COMPUTER MONITORING SYSTEM

By

Nur Hariza binti Jamaluddin

Dissertation Submitted in Partial Fulfillment of
the Requirements for the Degree
Bachelor of Technology (Hons)
(Business Information System)

JUNE 2006

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

t
QA
76.73
.B3
N974 1) BASIC (computer program language)
2006 2) Microsoft .NET framework

CERTIFICATION OF APPROVAL

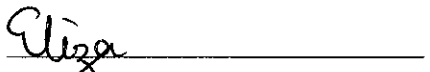
COMPUTER MONITORING SYSTEM

by

Nur Hariza binti Jamaluddin

A final project submitted to the
Business Information System and Information Computer Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Technology (Hons)
(Business Information System)

Approved:



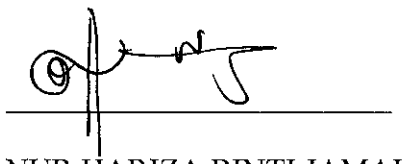
Ms. Eliza Mazmee bt. Mazlan
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



NUR HARIZA BINTI JAMALUDDIN

ABSTRACT

Monitoring of CPU consumption is a very basic requirement in many areas of software. It is especially valuable in the frame of Internet applications, in support of specific aspects such as security, reliability, and adaptability. This report was prepared for final year project emphasizing on computer monitoring system. The computer monitoring system project aims to enable the computer user to retrieve computer status including hardware and software details as well as monitor their computer performance. In order to gather useful information regarding the computer monitoring components and counter, analysis involving the existing computer monitoring system will be conducted. Research on related topics also will be carried out to gain a clear understanding of the system as a whole. At the end of the Part B, the project is expected to come up with a working prototype and complete analysis, design and full documentation for future references. The design phase will covered as aspect of product design starting from the logical design of the software (construction of system flow, data flow and process flow), interfaces design, and planning and physical design of the prototype which includes the program coding and syntax. At the end of the completion of the prototype, it expected to meet the objectives of the project as well as fulfill the FYP undergraduate requirement. As a conclusion, understanding the overall computer's components, counters and attributes to be measured is necessary in determining the efficiency and reliability of a system. Computer monitoring system that provides proper and reliable information and diagnosis will benefit user in terms of time, cost and productivity.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

In the name of Allah the most merciful

First and foremost, I would like to express my utmost gratitude to Miss Eliza Mazmee Mazlan, my supervisor for her immeasurable amount of guidance, advice and assistance through ups and downs of the project's lifetime. Her pertinent and significant help really encourage me to go through this project with full determination to complete it successfully. Not to forget all the lecturers who have helped me with tips and advices throughout the project development.

My heartfelt thanks go to all my friends (especially my housemates) – Siti Sarah Md Jonid, Nurul Amira Ismail, Lubna Shahri, Norhana Yaacob, Nurul Jamilah Abdul Samad, Ayuni Ayatillah and Arfah Md. Nor for their commitment and supports in realizing the goals and the successful completion of the project. Will all their guidance and technical assistance, I managed to come up with this astonishing report and prototype of the system

In addition, my appreciation goes to my mom and dad for their moral support. Even though we are separated by distance and rarely seeing each other during my studies in UTP, their words, loves and understandings give me strength and courage through rough time completing the course.

Special thanks and grateful acknowledgement are also due to those who have proofread the text. I am especially grateful for those who have made afford to ponder valuable advises in the development of the system. The positive encouragement and practical support of these people has been immeasurable importance to me. Thank you.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Operating system statistics provide information on the usage and performance of the main hardware components of the system, as well as the performance of the operating system itself. This information is crucial for detecting potential resource exhaustion, such as CPU cycles and physical memory, and for detecting bad performance of peripherals, such as disk drives.

Operating system statistics are only an indication of how the hardware and operating system are working. Many system performance analysts react to a hardware resource shortage by installing more hardware. This is a reactionary response to a series of symptoms shown in the operating system statistics.

Therefore, here comes the important of having tools that will enable the computer user to frequently check their operating system as well as central processing unit (CPU). The **Computer Monitoring System** will allow you to graphically display information about activity on your system. This includes information about CPU usage, free memory, free blocks, paging, and swapping. There are a series of commands that enable system administrators, programmers, and users to examine each of the resources that a Windows system uses. By manipulating the command and data from the Windows system into an appropriate information which later will be convert into usable output, it could examining these resources thus determining if the system is operating properly or poorly. The system will also enable to provide the user the performance and the critical problem encountered by the system at the moment.

For addition, the Computer Monitoring System also will provide the details information about the computer systems and hardware, features and its capability in supporting various newly functionality and platform such as Windows 64 bits.

