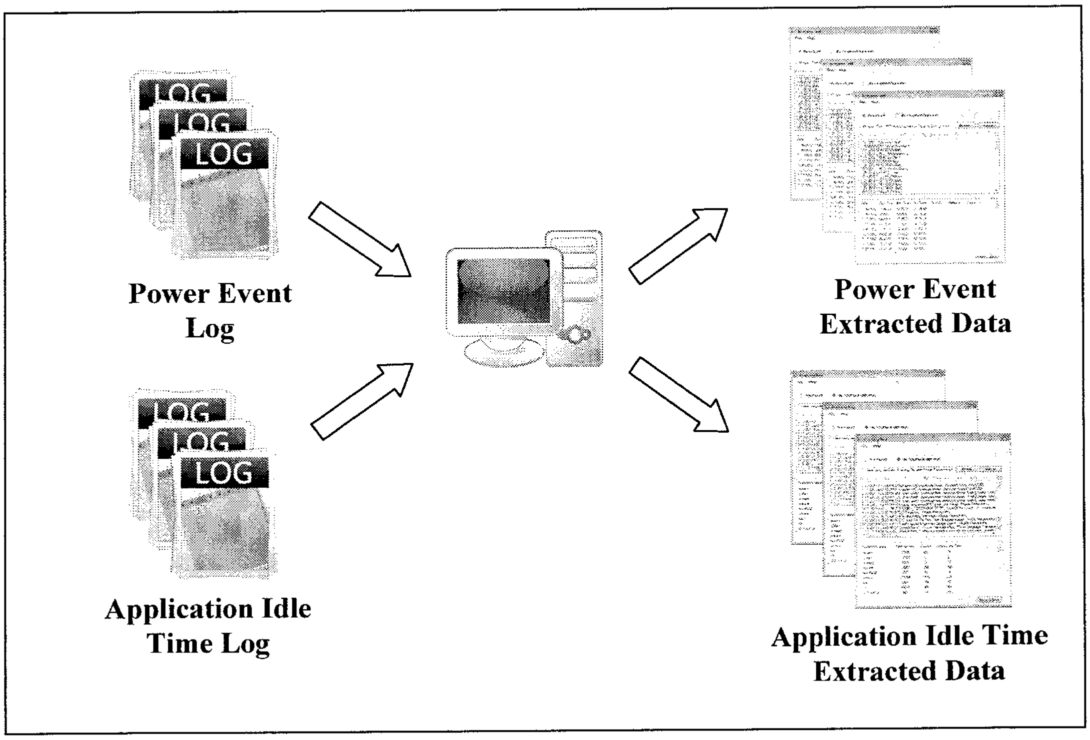


Figure 4.11. The system architecture of data extracting tool.

4.5.2. System Flowchart of Data Extracting Tool

The system flowchart of the data extracting tool is shown in Figure 4.12. It depicts the flow of the system and how the data extracting tool works.

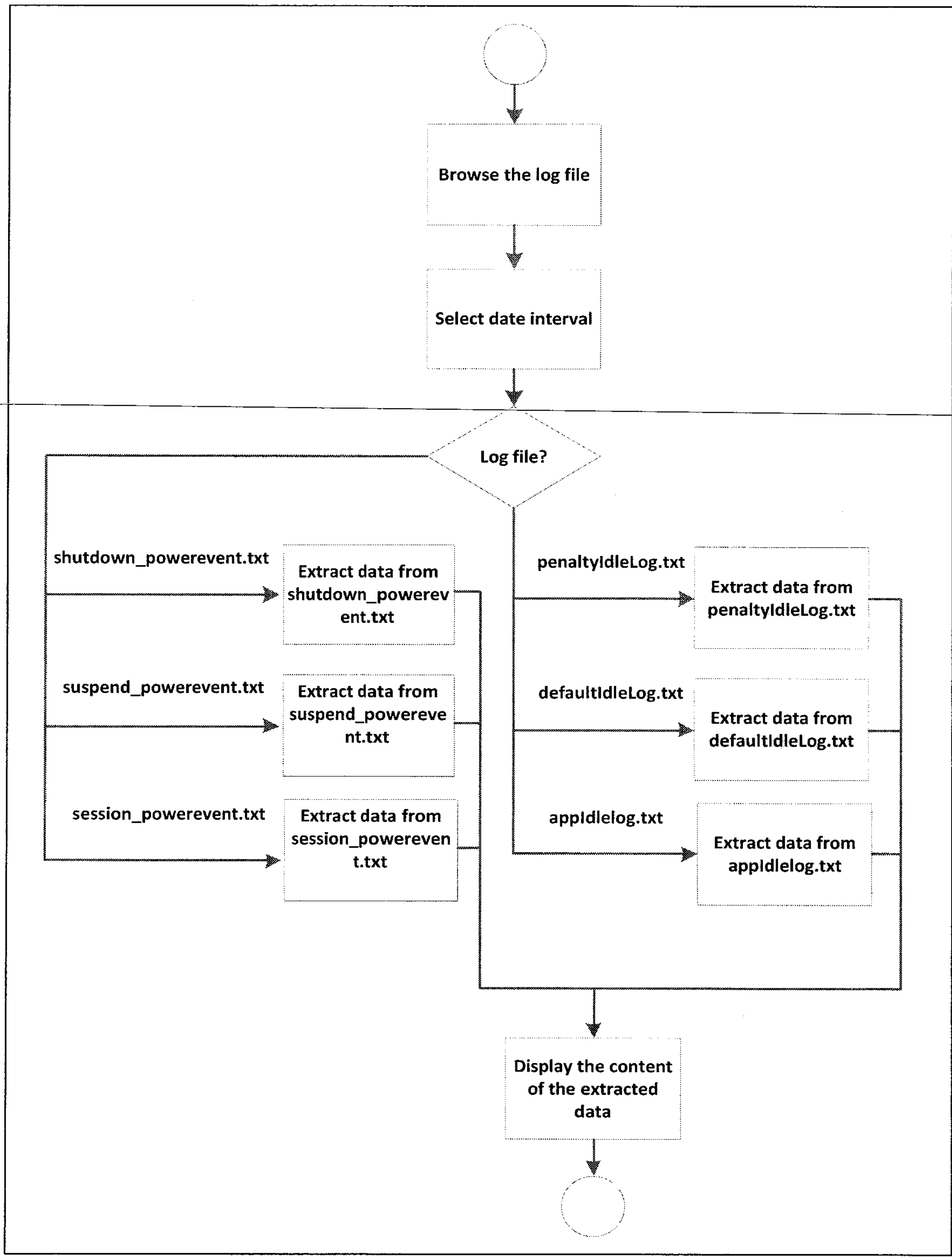


Figure 4.12. The system flowchart of the data extraction tool.

4.6. Power Management based on Application Idle Time and Power Transitioning Event Profiling (PMaitptep)

The power management based on application idle time and power transitioning event profiling is an application implemented for the purpose of evaluating the proposed components. Therefore, it utilizes the power transitioning event and idle time profile to control the power state of the computer and its display screen. The architecture, pseudocode, and system flowchart of the PMaitptep are presented in the following section.

4.6.1. PMaitptep System Architecture

PMaitptep is the updated version of DPMaitp. It provides additional functionality on top of the power transitioning event profiling component and a new action enforcer as shown in Figure 4.13. A new computer standby power state is introduced and controlled by these new components.

The power transitioning event profiling component handles the power event transitioning event monitoring process and log file generation process. It extends the functionality of the power transitioning monitoring component to allow the profiling of the history of the power event in the log file and generates a unique timeout for the computer to control the computer standby power state.

For the new action enforcer component, it has a lower priority as compared to the action enforcer component called by the application idle time profiling component. It checks the current system power event and current system-on time, and locates the timeout in the timeout setting file when the computer stays on the screen dimming state for at least 30 minutes. It compares the current system-on time with the timeout. A warning panel with ten-seconds warning interval has appeared on the screen once the system-on time is equal or exceeds the timeout. A ten-second warning interval is used to alert user attention. Without any interaction, the display will be switched to computer standby state. Otherwise, a penalty policy will be imposed by increasing the timeout duration.