1.2 Problem Statement

Since the numbers of computers user around the world increase tremendously due to the shift from the traditional ways; sharing, storing and transferring data, finding information and communications manually, to the usage of computers, it is believed that not all the user know how to monitor their computers by manipulating the command from the system. They did not know the overall performance of the computer. Perhaps the system is taking too long to process jobs or is far behind on the number of jobs that it normally processes. Perhaps the response times for users have deteriorated to the point where users are becoming distracted and unproductive. Even worst, they did not realized that one of their hardware causing serious problem as it does not indicated any significant sign throughout the system. The process of monitoring the computer become slightly important to the PC user as it may utilize as well as maximizing the performance of the computer. Other than that, the details of the hardware in the computer are also important to be identified for the future reference when encountering hardware problem and compatibility.

1.3 Objective and Scope of Study

1.3.1 Objective

The main objective of this project are :

- To ease the computer user to monitor their CPU and computer performance as well as maximizing and pulling the full potential of the computer itself.
- Finding hardware details and computer specifications become much easier without needs to access the BIOS files which is extremely fragile for editing.

- Reduce time consumption, energy and money especially when dealing with sending the computer for services due to the unknown computer problems which is basically just need a small tuning in
- As an indicator to the computer user that their computer is currently not in good condition and need a little bit tuning.

1.3.2 Scope of Study

The scope of the study on this project is basically to understand and explore on how the computer and its operating system synchronizing between themselves and the various hardware. The project will be focused more on the attribute within the computer and operating system that may cause system threshold and overload resource, how this attribute or counter interact with the specific component and get the status value and the data flow between the components, attributes and the operating system. Aside from that, it is also important to gain depth understanding in the status value itself; the type, how it can be converted into usable information and the reliability of the data.

CHAPTER 2
LITERATURE REVIEW AND/OR THEORY

2.1 CPU Monitoring

Resource monitoring i.e., accounting and controlling physical resources like CPU, memory, bandwidth is a prerequisite in many platforms to increase security and reliability, e.g., to avoid denial-of-service attacks. Context-awareness is another benefit that can be gained from a better understanding of resource control. Run time profiling of applications is crucial in server environments in order to protect the host from malicious or badly programmed code.

In this paper we concentrate on the CPU resource. Its monitoring is probably the most challenging of all resource kinds, due to the specificity that we cannot identify explicit CPU consumption places in the code. Indeed, to the opposite of other resources, it is rather considered continuous, i.e., to the opposite of memory that is explicitly always allocated. For that reason, it would be extremely valuable to be able to properly manage CPU consumption, especially if we consider the broad families of Internet applications that exist such as utility computing, web services, grid computing, embedded systems, and mobile object (mobile agent) systems that can be extended and customized by mobile code. Moreover, implementing agent-oriented, context-aware software entities needs the realization of notions of self-organization and self-healing.[3]

2.2 Needs For Computer Monitoring

The system administrator should have a thorough understanding of the activities on the system before users are affected by a crisis. He should know the characteristics of each group of users on the system. This includes the type of work that they submit while they are present during the day, as well as the jobs that are to be processed during the evening. What is the size of the CPU requirement, the I/O requirement, and the memory requirement of the most frequently occurring and/or the most important jobs? What impact do these jobs have on the networks connected to the machine? Also important is the time-sensitivity of the jobs, the classic example being payrolls that must be completed by a given time and date.[1]

These profiles of system activity and user requirements can help the system administrator acquire a holistic understanding of the activity on the system. That knowledge will not only be of assistance if there is a sudden crisis in performance, but also if there is a gradual erosion of it.[1]

This review shows that planning on the resource distribution in CPU processing is important to be monitored to increase and optimized the performance. Therefore, theoretically, before the efficient resource distribution could be conducted, the status of each attributes and components that consume CPU resource should be identified. From here, basically we can identify the area in which usually the CPU resource is being divided to; CPU processes, I/O component, memory and paging. Additionally, operating system statistics provide information on the usage and performance of the main hardware components of the system, as well as the performance of the operating system itself. This information is crucial for detecting potential resource exhaustion, such as CPU cycles and physical memory, and for detecting bad performance of peripherals, such as disk drives.

At it's best, the monitoring tool may provide an early warning mechanism. This ability would warn the user when certain thresholds have been exceeded, such as abnormal and excessive CPU usage, which may indicate a runaway situation. Before this condition brings the entire system to its knees, the user can have he opportunity to not only terminate the ill-behaved culprit, but more importantly, determine the cause. Early warning mechanisms are seen most often in real-time mission-critical systems.[2]

Here, we could see that the diagnosis and CPU status that the monitoring system provides is essential in providing an early warning mechanism to the user. Even though the product itself cannot automatically stabilized the computer or protect it from threshold and corruption, at least the early indicator will help the user from facing an extra cost for purchasing new hardware or computer maintenance upon computer corruption. The user can prevent their hardware from being harmed by inefficient or overload usage.

The main goal of performance testing is to identify how well your application performs in relation to your performance objectives. Some of the other goals of performance testing include the following:

- *Identify bottlenecks and their causes.*
- *Optimize and tune the platform configuration (both the hardware and software) for maximum performance.*
- *Verify the reliability of your application under stress.[4]*

When an application, a hardware system, or a network is not behaving correctly, this is referred to as a *functional problem*. For example, an application or a system with a memory leak has a functional problem. Sometimes functional problems lead to performance problems; for example, when the functions are being achieved, but the speeds of the functions are slow. In these cases, rather than tune the system, it is more important to determine the root cause of the problem and fix it. From this review, we could clearly see the significant of having computer monitoring system. Another example would be when communication is slowed because of networks or name servers that are down.

2.3 Complexity in CPU Monitoring

..monitoring is probably the most challenging of all resource kinds, due to the specificity that we cannot identify explicit CPU consumption places in the code. Indeed, to the opposite of other resources, it is rather considered continuous, i.e., to the opposite of memory that is explicitly always allocated. For that reason, it would be extremely valuable to be able to properly manage CPU consumption, especially if we consider the broad families of Internet applications that exist such as utility computing, web services, grid computing, embedded systems, and mobile object (mobile agent) systems that can be extended and customized by mobile code. Moreover, implementing agent-oriented, context-aware software entities needs the realization of notions of self-organization and self-healing.[5]

From this literature review, monitoring computer performance could be classified as really challenging since the user of the computer system need to keep update with the current computer performance value while the system keep running continuously. It is also crucial to make sure that the interval between each read of the performance monitoring data from the computer system into the monitoring system is suitable so that the effective and reliable data could be triggered. Other than that, ensuring that the monitoring activities runs incessantly is also essential in sequence ensuring the performance could be monitored from time to time.

Additionally, this article portion shows that there is high complexity in ensuring the computer performance through manipulation of various computer system data from different platform as different operating system platform used different naming convention in handling the monitoring attribute data.

2.4 Example of Current Similar System

2.4.1 WhatsUp Premium Gold

The starting point for WhatsUp is network mapping. Once the program is installed, the first important step of configuration begins with discovery of network devices. WhatsUp provides a New Map Wizard to help with the task. The initial phase is automatic: WhatsUp polls the network to identify attached devices. While this is generally accurate, a second phase of editing device information is often necessary. The result is a comprehensive graphical rendition of the network (complete with sub-networks). Another process, called SmartScan, results in showing the hierarchical dependencies in the network (e.g., which servers rely on services from other servers). The complete graphical representation of the network is then used as the means of polling and accessing the operating status for any device on the network. Configuration and maintenance of individual devices is performed through this graphical approach.

Management of WhatsUp monitoring can be performed with either the WhatsUp console, a GUI program typically running on the WhatsUp server, or with Web access through a browser. WhatsUp provides its own Web server to validate administrative access (password or specific IP address) and provides the graphically elaborate network presentation. Web access provides remote monitoring of any device on the network. Network mapping is maintained via scheduled automatic discovery sweeps.

WhatsUp Premium Gold comes with two substantial application monitoring components. Monitoring for Microsoft Exchange servers includes monitoring the real-time status of important services, such as the Mail Transfer Agent and the Routing Engine Service. It also enables threshold monitoring for CPU, disk space, and memory use for Exchange servers. Likewise, Monitoring for Microsoft SQL Server covers system resources, database system errors, and assesses the use of cache data for user queries.[6]

WhatsUp Premium Gold is basically Network device monitoring software for Microsoft Windows, Exchange, and SQL servers produced by Ipswitch for Windows NT 4.0 SP6, Windows 2000, Windows 98, Windows ME, Windows XP, Windows Server 2003 platform. This product can be viewed as a similar product for monitoring purposes, opposed the similar functionality with the computer Monitoring System. The concept and the objectives is the same which is to provide monitoring tools for a better performance and increase in efficiency and effectiveness.

2.4.2 JConsole

The Java 2 Platform, Standard Edition (J2SE) 5.0 release provides comprehensive monitoring and management support. It not only defines the management interfaces for the Java virtual machine, but also provides out-of-the-box remote monitoring and management on the Java platform and of applications that run on it. In addition, JDK 5.0 includes the Java Monitoring and Management Console (JConsole) tool. It uses the extensive instrumentation of the Java virtual machine to provide information on performance and resource consumption of applications running on the Java platform using Java Management Extension (JMX) technology.