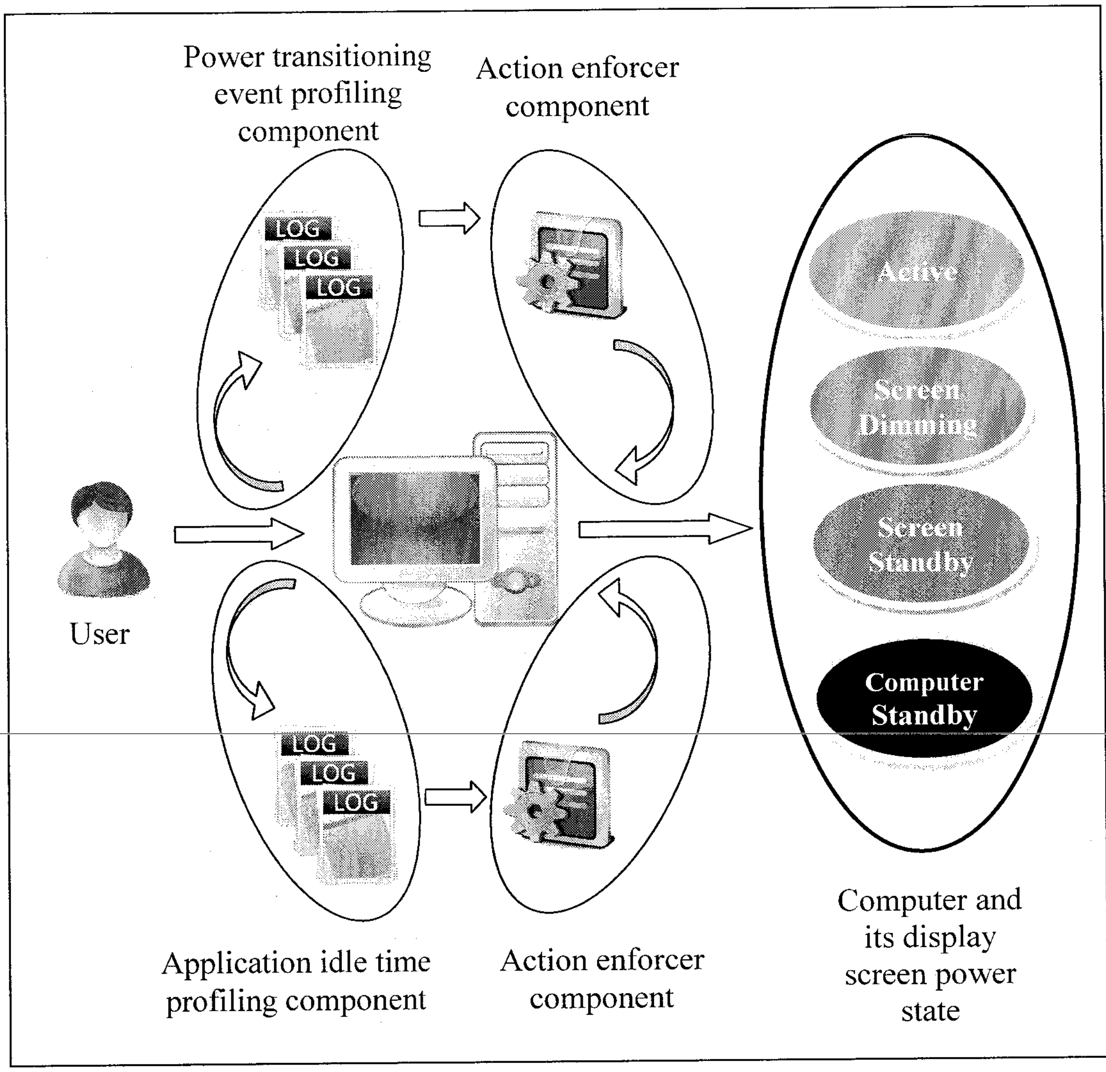


Figure 4.13. The system architecture of power management based on application idle time and power transitioning event profiling.

4.6.2. Pseudocode and System Flowchart of PMaitptep

Figure 4.14 and Figure 4.15 show the pseudocode and system flowchart of PMaitptep is shown in. They depict how the application works and its activity flow of the application. Extended from DPMaitp, PMaitptep enhances the functions utilized by DPMaitp and adds a few new main functions such as power transitioning event monitoring function, power event analyzing function, timeout setting generating function, get-current-power-state function, and power state selection function.

The power transitioning event monitoring function is invoked when the power transitioning event profiling component starts. It handles the monitoring process and generates log files. A timeout to computer standby state is generated by using the power event analyzing function and the timeout setting generating function. Then, the current power state is retrieved through the get-current-power-state function. A nested if-else clause is used in the power state selection function. This function decides whether to switch the computer to standby state.

```
Start
1.0 Start DPMaitp
2.0 Start power transitioning event profiling component
3.0 Analyze system on time in the log files
4.0 Generate timeout/threshold setting for the computer and store in the
    setting file
5.0 Get current power state
6.0 If (screen_standby or system_idle> 30minutes)
    6.1. Get current system on time and timeout/threshold
    6.2. If (system_on_time>= timeout)
        6.2.1.Display warning popup panel for 10 seconds
        6.2.2.If (No user_interaction for 10 seconds)
            6.2.2.1. Switch the computer to standby state
        6.2.3.Else
            6.2.3.1. Set user interaction penalty
7.0 Go to 5.0
End
```

Figure 4.14. The pseudocode of power management based on application idle time and power transitioning event profiling.

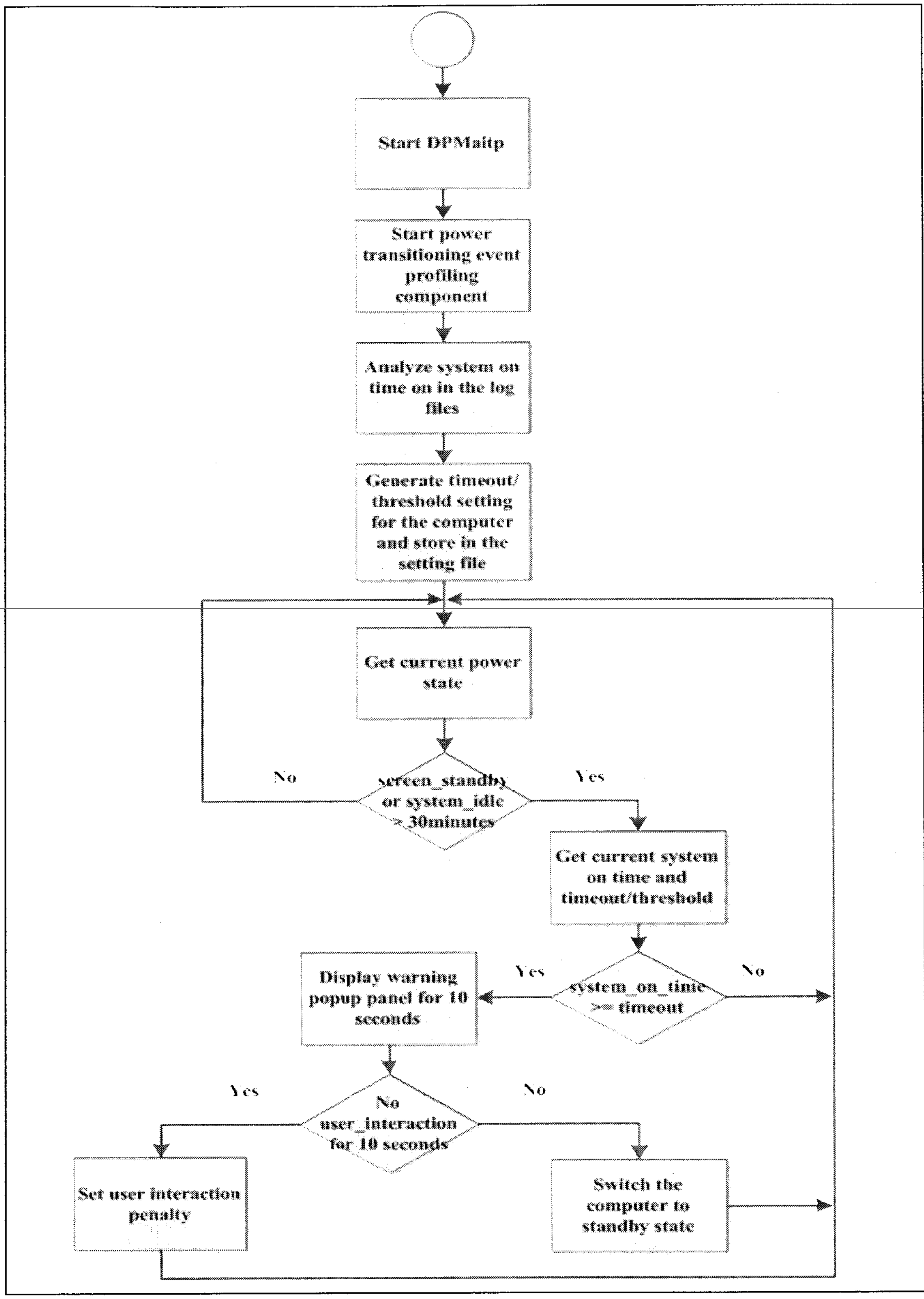


Figure 4.15. The system flowchart of power management based on application idle time and power transitioning event profiling.

4.7. Development Tools

All hardware and software used for developing the components, tool, and applications in this research are listed below:

Hardware

- A laptop (ASUS F5RL) with following specification
  - Intel Pentium Dual Core 1.67GHz
  - 2GB DD2 RAM
  - 80GB Hard Disk

Software

- Visual Studio C# Express Edition 2010

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter highlights the results of the experimental works that have been conducted throughout this research work. The vital results from this phase are analyzed and discussed. The experimental data and the results of those activities are described and examined or analyzed thoroughly.

5.1. Initial Problem Identification

Initial problem identification was performed by measuring the computer power consumption to identify computer power wastage issue in the labs. This marked the beginning point of investigation that leads to the understanding of computer power consumption in the experimental environment.

Computer labs in two the main buildings (block 1 and block 2) in Universiti Teknologi PETRONAS were inspected to identify the power wastage in order to select the lab environment which are suitable for setting up the experiment. Permission was granted by the IT media department to perform power measurement on selected PCs.

5.1.1. Computer Power Measurement

The power measurement of the computers and their display screen in a practical computer lab was taken for a period of time. This lab contains 40 desktop computers and 40 cathode-ray tube display screens with the same model and specification. Therefore, only five power measurements were taken. The mean of these five measurements was taken as the experimental result.

Table 5.1. The power consumed by the desktop computers and its CRT in the experimental lab.

Average Power (watts)		
Power State	**Desktop Computer**	**CRT**
Off	2.9	3.3
Active	75	72
Standby	3	3.3
Screen Dimming	-	58
White-Screen	-	65
Black-Screen	-	52
Note: The desktops and its CRT monitor consume a little bit of power even though they are in Off state.		

The values of the measurement are shown in Table 5.1. The measurement is divided into two types i.e. desktop computer with three power states and CRT display screen with six power states. It is obvious that the consumed power varies from state to state.

On a desktop computer, three power states were measured. They were Off state, Active state, and Standby state. It is noticed that the Off state consumes the least power as compared to other power states. Moreover, the power in Standby state is comparable to that in the Off state. Standby state is used to replace the Off state when the computer needs to preserve its system context. The computers can be switched to standby state if the users forget or neglect to turn off their machine after using it. The standby state allows the computer resumes in its last state without going through a booting process which is a time-consuming process.

On a CRT display screen, six states were measured which are Off state, Active state, Standby state, Screen Dimming state, White-Screen state, and Black-Screen state. The power in the Off state and Standby state consumes the least power as compared to the other power states. Moreover, these two states consume the same amount of power. That is because the CRT display screen does not support the standby state so when the computer sends a request to switch the display screen to standby state, it switches the display screen to the Off state instead.

As the result depicted in Table 5.1, the power in the White-screen and Black-screen state show that the color of the CRT display screen did affect the power consumption of the computer system. The power in Active state consumes the most power because the color of the screen keeps changing which leads to additional power consumption for color computing.