The article describes how JConsole can be used to observe information about an application running on the Java platform. The article first gives an overview of the J2SE 5.0 monitoring and management architecture and how JConsole plugs into the architecture. It then describes how to use JConsole to access several core monitoring and management functionalities provided by the Java platform including:

- *Detect Low Memory*
- *Enable or disable GC and class loading verbose tracing*
- *Detect deadlocks*
- *Control the log level of any loggers in an application*
- *Access OS resources—Sun's platform extension*
- *Manage an application's Managed Beans (MBeans)*

In the 5.0 release, the Java virtual machine (JVM) is highly instrumented for monitoring and management. The platform instrumentation provides information on performance, resource consumption, and the JVM and logging settings of applications running on the Java platform.

JMX provides a standard way to instrument the Java runtime environment and applications, and the JMX Remote API allows that instrumentation to be accessed remotely. The instrumentation is accessible through the JMX managed bean (MBean) interfaces, which are registered in the platform MBean server. Applications can also create their own MBeans and register them in the platform MBean server, which can serve as a single point for remote access. A JMX-compliant client, such as JConsole, can connect to the platform MBean server and manage the application (as well as the Java platform) using JMX technology.[7]

2.5 Balancing Performance, Energy and Quality in Pervasive Computing

The CPU monitor predicts availability using the smoothed estimated of recent load. The monitor first determines the amount of competition for the CPU by measuring the percentage of the cycles recently used by other processes. It then calculates the percentage of cycles available for operation execution by assuming that background load will remain unchanged and that the operation will get fair share of the CPU. It multiples this value by processor speed to predict the cycles per second the operation will receive.

The monitor observes CPU usage by associating the operation by the identifier of the executing process. This distinguishes the CPU usage of concurrent operations. Before and after execution, the monitor observes the CPU statistics for the executing processes and its children using Linux's file system. It returns total cycles used by the operation.[10]

This literature review basically explained on how the CPU particularly prioritizes its processing resource in order to allocate the resources to each of the processes executed within the computer. By doing so, computer could be run more efficiently.

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Procedure

3.1.1 Planning

Basically this project will be executed in two phase; first semester (Final Year Project Part 1) and second semester (Final Year Project Part 2).

First Semester

For the first semester, all works will be concentrated on the research, literature review, collecting data and planning. Particularly, it is expected that by the end of the first semester, the author would be able to layout the findings and identify the potential software design and tools to be implemented that will be used for the development phase in the second semester as well as design and implementation plan. The research area that the author going to focus on is particularly on the component (object) and the counter that need to be monitor in the computer by using the data provided by the operating system. Within this phase, it is also crucial to identify the synchronization between required tools to develop the software with the operating system. Aside from that, the author will also do some design and analysis on the user acceptance part in order to ensure upon completion, the objectives and scope of study for this project will be achieved.

Second Semester

As for the second semester, the chosen field will be studied in depth in terms of triggering and indicating the component and counter practically. In addition to that, time will be allocated in developing skills in using HTML and scripting to develop the monitoring system. During this phase, all the theory and findings on developing the system will be put into practical to achieve the completed product by the end of the semester. Modules for each monitored counters and component will be developed stage by stage (or unit by unit) to avoid inefficiency and to provide a stable platform for the author to proceed to another area of computer component that need to be monitored. By the end of the semester, questionnaire will be conducted for the user acceptance analysis purpose.

3.1.2 Analysis

The analysis part is mainly focusing on how the operating system will provide the required data and diagnosis to the software through the VBScript modules and Windows Management Instrumentation platform, the main tools that will be used to construct the product, how it will provide indication to user when symptoms and computer bottleneck is identify and the analysis on how the computer user could increase their computer efficiency through modification of computer component. Since the computer can be operated in various version of operating system (OS) platform, the product will be mainly developed based on the Windows XP structure because each OS requires different script and ways on how the required monitoring data could be triggered from the OS.

This application will be enable to monitor computer performance as well as providing the user with the information about the hardware used, system

software and the other supporting functionality supported by the current computer setting and features.

During the analysis phase, the author have identify the required monitoring attribute that is going to be implemented within the system as well as successfully classified and categorized each of the attributes based on its object classes. The outcome of the analysis could be found in the Result and Discussion chapter later on. Other than that, analysis also has been conducted on how the system will be able to meet the objectives of the project especially in reducing time consumption, money and energy to the computer users.

3.1.3 Design

The design part has been divided into two areas; logical design and physical design. The physical design will only be executed during the second phase of the project whereby it involved the product development, dealing with construction of coding and modules.

Logical Design

Firstly, the logical design phase involved the construction of the system itself whereby the author need to identify how the system is going to behave upon receiving input from the user or from any related source. It is also crucial to identify process involved in order to process the input and produce expected output. Therefore, the author choose use case model as well as process model to construct both design needed in sequence to obtain the necessary requirement desired before the author can proceed with the physical design phase.

Basically, a use case will describe a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse. Use case diagram (UCD) can be used to describe the functionality of a system in a horizontal way. That is, rather than merely representing the details of individual features of the system, UCD can be used to show all of its available

functionality. It is important to note, though, that UCD are fundamentally different from sequence diagrams or flow charts because it does not make any attempt to represent the order or number of times that the systems actions and sub-actions should be executed.

While for the process flow diagram, the author fundamentally used it as a graphical presentation of the sequence of all operations, movements, inspections and storage activities of a process. It describes the sequence and relationship of the tasks that make up an activity.

In order to identify the source of the data needed by the system as well as where the data is going to be sent, the author have decided to use data flow model design to keep track and recognize which portion of computer component attribute is needed. In data flow model, functional transformations will process their input into outputs. Data flows from one to another point (e.g from operating system to the Computer Monitoring system software) and it is transformed as it moves through the sequence. Each processing steps is implemented as a transform. Input data flows through these transform until converted into outputs.

Physical Design

The physical design phase involved the construction of the prototype which includes the interface construction, method for the system process and procedure and the function of the system. The author has divided this phase into two whereby the construction of the system interface is conducted during the first part of the project while the remaining tasks for the physical design will be conducted throughout Part B. Minor changes for the system interface was carried out during the semester break last time to suit the programming necessity.

During Part A, the construction of prototype interface was made starting with the design of the system storyboard so that the author could obtain brief view about the system. Followed by the creating the prototype interface, plus a little

portion of system functional coding as a testing base before I proceed with overall system construction as the function for each attribute involved are mostly the same.

Later on, during Part B, the remaining prototype construction will be completed as well as any minor changes that may occur during the process.

3.2 Tools

The following are the tools used in the development of the project :

3.2.1 Softwares

• Macromedia Dreamweaver 2000	Cascade Style Sheet Development HTML Application Development
• Microsoft Visual .NET	Scripting using VBScript
• Adobe Photoshop	Visual and Image Editing
• JavaEditor	HTML Scripting
• WMI Editor	Retrieving Data

3.2.2 System Requirement

- Desktop or Laptop with Windows Operating System installed
- 256 MB of RAM (at least)

CHAPTER 4
RESULT AND DISCUSSION

4.1 Result of the Analysis

The computer monitoring system concerned with the reliable scrutinized over several attributes in computer operating system which had been identified and selected earlier. These include the hardware attribute, operating system attribute, software and the process.

The monitoring attributes include :

Attribute	Purpose
CPU Usage	To identify and resolve performance bottlenecks. This can be invaluable when trying to understand why and when your system runs slow and the ability to identify what processes cause the degradation.
Memory Utilization	How much memory is currently in use, so one can gain insight into system tuning? The more components that are simultaneously active, the more memory is demanded until excess swapping occurs.
Swap Utilization	To identify excess and frequent swap activity when memory is over committed. This can be a good identifier for needing more memory. When physical memory is exceeded, based on the programs simultaneously running and their memory requirements, the operating system responds by temporarily "moving" pieces (4KB pages) of less active memory out to disk. These chunks of memory

	are brought back into memory when the memory is referenced. A Swap monitor can provide insight into excess swapping.
Program Performance	Enable the user to assess the impact of running program(s). Today's diverse software and multimedia craze requires the high performance of at least an i486 based or equivalent processor to keep up with CPU and memory intensive applications. The ability to examine the CPU usage of a program can help identify demanding programs and perhaps ill-behaved programs. Program monitoring can also offer the ability to see the number programs active and what state each program is in.

Table 1 List of Monitoring Attribute Areas.

In order to allow the developed system to provide user with reliable and effective data about the monitored value within computer monitoring system, it is crucial to identify the origin and where the value could be retrieved from. This monitoring or notification value could be retrieved from various sources depending on the type of the value; either it is a raw data or the processed data (information).

These types of notifications typically are produced by :

- operating system error reporting mechanisms
- log file filtering tools
- vendor or custom-developed management software
- custom error reporting mechanisms from service or application programs (including tools)

The main source of the monitoring attribute values could be triggered from the operating system reporting mechanism; a log file that sustained the monitoring process and its value throughout the system upon system up-time. Its provides statistics about the system performance which includes :

- total resource use over time (CPU, memory [used, free], disk [used, free])
- file system warnings (low free space, too many open files, file exceeding allocated size)
- systems and hardware devices information and details
- hardware availability (modems, network interface cards, memory)
- file system status (where mounted, free space by partition, open files, biggest file) over time and at specific times
- disk counters (input/output, queue lengths) over time and at specific times

After configuring the main source of obtaining the required information and value needed by Computer Monitoring System Software, it is also identical to understand the computer problem areas so that the author could gain depth understanding on the behavioral and system activities that may interrupt system performance. To address computer problems, first establishment of appropriate expectations for the amount of CPU resources the system should be using is important. Then, we need to distinguish whether sufficient CPU resources are available and recognize when the system is consuming too many resources. As example, determining the amount of CPU resources the Oracle instance utilizes in three cases, when the system is:

- Idle
- At average workloads
- At peak workloads

Workload is an important factor when evaluating system's level of CPU utilization. During peak workload hours, 90% CPU utilization with 10% idle and waiting time may be acceptable; 30% utilization at a time of low workload may also be understandable. However, if the system shows high utilization at normal workload, there is no room for a peak workload. For example, Figure 18-1 illustrates workload over time for an application having peak periods at 10:00 AM and 2:00 PM.