5.1.2. Computer Power Wastage Problem

To determine computer power wastage in the labs, observation and inspection were performed in each computer lab in the two selected buildings. Two scenarios were selected i.e. at-the-end-of-practical session scenario and after-the-operating-hour scenario to explore the usage of each computer lab.

Nine computer labs in each building were explored, four labs on the first floor and five labs on the second floor in each building. Each computer lab has roughly 40 desktop computers. Some of the computers are sometimes removed for maintenance purpose. Each computer lab was observed in a different time slot based on the timetable assigned to each individual lab. Less than 25 percent of the computers have their power management setting configured.

The observation result was collected in the labs in both buildings and is shown in Table 5.2. The result is tabulated with respect to the percentage of computers which were left powered on after usage.

In at-the-end-of practical session scenario, an average of 52.88 percent of computers were left idle in the labs in building one while an average of 56.94 percent were left idle in the labs in building two. On top of that, it is noticed that at least 30 percent of the computers in the labs of both buildings were left powered on. Therefore, those computers sat idly consuming the power wastefully.

In after-the-operating-hour scenario, an average of 29.72 percent of computers were powered on without any user interaction in the labs in building one while an average of 34.44 percent were left idle in the labs in building two. It is also noticed that at least 25 percent of the computers in the labs for both buildings were left unattended. This problem leads to wasteful power consumption in each computer lab in both the buildings.

Table 5.2. Percentage of computers being left on at the end of practical session and after operating hour in the chosen experimental buildings.

Building one			Building two		
Lab	Practical Session Ended (%)	After the Operating Hour (%)	Lab	Practical Session Ended (%)	After the Operating Hour (%)
01-00-03	70	50	02-00-02	65	30
01-00-06	65	32.5	02-00-05	67.5	27.5
01-00-07	75	35	02-00-07	30	37.5
01-00-10	30	25	02-00-10	45	50
01-01-02	32.5	30	02-01-02	52.5	35
01-01-05	37.5	25	02-01-08	40	25
01-01-06	75	67.5	02-01-09	50	42.5
01-01-07	40	40	02-01-10	75	27.5
01-01-10	50	52.5	02-01-13	87.5	35
Average Percentage	**52.88**	**29.72**	**Average Percentage**	**56.94**	**34.44**

The end-of-practical session scenario has a higher average percentage of computers being left powering on after usage as compared to the after-operating-hour scenario. It was due to the scheduling of the lab session and the shutdown policy imposed by the lab technician. Even though the after-operating-hour scenario was handled by the lab technician with the shutdown policy, the percentage of the computers left idle was still high.

The problem with the observation was that, it was time consuming to collect the intended result. Moreover, not all information can be collected, such as, for how long the computers have been used and for how long the computers were left unattended. Therefore, a power transitioning event monitoring component was implemented with monitoring functionality to record all the power events in log files. The log file contains all the necessary information which can be used for analyzing how long the computer stayed in each power state. With this, the period that the computers consume power wastefully can be easily identified by analyzing the computer usage period and the timetable associated with each lab.

5.2. Power Transitioning Event Monitoring Component Usage

With the implementation of the power transitioning event monitoring component, the manual observation method– which was tedious and time-consuming that required one to go through each computer in the labs to collect the result manually, can be replaced with the log file analysis process.

The log files generated by the power transitioning event monitoring component required the lab timetable as parameter in the analyzing process. The log files can be collected using an automatic collecting process by utilizing a centralized shared folder. A batch file was used to copy the log file for each computer.

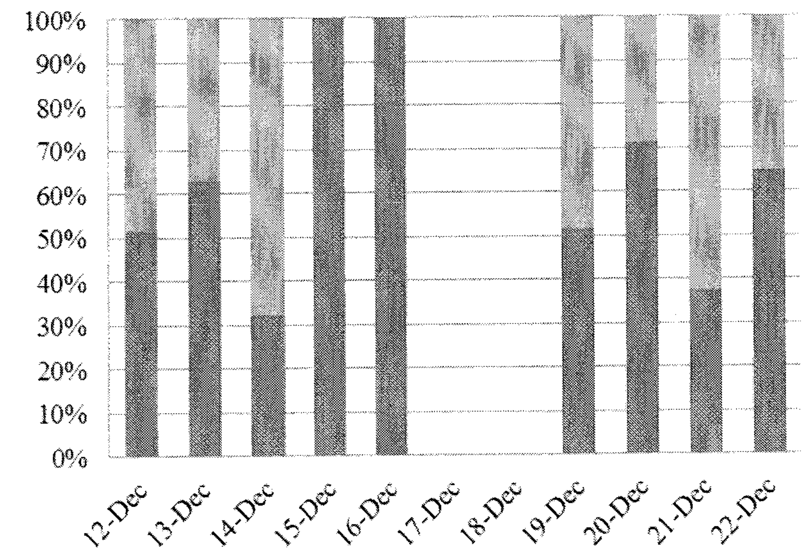
Among all 18 computer labs in the experiment, a lab located on the first floor in building one has been installed with the power transitioning monitoring component. Among all 40 computers in the lab, three computers were under maintenance. The power transitioning event in this lab was monitored for two weeks. Lab usage can be divided into two categories: a lab practical session and a lecture session. Usage time for each section is shown in Table 5.3.

Table 5.3. Lab utilization in the practical computer lab over the weekdays.

	Monday	Tuesday	Wednesday	Thursday	Friday
Lab (Hours)	4	4	2	4	0
Lecture (Hours)	1	2	1	0	2

The results collected from the monitoring service show that the usage of computers during the lecture section has less impact on computer power consumption because less than five computers were used and they were turned off right after the class ended. Thus, the lab practical session was the only main focus of the analysis. The lab practical session experiment gave an interesting result in a selected time frame and session as shown in Figure 5.1 and Figure 5.2.

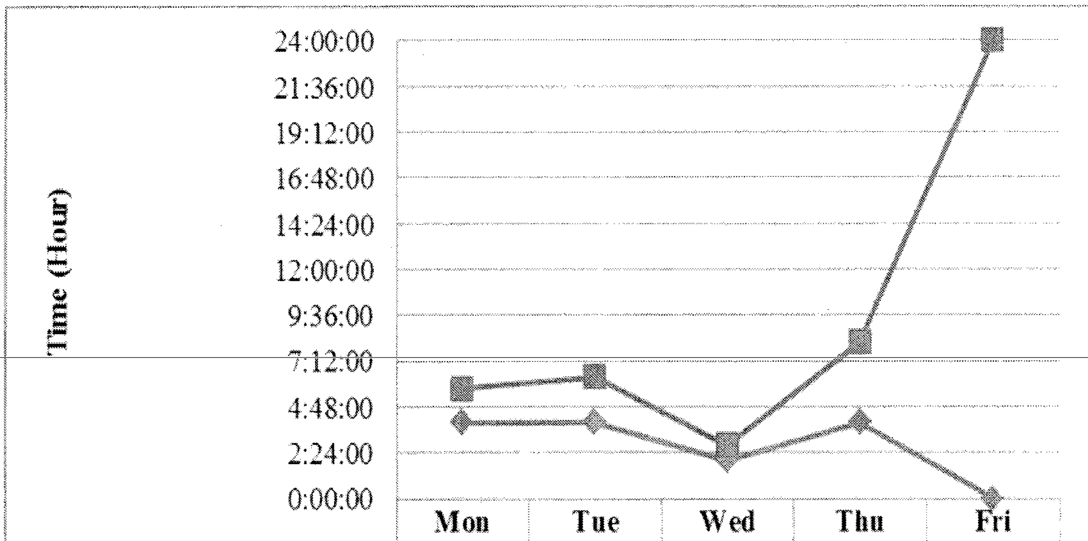
Figure 5.1 shows that each day more than 30 percent of the computers were left powered on until the shutdown batch file was executed by a master computer to turn off all the computers. It was noticed that two computers had been left powered on for two days and five days, respectively. It was later observed that those computers' network cable was unplugged, so the remote shutdown batch file did not take effect. On the contrary, the utilization of the log files generated by the monitoring module gave a more precise result because the actual numbers of computer which have been used or left on can be identified easily as opposed to the manual observation process. When the computer is turned on or off, the event is recorded in the log file and it can be easily retrieved through the data extracting tool. In contrast, the result from manual observation includes only the number of the computer left powering on after each usage but the actual numbers of computers that have been used before the lab session ended cannot be retrieved.



	12-Dec	13-Dec	14-Dec	15-Dec	16-Dec	19-Dec	20-Dec	21-Dec	22-Dec
Powering off after using	16	10	19	0	0	15	10	15	7
Powering on without using	17	17	9	37	37	16	25	9	13

Figure 5.1. The powered on computers and the powered off computers after usage.

Comparing the lab usage time with the average PC powered on time in Figure 5.2, we noticed that the average time that a PC was left powered on is significant in comparison to its actual usage time because most of the users did not turn off their PC after using it. On top of that, PCs left powered on were sitting idle for times ranging from 47 minutes to 24 hours, waiting for the remote shutdown batch file to be executed.



	Mon	Tue	Wed	Thu	Fri
Lab Usage Time	4:00:00	4:00:00	2:00:00	4:00:00	0:00:00
Avg. PC Powering On Time	5:46:05	6:21:17	2:47:01	8:11:45	24:00:00

Figure 5.2. Comparison between the lab usage time and the average computer on time.

5.3. Power Transitioning Event Profiling

A timeout to control the computer standby power state is generated by the power transitioning event profiling component. This timeout is the average of all the computer usage times for each day of the week that have been recorded in the log files.

The timeout depends on the historical data on the usage time in each day of the week. Table 5.4 shows the three weeks' data on the usage time for each day of the week. These data are retrieved from the log file. The timeout associated with each day of the week is generated by averaging all the three weeks usage time value.

A comparison between the current usage time and the generated usage timeout time is performed after each time the computer switched to the screen standby state. If the computer is used for more than 30 minutes and its current usage time exceeds the generated usage timeout time, the computer will be switched to the computer standby state.

Table 5.4. Timeout generated by the power transitioning event profiling component of the three week computer usage period.