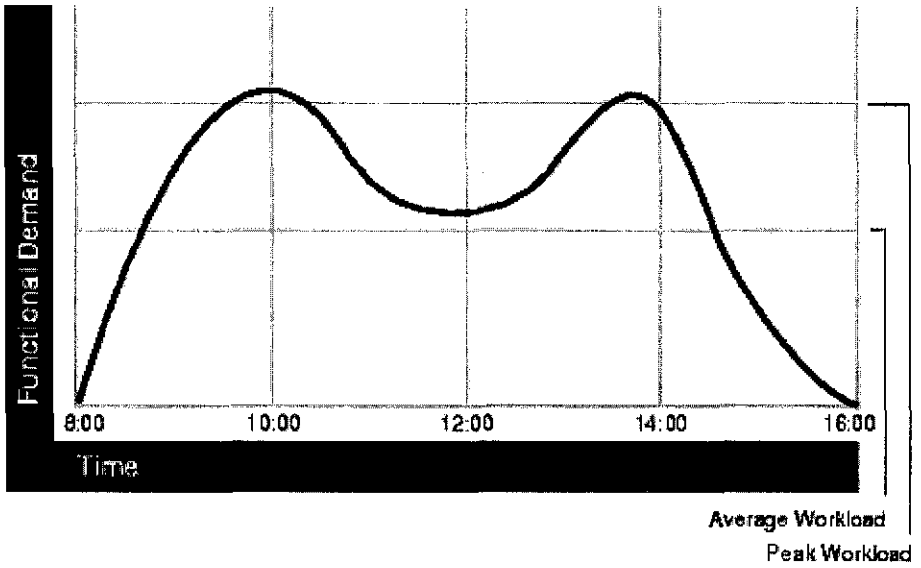


Figure 1 Workload Over Time for an Application

For the user acceptance analysis, questionnaire also will be conducted to obtain a precise feedback from the existing similar monitoring system users. In this case, two parties are seen as the suitable candidates; the normal computer user and the expert user who have knowledge about the system monitoring and very familiar to the subject. Among the areas to be included in the questionnaire will focus on the reliability of the product – how the existing product did help the user in providing the correct and precise monitoring data, efficiency – how does the product provide necessary information that could help the user to reduce their time consumption, energy and cost saving when dealing with computer maintenance, constraints – what influence the overall performance and suggestion for improvements from the user perspectives. Based on the feedbacks, system performance is analyzed and possible improvements are determined.

As the system comes to the construction phase, therefore it is crucial to identify the source of the attributes in advance before the coding could be construction to build the system. The data is basically could be obtained from Windows Management Instrumentation (WMI) which provide standard method of accessing system information, performance counters and application monitors. It provides a way to interact with system management information and the underlying WMI APIs. WMI is used primarily by system management application developers and administrators to access and manipulate system management information.

4.1.1 Windows Management Instrumentation

WMI has a wide range of functionality within the computer. It can be used to build tools that organize and manage system information so that administrators or system managers can monitor system activities more closely. For example, development of an application using WMI that pages an administrator when a Web server crashes.

Based on the analysis the author has conducted in understanding WMI, there are several components that make up the WMI architecture, which are :

- Provider
- Consumers
- Common Information Model (CIM) repository
- Common Information Model object manager (CIMOM)

These components are built upon the CIM. It is object-oriented, system-independent, and application-independent. It provides a hierarchical schema to define objects in a managed computing environment. The CIM defines objects in the managed environment through classes. These classes include methods to describe behavior and properties to describe data. Some objects included in the CIM are applications, networks, printers, and drivers.

A provider is an intermediate agent between the system to be managed (such as an operating system, service, application, device driver, and so forth) and the CIMOM. Its purpose is to extract management information from the system using whatever interfaces a system presents for management. Through various COM interfaces, providers can supply data to WMI, handle requests from system management applications, and generate event notifications. The provider also maps the management information and interfaces to the object classes that are defined and stored in the CIM repository.

Consumers are system management applications, such as Microsoft Systems Management Server (SMS), or third-party applications or scripts. Consumers need to know only the classes of the objects that they want to get information about, where the information comes from and how it is actually obtained are hidden and not relevant. Consumers can obtain information about the computer, operating system, applications, and devices as well as information available through other management protocols.