Day of the week		Monday	Tuesday	Wednesday	Thursday	Friday
Usage Time	Week 1	2:53:32	7:12:05	7:42:31	5:40:38	2:46:04
	Week 2	4:44:11	7:21:40	8:05:59	7:12:51	7:07:26
	Week 3	8:14:18	8:26:17	7:15:20	7:05:04	8:44:51
Total Usage Time		15:52:01	23:00:02	23:03:50	19:58:33	18:38:21
Average of Usage Time (Timeout)		5:17:20	7:40:00	7:41:16	6:39:31	6:12:47

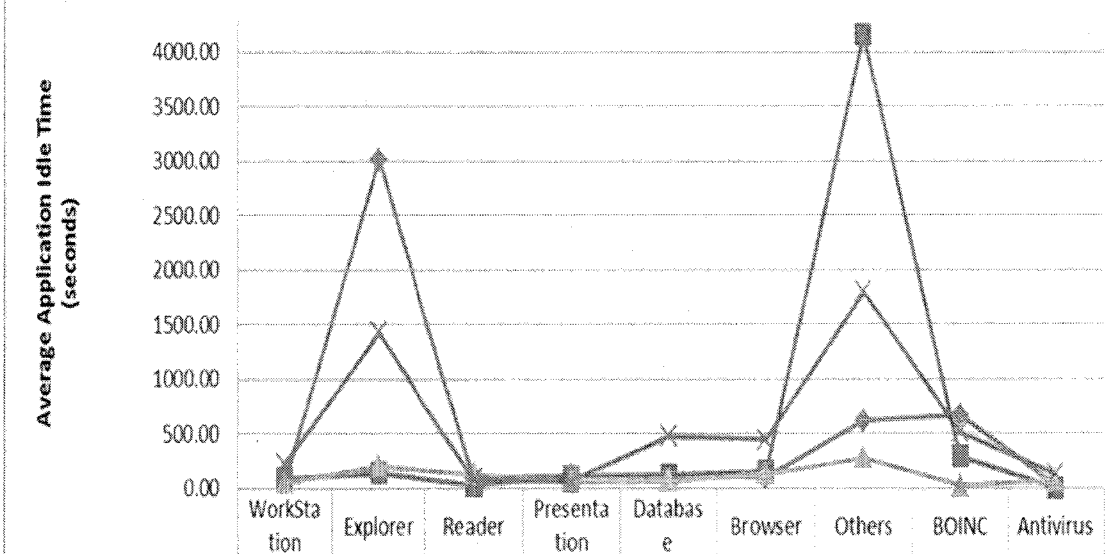
5.4. Application Idle Time Monitoring

The applications frequently used in the experimental lab are categorized into nine categories which are Workstation, Explorer, Reader, Presentation, Database, Browser, Others, The Berkeley Open Infrastructure for Network Computing (BOINC), and Antivirus.

For workstation category, it consists of applications such as Microsoft Word, and Excel. Explorer category contains Windows Explorer, and file operation. Adobe Reader and Acrobat Reader are grouped in reader category. Microsoft PowerPoint is the presentation category. For database category, it contains application such Mysql and SAP. The Browser category includes browser applications such as internet explorer, Firefox, and Google Chrome. For BOINC, it is just an application but it has many background processes and interfaces. Thus, it is considered as a single category. The Esetnod32 antivirus application is installed on all computers in the lab. It is also considered as one of the categories.

The application idle information related to these nine application categories can be retrieved from the log file generated by the application idle time monitoring component. The application idle information for each application in the log file was filtered and grouped into the relevant categories. The average idle time of each group of application can then be calculated.

Only four out of the nine categories i.e. Workstation, Reader, Presentation, and Antivirus gave a promising result because the application idle time data contain a low fluctuated data as compared to the other five categories which are the Explorer, Database, Browser, Others, and BOINC that contains a high fluctuated data as shown in Figure 5.3.



	WorkStation	Explorer	Reader	Presentation	Database	Browser	Others	BOINC	Antivirus
PC40	64.00	3015.50	92.00	116.00	118.64	100.00	614.70	668.50	23.00
PC3	89.00	141.15	25.00	107.00	121.41	150.75	4157.25	287.50	0.00
PC5	56.00	199.38	125.00	52.00	66.89	127.00	274.80	23.00	69.00
All 40 computers	222.78	1423.49	77.38	57.83	476.15	440.42	1797.65	511.98	124.76

Figure 5.3. The average application idle time of nine application categories for three computers and the average idle time of all 40 computers in the lab.

One of the purposes of this categorization process is to generate timeout for the application in each group to control and monitor the computer power state. The timeout is defined as the average of the application idle time in each group. However, this cannot be achieved because all applications need to be grouped in advance before a timeout can be generated. The application idle time monitoring component was implemented in the application idle time profiling component to generate the timeout for individual application rather than for a group of applications. Therefore, the application grouping process can be eliminated from the timeout generation process.

5.5. Application Idle Time Profiling

A timeout to control the power state of the computer system is generated by the application idle time profiling component. This timeout is the result calculated based on the moving average of the application idle time history data in the log file. The profiling component was tested for two moving average methods: cumulative moving average and simple moving average. For simple moving average, three different constant periods have been tested for timeout prediction which is a fifteen constant period (SMA-15), a twenty constant period (SMA-20), and a thirty constant period (SMA-30). The idle time information about Firefox in a computer was used to determine its timeout to control the power state of the computer.

Figure 5.4 shows that there are four peeks with high value which should be remarked in the default idle time line. These are where the application becomes idle or inactive for a long period of time. By using cumulative moving average (CMA), it generates data points which can be used as the threshold to define whether the application was inactive as shown in Figure 5.4.

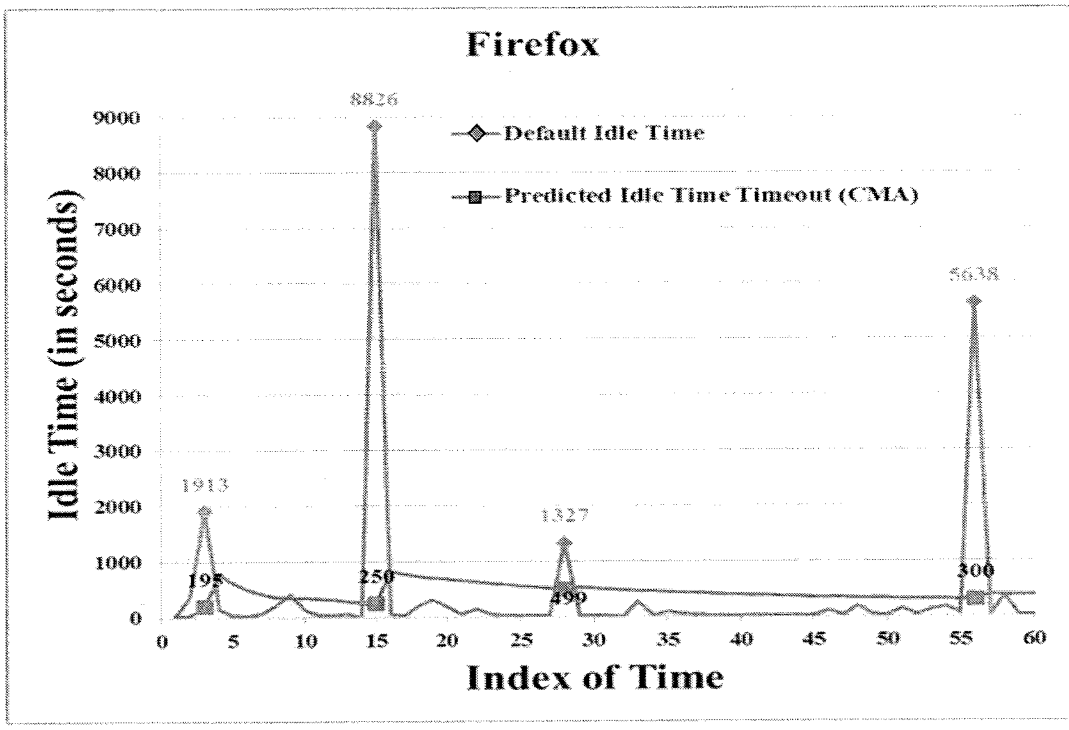


Figure 5.4. Application idle time timeout prediction using the cumulative moving average.

By using SMA-15, a few new unusual idle events were detected in the interval between 30 and 55 as shown in Figure 5.5. These unusual idle events cannot be identified previously by the CMA method. However, the timeout prediction of these unusual idle events in the interval between 45 and 55 might contain few wrong predictions because the idle events happened in between that interval tend to be influenced by new user behavior toward the usage of the Firefox application. Therefore, if shorter timeouts are used in that interval, the users are expected to be interrupted more often which irritate the user.

It is noticed that at point 28 the timeout generated by the SMA-15 method is bigger than the timeout generated by the CMA method. It is because the timeout generated by the SMA-15 receives a high influence of the high value of application idle time at point 15. On the other hand, for the timeout generated by the CMA, the high influence of the value at point 15 is shared among the other low application idle time values before that point of time.