The CIM repository is an object database where defined objects — such as static class definitions and instances that are used to access and manipulate system management information — are stored. The CIM repository is managed by the CIMOM, which acts as an agent for object requests. The CIMOM tracks available classes and determines which provider is responsible for supplying instances of these classes. The CIMOM and the CIM repository are represented by a system service called WinMgmt and are accessed through a set of COM interfaces.

The following figure illustrates how these components work together.

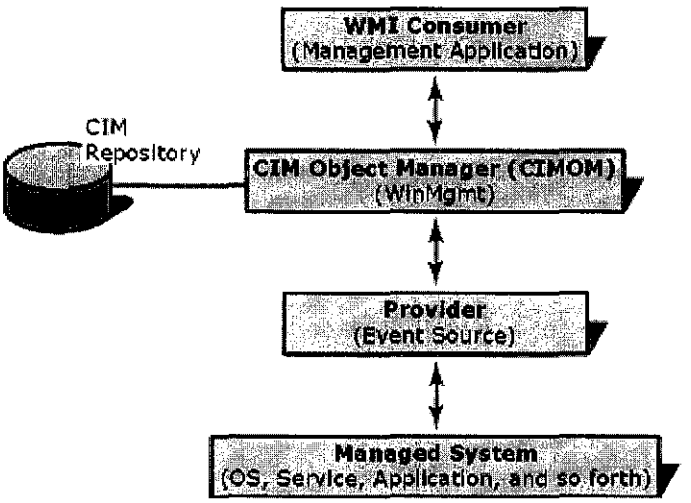


Figure 2 WMI Component Interaction

When a request for management information comes from a WMI consumer to the CIMOM, the CIMOM evaluates the request, identifies which provider has the information, and returns the data to the consumer. The consumer requests only the information it wants. It never knows the exact source of the information or any details of how it is extracted from the underlying API.

From the previous analysis, WMI could be classified into numbers of classes such as Win32_OperatingSystem, Win32_ComputerSystem and Win32_DiskDrive. Each of the classes contains specific information regarding the attributes that belongs to it. As example, Win32_DiskDrive can provide the user with the partition details, disk volume and space remain and many more.

The next table will list out the required information needed by the monitoring software system and the original resource of the data extracted from the computer system.

Attribute/Function	Source
Computer Baseboard Properties	Win32_BaseBoard
System Uptime	Win32_OperatingSystem
Memory Property	Win32_MemoryDevice
Memory Performance	Win32_PerfFormattedData_PerfOS_Memory
Motherboard Devices	Win32_MotherboardDevice
Page File Properties	Win32_PageFile
Page File Performance	Win32_PerfFormattedData_PerfOS_PagingFile
Page File Usage	Win32_PageFileUsage
Chassis Type	Win32_SystemEnclosure
BIOS Information	Win32_BIOS
Fan Info	Win32_Fan
Operating System Object	Win32_PerfFormattedData_PerfOS_Objects
Operating System Performance	Win32_PerfFormattedData_PerfOS_System
Processor Type	Win32_Processor
Processor Usage	Win32_PerfFormattedData_PerfOS_Processor
Power Supply Unit	Win32_PowerManagementEvent
Cache Memory Properties	Win32_CacheMemory
Cache Performance	Win32_PerfFormattedData_PerfOS_Cache
Hardware : Sound Card	Win32_SoundDevice
Physical Memory Property	Win32_PhysicalMemoryArray
Physical Memory Configuration	Win32_PhysicalMemory
Enumerating Processor Info	Win32_Processor
Enumerating Onboard Devices	Win32_OnBoardDevice

Modem Information	Win32_POTSModem
Plug and Play Devices	Win32_PnPEntity
Plug and Play Signed Driver	Win32_PnPSignedDriver
Pointing Devices Properties	Win32_PointingDevice
Keyboard Properties	Win32_Keyboard
IRQ Setting	Win32_IRQResource
SMBIOS Information	Win32_BIOS
Bus Properties	Win32_Bus

Table 2 List of Monitoring Attributes and Source

Using the obtainable information provided by the each of the above source, the data will be processed by each module of the monitoring areas. The author has decided to separate the data processing into several sub-modules so that the physical design of the system could be implemented incrementally; sub-module by sub-module, before the integration of each functionality is made. This could avoid unnecessary system development error as well as could detect the origin of the defect and deficiency of the program.