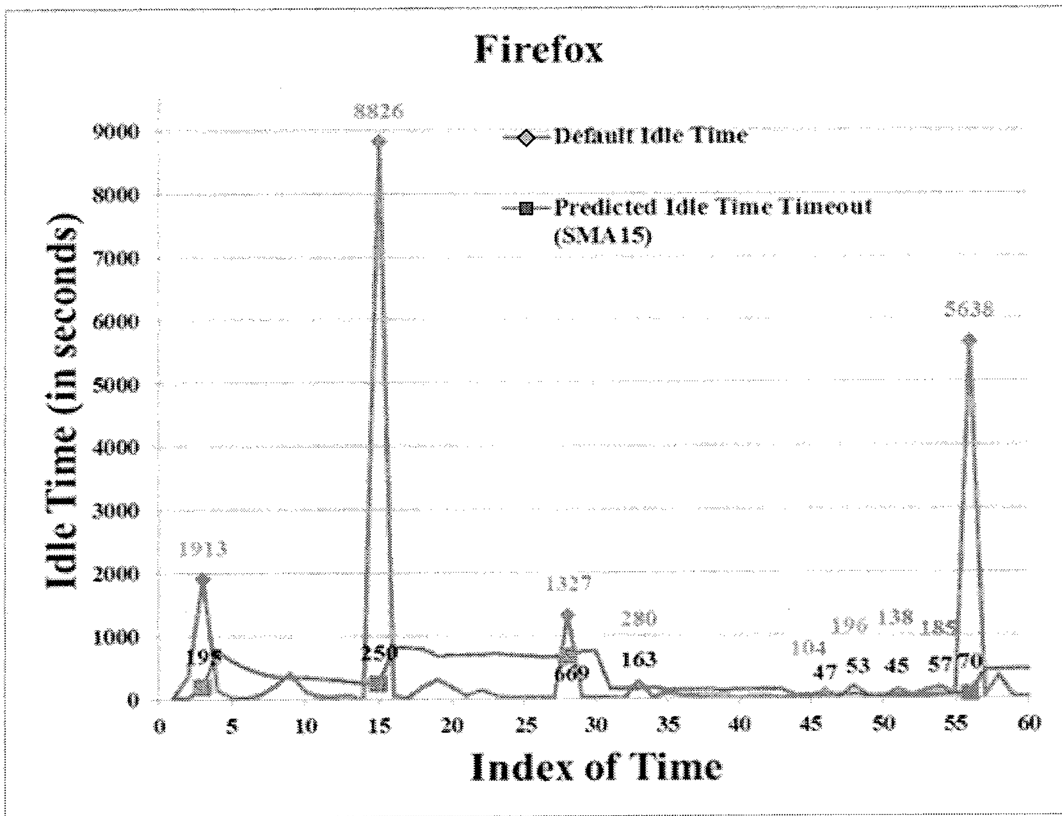


Figure 5.5. Application idle time timeout prediction using the simple moving average with a fifteen constant period.

Figure 5.6 shows how the SMA-20 is used to predict the application idleness with respect to the default application idle time. The SMA-20 gives a slightly different timeout prediction as compared to SMA-15 because of a longer historical data are needed for the prediction. The computer at point 33, which was previously predicted to be inactive using the SMA-15, was predicted to be active because it was influenced by a high idle time value at point 15. Likewise, the idle time value at point 46 was influenced by the value at point 28.

It is noticed that a new idle event was identified in point 58. However, the predicted timeout generated by the SMA-20 for that point is close to the default idle time value which is the point where the user starts a new interaction. Therefore, this prediction might irritate the users as the gap between the low power state and the working state is relatively short.

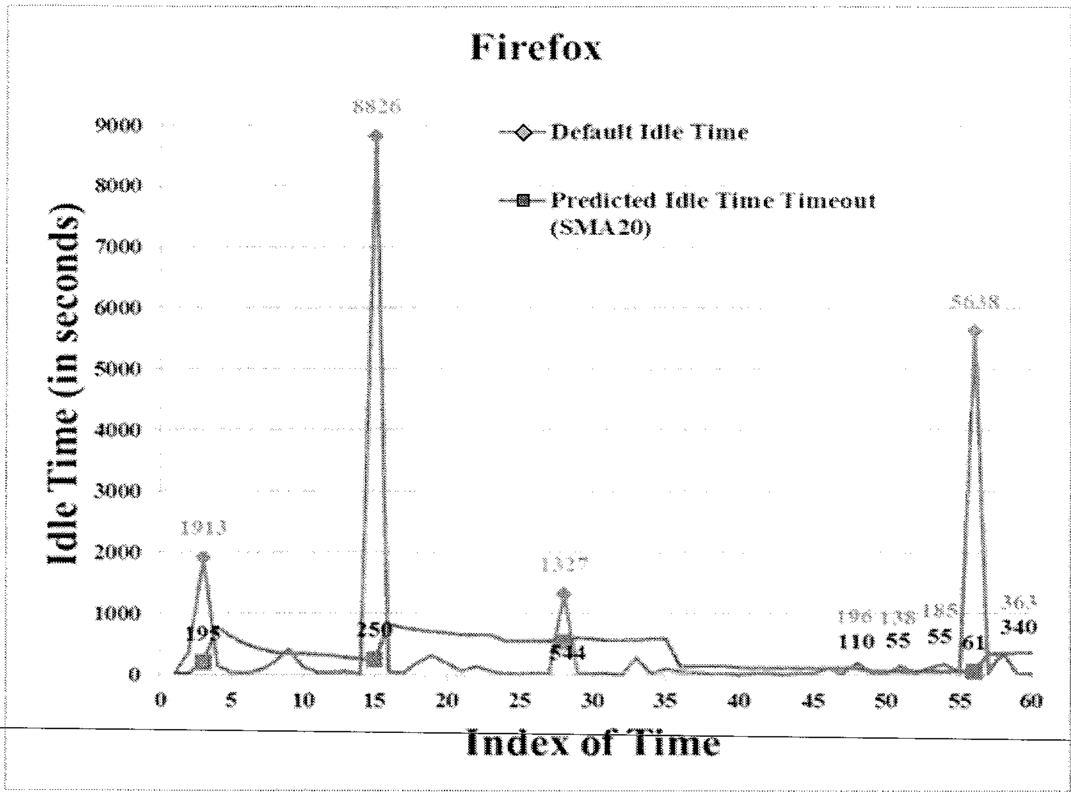


Figure 5.6. Application idle time timeout prediction using the simple moving average with a twenty constant period.

The timeout predicted points generated using SMA-30 were similar to that generated using SMA-20 as shown Figure 5.7. Moreover, SMA-30 gives the same predicted point. The difference is the value of the predicted timeout. Thus, SMA-20 was selected to be implemented in DPMaitp and PMaitptep to generate the timeout for each application for controlling the power state of the computer.

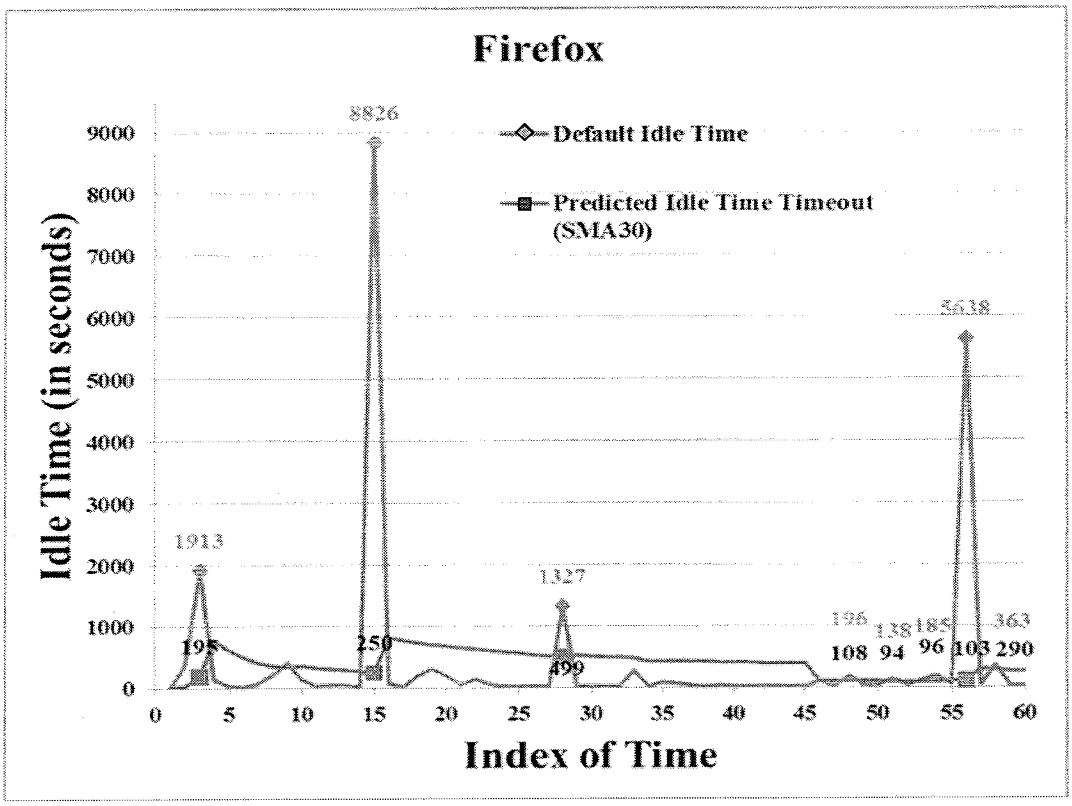


Figure 5.7. Application idle time timeout prediction using the simple moving average with a thirty constant period.

5.6. Evaluation

To evaluate the proposed work, a prototype application called power management based on application idle time and power transitioning event profiling was developed. Some analysis on the time saving of the proposed application, timeout prediction accuracy, and user irritation were performed.

5.6.1. Time Saving Analysis

Time is a critical parameter for calculating how long a computer has been used, how long a computer has been stayed in each power state, how much power has been consumed, and how much power can be saved. Therefore, this parameter was selected to evaluate the performance of the proposed application.

A time saving comparison was made between the built-in power management tool based on a traditional static timeout and the proposed power management application. The comparison was for the computers in the selected lab. Figure 5.8 shows the average percentage time saving of the computers implemented with PMaitptep. It was noticed that the time computers in the lab that stayed idle during active state consuming power wastefully could be saved up to 96 percent. In overall, an average of 74 percent of the idle time was saved by implementing PMaitptep in that particular lab.

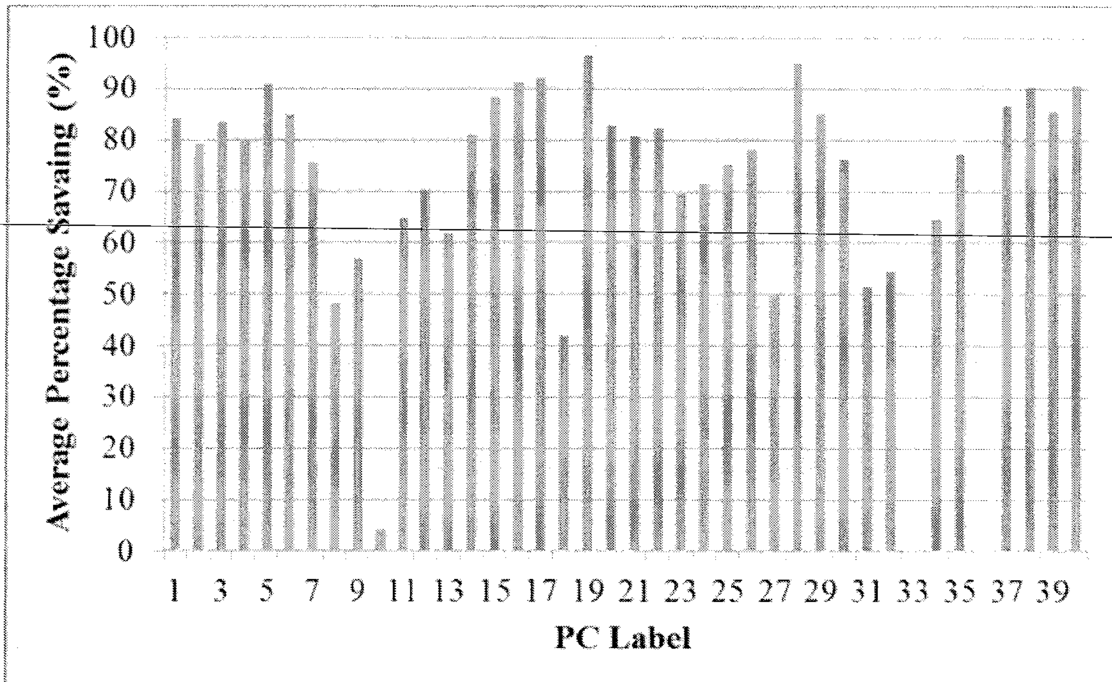


Figure 5.8. The average time saving in percentage of the computers installed with PMaitptep.

5.6.2. Prediction Accuracy Analysis

To understand how well the proposed application predicts the idleness of the computers, an analysis of the SMA-20 method implemented in PMaitptep was performed. For all the computers in the lab installed with PMaitptep and with SMA-20, the idle event can be predicted accurately up to 95.3 percent as shown in Figure

5.9. In overall, it was noticed that an average of 85.8 percent idle events can be predicted correctly with minimal interruption from the user.

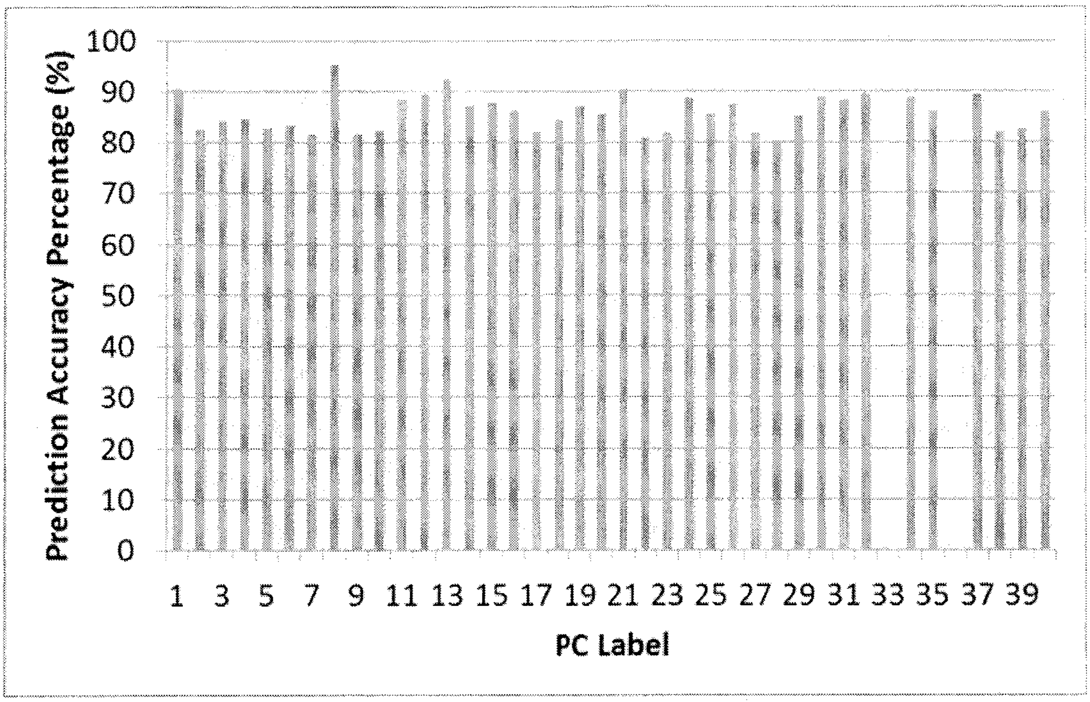


Figure 5.9. The prediction accuracy in percentage of SMA-20 implemented in PMaitptep.

5.6.3. User Irritation Analysis

A user irritation index is used to evaluate the effect of the proposed power management application on users. The user irritation index is the result of the division of the user interruption occurrences and the total power state switching occurrences. It is scaled from low to high ranging from 0 to 1.

The user irritation index for the computers in the selected computer lab is shown in Figure 5.10. It was noticed that the highest index was around 0.2 with an average of overall index of roughly 0.14. It was considered an acceptable range. Furthermore, it was noticed that the user often interrupts the switching process in the early stage (learning stage) which causes the user irritation index to be higher in the early state. Therefore, the user irritation index tends to decrease as more data are recorded in the log file.

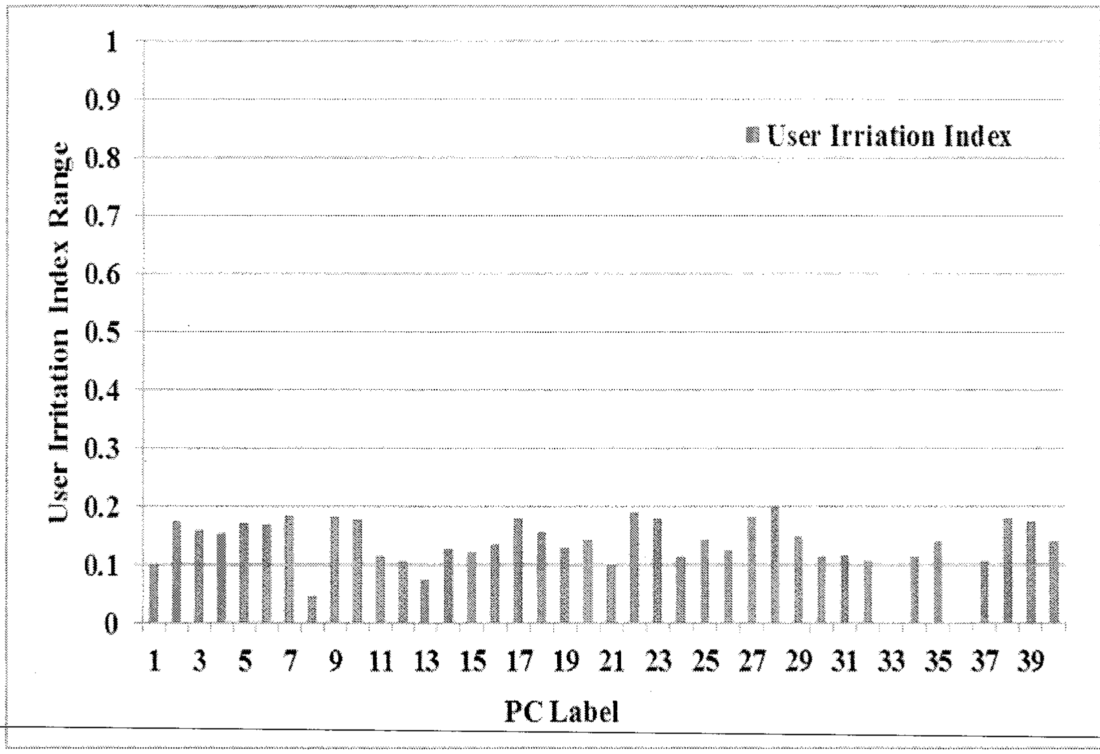


Figure 5.10. The user irritation index of the computer in the selected lab.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter highlights the achievements of this research work and concludes with the contributions of the research work. The objectives of this research were fully completed. This chapter also gives future research direction and future works.

6.1. Achievements

Two components, namely the power transitioning event monitoring component and the application idle time monitoring component are introduced to detect the usage pattern of each computer and therefore able to monitor the user activity on the computer. By utilizing the log generated by the power transitioning event monitoring component, the computer usage time can be identified clearly with a more precise information if compared to a manual observation.

This thesis focused on research in power management which utilizes a novel method in utilizing the application idle time and power event of computers to generate the dynamic timeout to control the computer power states. The generation of the timeout is based on simple prediction methods which are the cumulative average and three methods of simple moving averages. Therefore, the complexity of the prediction is low as compared to other methods with acceptable prediction accuracy.

6.2. Contribution

The contributions of this research are summarized in the following facts:

a) The study of the literature review has shown that there are various kinds of method which are used to implement the power management such as timeout-based method, Sonar-based method, and camera-based method. These methods show a different performance using different detection method and algorithm. This study is the platform for those who are interested in the power management area to extend their knowledge level. Furthermore, the introduction of the proposed work in this thesis offers a new knowledge which is looking at the characteristics and the possible solutions of the power transitioning event and application idle time to be implemented in the power management domain to optimize the computer's power wastage.

b) An analytical analysis of the data retrieved from the log files generated by the power transitioning event monitoring component and the application idle time monitoring component showed that the power event and the application idle time information are proven to be useful. It can be exploited to generate the dynamic timeout to control not only the computer display power state but also the whole computer system.

c) The implementation of the power transitioning event profiling component and the application idle time profiling component which extends the power transitioning event monitoring component and the application idle time profiling component have, respectively, shown a better usage of the proposed components to a working prototype that produces an acceptable result.

6.3. Conclusion

The objectives of this thesis were to explore factors and issues that influence computer power consumption, to monitor user activities and computer usage patterns; thus, to improve effectiveness of the built-in power management tool in the operating system. This is achieved by proposing a power event and an idle time profiling components which extend the built-in power management functionality and to reduce the waiting time in switching the computers to a low power state. These objectives were fully achieved.

Factors and issues that influence computer power consumption were reviewed in Chapter 2. They have a close relationship to the user usage habit, wrong belief, and computer myths. User activities and computer usage pattern were monitored using the proposed components: the application idle time monitoring component and the power transitioning event monitoring component. Moreover, the unusual events such as leaving the computer unattended after the lab session ended and after the operating hour were identified using the log file generated by the proposed component. Their analysis was discussed in Chapter 4. A prototype power management based on application idle time and power transitioning event profiling were implemented to extend the functionality of the built-in power management application. The evaluation in term of time saving, prediction accuracy, and user irritation were performed and presented in Chapter 4. It showed an interesting fact that by implementing the proposed components in the application, the time wasted that a computer usually sits idly consuming power was reduced by an average of 74 percent with the average accuracy of 85.8 percent and the average of irritation index of 0.14. SMA-20 was used for the prediction in the prototype power management application.

6.4. Future Works

Several aspects to the work presented in this dissertation offer potential for the future research. Some of these areas are listed below.

1. Since the potential market tends to switch to the usage of portable smart devices, it is possible and recommended to replicate and extend the work to accommodate not only in computers but also in portable smart devices such as smart phone and tablet. Therefore, all the concepts and methods may be extended to other platforms.

2. Battery life is becoming a main issue and challenge for the development of the contemporary technology which opens an opportunity for the power management research. Up until now, the concept of utilizing the application idle time and the power transitioning event to generate unique dynamic timeout to control the power states of computers and smart devices is still limited. This research has set

a point with acceptable experimental results for researchers who may be interested to perfect the work as currently only two prediction methods have been tested, which are the cumulative average and the simple moving average. Therefore, more accurate methods with better performance can be investigated further.

REFERENCES

[1] "Green Supply Chain News: BP Statistical Review of World Energy 2013." [Online]. Available: http://www.thegreensupplychain.com/news/13-06-19-1.php. [Accessed: 30-Jun-2013].

[2] "BP Statistical Review of World Energy About this review Contents," 2013.

[3] T. Holmes, "Eco-visualization : Combining Art and Technology to Reduce Energy Consumption," *Cognition*, pp. 153–162, 2007.

[4] M. Mark, *It's not easy being green*. 2007.

[5] "Fight Climate Change | Department of Environment Malaysia." [Online]. Available: http://www.doe.gov.my/portal/climate-change/fight-climate-change/. [Accessed: 30-Dec-2012].

[6] "Global Warming & Climate Change (Doha Talks, 2012)." [Online]. Available: http://topics.nytimes.com/top/news/science/topics/globalwarming/index.html. [Accessed: 28-Dec-2012].

[7] L. Chiaraviglio and M. Mellia, "PoliSave: Efficient Power Management of Campus PCs," in *Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on*, 2010, vol. 1, pp. 82–87.

[8] "Policies & Regulations." [Online]. Available: http://www.greentechmalaysia.my/GreenLearn/LearnPoliciesAndActs.aspx#first. [Accessed: 31-Dec-2012].

[9] N. M. Kamarudin Abu Bakar, Mohd Fazli Modh Sam, Md Nor Hayati Tahir, Ismi Rajiani, "Green Technology Compliance In Malaysia For Sustainable Business Development," *Journal of Global ...*, vol. 2, no. 1, pp. 55–65, 2011.

[10] C. Makuhari, "Green Technology in Malaysia," 2009.

[11] "Green Tech Malaysia News." [Online]. Available: http://www.greentechmalaysia.my/GreenLearn/LearnNews.aspx. [Accessed: 27-Jan-2013].

[12] S. C. Chua and T. H. Oh, "Green progress and prospect in Malaysia," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2850–2861, Aug. 2011.

[13] "Triumfant-Green IT Power Management." [Online]. Available: http://www.triumfant.com/IT_power_management.asp. [Accessed: 23-Dec-2011].

[14] C. Baseline, K. W. Roth, and J. Kleinman, "IT and the $8 Billion Electric Bill," no. December, pp. 49–56, 2004.

[15] "Managing Energy Costs in Colleges and Universities," 2010.

[16] "1E-Market leader in IT efficiency | NightWatchman Enterprise for PC power management." [Online]. Available: http://www.1e.com/it-efficiency/software/nightwatchman-enterprise-pc-power-management/. [Accessed: 23-Dec-2011].

[17] "Verdiem." [Online]. Available: http://www.verdiem.com/. [Accessed: 23-Dec-2011].

[18] C. A. Webber, J. A. Roberson, R. E. Brown, C. T. Payne, B. Nordman, and J. G. Koomey, "Field Surveys of Office Equipment Operating Patterns," *Environmental Protection*, no. September, 2001.

[19] J. A. Roberson, C. A. Webber, M. C. Mcwhinney, R. E. Brown, M. J. Pinckard, J. F. Busch, E. Orlando, and L. Berkeley, "After-hours Power Status of Office Equipment and Energy Use of Miscellaneous Plug-Load Equipment," *Environmental Protection*, no. May, 2004.

[20] Energy Star Annual Report, E. Efficiency, G. Our, A. C. Change, and E. S. Partners, "ENERGY STAR ® OVERVIEW OF 2010 ACHIEVEMENTS," *Energy*, pp. 1–4, 2011.

[21] "Energy Efficiency Pyramids." [Online]. Available: http://www.tellerenergy.com/?p=584. [Accessed: 05-Feb-2013].

[22] "Why Energy Efficiency? -- Ipswich Utilities Department." [Online]. Available: http://www.ipswichutilities.org/Print.aspx?Page=ENERGY_WHY. [Accessed: 05-Feb-2013].

[23] J. Mankoff, D. Matthews, S. R. Fussell, and M. Johnson, "Leveraging Social Networks To Motivate Individuals to Reduce their Ecological Footprints," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2007.

[24] J. Mankoff, S. Fussell, and T. Dillahunt, "StepGreen. org: Increasing energy saving behaviors via social networks," *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 106–113, 2010.

[25] M. Rouse and J. Madden, *Desktop virtualization.* TechTarget.

[26] *Microsoft Application Virtualization Technical Overview.* Microsoft.

[27] M. W. Lawrence W., *Green tech: how to plan and implement sustainable IT solutions.* AMACOM Div American Mgmt Assn, 2009, p. 292.

[28] M. W. Lawrence W., L. WEBBER, and M. WALLACE, *Green Tech: how to plan and implement sustainable it solutions.* AMACOM Div American Mgmt Assn, 2009, p. 292.

[29] "Description of the Different Advanced Power Management States." [Online]. Available: http://support.microsoft.com/kb/308535. [Accessed: 23-Dec-2011].

[30] "Advanced Power Management (APM) BIOS Interface Specification V1.2." 1996.

[31] "Advanced Configuration and Power Interface Specification V4.0a," 2010.

[32] J. Kim, J. Choi, and K. Choi, "Design of automatic energy saving-monitor for reducing the waste of PC electricity," *Networked Computing (INC), 2011*, 2011.

[33] T. Bolioli, D. Korn, R. Huang, D. Beavers, T. Novum, and M. Walker, "Power Management of Computers," pp. 128–131, 2004.

[34] K. Liu, "A cost-effective desktop computer power management architecture for the Energy Star Computer program," *Power Electronics Specialists Conference, PESC'94 ...*, pp. 1337–1341, 1994.

[35] B. Nordman, M. A. Piette, K. Kinney, and C. Webber, "User Guide to Power Management for PCs and Monitors," *Management*, no. January, 1997.

[36] "Top 10 Computer Power Management Myths...and Realities: ENERGY STAR." [Online]. Available: http://www.energystar.gov/index.cfm?c=power_mgt.pr_power_mgt_myths. [Accessed: 14-Jan-2013].

[37] "Power Management Frequently Asked Questions: ENERGY STAR." [Online]. Available: http://www.energystar.gov/index.cfm?c=power_mgt.pr_pm_faq. [Accessed: 15-Jan-2013].

[38] "Frequently Asked Questions (FAQs): ENERGY STAR." [Online]. Available: http://www.energystar.gov/index.cfm?c=power_mgt.pr_power_mgt_faq. [Accessed: 15-Jan-2013].

[39] "Power Management Myths Explained." [Online]. Available: http://www.it.northwestern.edu/hardware/eco/myth.html. [Accessed: 14-Jan-2013].

[40] "Power Management Myths." [Online]. Available: http://www.sustainability.jhu.edu/resources/power_management_myths. [Accessed: 14-Jan-2013].

[41] "Debunking PC Power Myths." [Online]. Available: http://pcworld.about.net/od/officehardware/Debunking-PC-Power-Myths.htm. [Accessed: 15-Jan-2013].

[42] "A Brief Hisory of Screen Saver." [Online]. Available: http://www.coolscreens.com/screensaver_brief_history.html. [Accessed: 15-Jan-2013].

[43] The Northeast Recycling Council, "Power Management for Computers and other Office Equipment," 2009.

[44] S. P. Tarzia and R. P. Dick, "Display Power Management Policies in Practice," 2010.

[45] S. P. Tarzia, R. P. Dick, P. a. Dinda, and G. Memik, "Sonar-based measurement of user presence and attention," *Proceedings of the 11th international conference on Ubiquitous computing - Ubicomp '09*, p. 89, 2009.

[46] V. G. Moshnyaga, S. Member, K. Hashimoto, and T. Suetsugu, "A Camera-Driven Power Management of Computer Display," no. c, 2011.

[47] A. Dalton and C. Ellis, "Sensing user intention and context for energy management," in *Proceedings of the 9th conference on Hot Topics in Operating Systems*, 2003.

[48] J. M. Kim, M. Kim, J. Kong, H. B. Jang, and S. W. Chung, "Display Power Management That Detects User Intent," *Management*, pp. 60–66, 2011.

[49] D. Douxchamps and N. Campbell, "Robust real time face tracking for the analysis of human behaviour," pp. 1–10, Jun. 2007.

[50] S. KAWATO, "Scale-Adaptive Face Detection and Tracking in Real Time with SSR Filters and Support Vector Machine," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 12, pp. 2857–2863, Dec. 2005.

[51] S. Yamamoto and V. Moshnyaga, "Algorithm optimizations for low-complexity eye tracking," *Systems, Man and Cybernetics, ...*, no. October, pp. 18–22, 2009.

[52] T. Ohno, N. Mukawa, and S. Kawato, "Just blink your eyes," in *CHI '03 extended abstracts on Human factors in computing systems - CHI '03*, 2003, p. 950.

[53] N. K. Singh, D. S. Tomar, and B. N. Roy, "An Approach to Understand the End User Behavior through Log Analysis," *International Journal of Computer Applications*, vol. 5, no. 11, pp. 27–34, Aug. 2010.

[54] R. Saito and T. Kuboyama, "Understanding User Behavior through Summarization of Window Transition Logs," pp. 162–178, 2011.

[55] Wikipedia, "Moving average." [Online]. Available: http://en.wikipedia.org/wiki/Moving_average. [Accessed: 13-Feb-2013].

[56] P. Robert, "Moving Averages: The Heart of Trend Analysis," *Alchemist*, no. 33, pp. 12–14, 2003.

[57] R. Vadakkoot, M. D. Shah, and S. Shrivastava, "Enhanced Moving Average Computation," *2009 WRI World Congress on Computer Science and Information Engineering*, pp. 541–544, 2009.

[58] K. Charles, D. and D. Julie, R., *Technical Analysis-The Complete Resource For Financial Market Technicians*. 2007.

[59] B. John, C., *Mastering Technical Analysis- Using The Tools of Technical Analysis for Profitable Trading*. 2006.

[60] "ServiceBase.OnPowerEvent Method (System.ServiceProcess)." [Online]. Available: http://msdn.microsoft.com/en-us/library/system.serviceprocess.servicebase.onpowerevent.aspx. [Accessed: 12-Mar-2013].

[61] "PowerBroadcastStatus Enumeration (System.ServiceProcess)." [Online]. Available: http://msdn.microsoft.com/en-us/library/system.serviceprocess.powerbroadcaststatus.aspx. [Accessed: 12-Mar-2013].

[62] "ServiceBase.OnSessionChange Method (System.ServiceProcess)." [Online]. Available: http://msdn.microsoft.com/en-us/library/system.serviceprocess.servicebase.onsessionchange.aspx. [Accessed: 07-Apr-2013].

[63] "Remote Deployment - Automated Silent Remote Software Deployment." [Online]. Available: http://emcosoftware.com/remote-deployment. [Accessed: 15-Apr-2013].

[64] "What is idle time?" [Online]. Available: http://www.computerhope.com/jargon/i/idletime.htm. [Accessed: 02-May-2013].

[65] "GetLastInputInfo function (Windows)." [Online]. Available: http://msdn.microsoft.com/en-us/library/ms646302(VS.85).aspx. [Accessed: 08-Dec-2012].

[66] "Definition of irritation in Oxford Dictionaries (British & World English)." [Online]. Available: http://oxforddictionaries.com/definition/english/irritation. [Accessed: 30-Apr-2013].

[67] "Irritation | Define Irritation at Dictionary.com." [Online]. Available: http://dictionary.reference.com/browse/irritation. [Accessed: 30-Apr-2013].

APPENDIX A

A.1. Shutdown Policies

A.1.1. Shutdown Batch File

The remote shutdown script is written in batch file which utilizes "psshutdown" that allows the administrator to shut down the machine both locally and remotely. "psshutdown" provides more functionality comparing to "shutdown" command-line utilities. In addition to supporting option provided by "shut down" utility, *"shutdown"* can log off the console user or lock the console but it is only supported Windows 2000 or higher for locking. "psshutdown" can be download from http://technet.microsoft.com/en-us/sysinternals/bb897541.aspx *and it* does not require any installation.

There are two ways to use the "psshutdown" command-line utility. The first one is to include full path to the location of the utility so that the script in the batch file can be run properly. The second way is to copy the "psshutdown" to the location of the batch file to be executed. The detail description of the option provided by "psshutdown" utility is shown in Figure i.

The following script is used by lab technicians to shut down the machine after the operating hours: **"psshutdown.exe –f –k –t 0 list_of_computers.txt"**. When this script executed with –f –k flags, the computers in the "list_of_computers.txt" will be forced to shut down even though some applications are still running. For computers which don't support PowerOff, It will automatically reboot. In addition, the computers will be powered off immediately as the –t flag is set to 0 which means the computers has a zero second countdown to shut down.

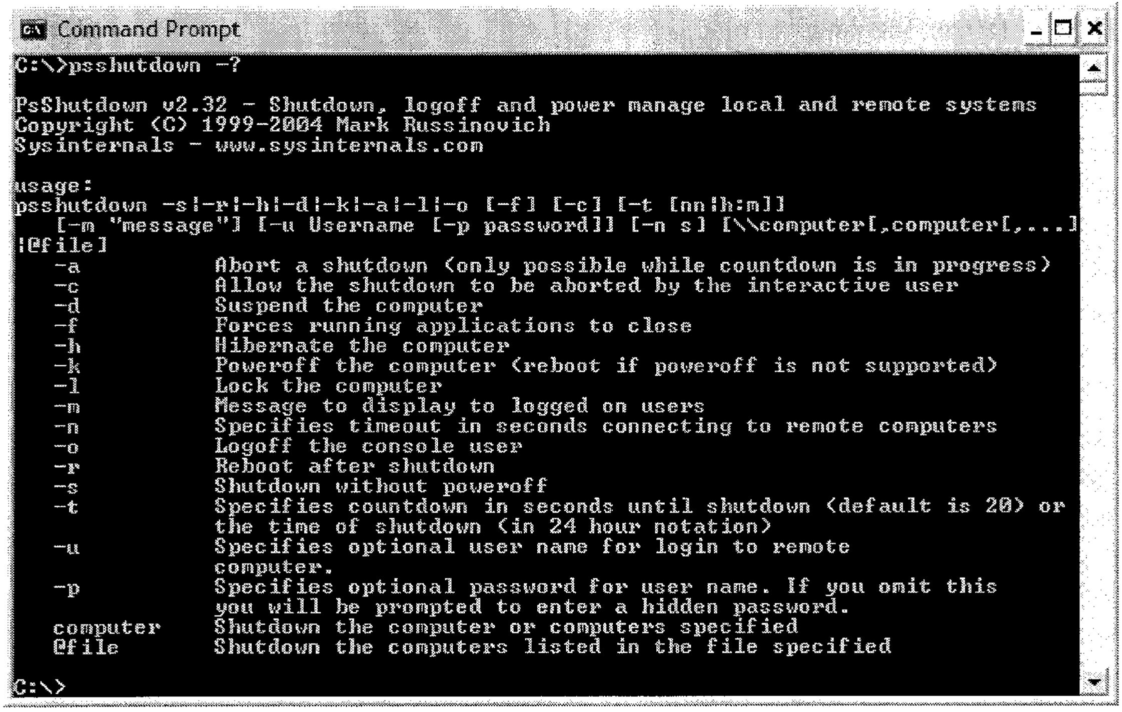


Figure i. Psshutdown command-line utility option.

Shutdown Scheduler

Windows based operating systems allow the users the ability to schedule tasks such as shutdown, reboot, and vice versa to be executed automatically in a repeated interval or at a specific time. This can be done through Windows task scheduler.

To open task scheduler for Windows XP, click **Start**, click **All Programs**, select Accessories, point to **System Tools**, and then click **Scheduled Tasks**.

We will give an example of how we can schedule a shutdown task for computers after the operating hour as following:

- After opening up the task scheduler using the above steps, we will create a new task by double clicking on **Add Scheduled Task**. It will pop up a **Scheduled Task Wizard**.
- Click **Next** to continue.
- Browse for the shutdown execution file from windows system32 folder located in**"%SystemRoot%\System32\shutdown.exe**,click**Open** and then Next.

- Under Perform this task, specify a name for the task and how frequently we want this task to run, and then click **Next**.
- Configure the start time and day for the task, click **Next**.
- Input user username and password, click **Next**.
- Select the **"Open advanced properties for this task"** before clicking **Finish**.
- Click the **Task Tab**. In the **Run** box, specify any additional flags or options for **Shutdown.exe**. Then click **OK**.

To set the shutdown schedule for a later version of Windows, we can refer to an online tutorial (http://www.howtogeek.com/howto/30758/make-your-pc-shut-down-at-night-but-only-when-youre-not-using-it/).

A.2. How to Collect the Log Files

To collect the log files from computers in the lab, a batch file to copy log files to a sharing location and the PSEXEC application are used to run the batch file remotely. A sharing folder location is created in the a administrative computer which accessible to all computers from the experimental lab with full authority. The usage of the script and PSEXEC are described in the following section.

A.2.1. Collecting Log Files using Batch Files

A batch file is used to collect log files from computers in experimental lab. The script with the XCOPY command with "/F /R /Y /E /I" is used. XCOPY is a command to copy files and/or directory trees from one location to another location. This batch file is stored in the proposed application installation path in client computers.

The syntax of **xcopy** is as follows:

xcopy source [destination] [option]

This command is tested in Windows XP. It works fine. However, this command is reported to be deprecated in later versions of Windows-based OS such as Windows Vista and Windows 2008.

The content of the batch file called "copy.bat" that has been used for copying the log file to the shared folder is shown in Figure ii.

```
xcopy "C:\Documents and Settings\guest.%computername%\Application Data\Dynamic
Power Transitioning" "\\05applearn01\log\%computername%\idle\" /F /R /Y /E /I
xcopy "C:\Program Files\DPT\Monitoring Service\log"
"\\05applearn01\log\%computername%\power\" /F /R /Y /E /I
```

Figure ii. The contents of "copy.bat" batch file.

A.2.2. PSEXEC Application

The **PSEXEC** application is used for executing the batch file to copy the log file in each computer to the shared folder remotely. This application can be executed through command line and batch file.

The content of the batch file called "run_psexec.bat" that has been used for copying and executing the "copy.bat" file using XCOPY and PSEXEC command, respectively, is shown in Figure iii.

```
xcopy "\\05applearn01\InstallerDPT\copy.bat" "C:\Program Files\DPT" /F /R /Y /E /I
psexec.exe \\05applearn38 -u user -p password cmd.exe /c
"\\05applearn01\InstallerDPT\copy.bat"
```

Figure iii. The contents of "run_psexec.bat" batch file.