Monitoring attributes and object is divided into six unit which are :

Attribute/Object Class	Attribute/Object Monitored
Summary	Contains summary of basic details for each unit
CPU	Processor ID Processor Type Processor Architecture Processor Clock Speed Processor Family Processor Socket Type Percentage Use

	Processor Stepping
	Processor Description
	Processor External Clock
	Processor Voltage Caps
	Processor L2 Cache Size and Speed
	Processor Manufacturer
	Processor Load Percentage
Main Board	Motherboard ID
	Motherboard Primary Bus Type
	Motherboard Secondary Bus Type
	Baseboard Details
	Bus Number
	Bus Type
	Bus Description
	Device ID
	Device Name
Memory	Physical Memory Description
	Physical Memory Maximum Capacity
	Physical Memory Device
	Physical Memory Manufacturer
	Physical Memory Type Details
	Physical Memory Speed
	Physical Memory Device Locator
	Physical Memory Capacity
	Physical Memory Available Bytes
	Physical Memory Cache Bytes
Sensor	Active Cooling
	Availability
	Device ID
	Name
	Status Information
Devices	Device Description
	Device Type
	Device Model

	Device Name Device Tag Device Version

Table 3 List of Monitoring Attribute According to Class

During this process, there will be several conditions that need to be checked. Since the system involved the continuous changed as the time flow, therefore sometime during the computer system executing its processes and do various task, there might be overloaded in the computer system resource usage occurred. Indirectly the monitored value will signified the inefficiency in computer thus providing the application with the current value of the computer system. This will indicate the inefficiency in system performance and as a result the Computer Monitoring System needs to point out the inefficient area to the user.

4.2 Logical Design Result

4.2.1 Computer Monitoring System Logical Design

The system logical design begins with the identification of basic functionality that the developed system should be provided. From the analysis conducted, a details system behavior, concept and design is identified and could basically presented using the use case diagram :

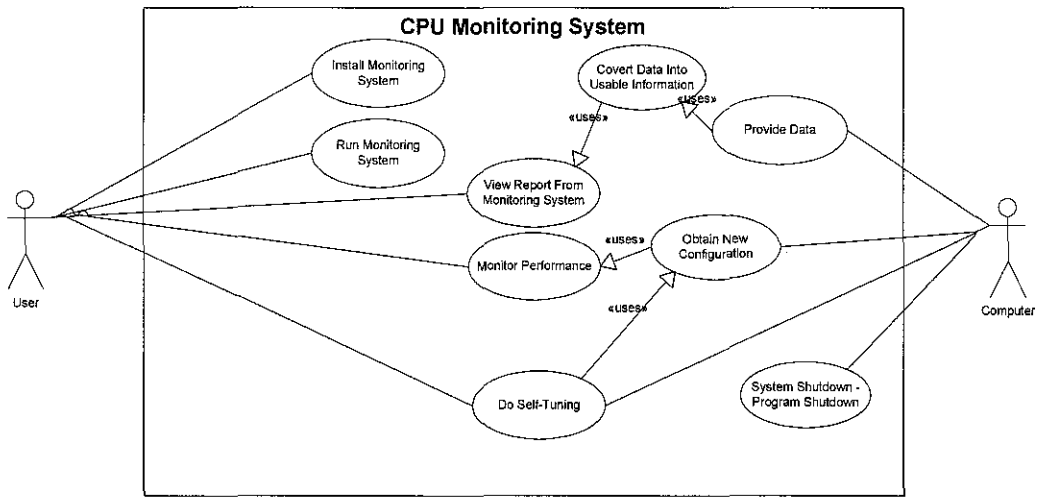


Figure 3 Computer Monitoring System Use Case Diagram

From this figure, behavior of each object involved (referring to the actor) could be triggered so that the basic functionality and what the product can actually do can be well measured. Aside from that, based on the analysis conducted, the data flow process for overall system starting from the computer user itself until the deepest part of the computer which is the operating system. This data relocation and transfer between each entity need to be clarify so that the main source of the data as well as the requested point can be assigned correctly to maintain the reliability.

Description for **'Installing Monitoring System'**

The user will firstly need to install the Monitoring System in order to use it. This is because the developed system is classified as standalone system and therefore it needed to be residing within the computer system to make it able to interact with the computer.

Description for **'Run Monitoring System'**

User run/execute the system since the system could not be automatically run as service during the computer start up.

Description for **'Provide Data'**

Upon running the program, computer (specifically the operating system within the computer) will provide the required data triggered by the monitoring system. The data include all the information regarding the system software and hardware used, the current monitoring attribute values and any inefficient monitoring attribute data condition occurred during the process.

Description for **'Convert Data into Usable Information'**

The Computer Monitoring System obtained the data provided by the computer and processes the data according to the required variable/attribute value assigned within the system. Since the data obtained is in raw form, hence the data firstly need to be converted into form that can be understood by user using the displayed interface implemented in the developed system.

Description for **'View Report from Monitoring System'**

User view the data provided by Monitoring System. It includes CPU performance details, Memory Performance, System Summary Performance, Storage Performance, System Sensor (Temperature and Fan Speed) and more.

Description for **'Monitor Performance'**

User could continuously monitor the performance through the Computer Monitoring System if the system is still running. The monitored value will be updated frequently to provide reliable and efficient data that indicates the current computer system performance.

Description for **'Do Self-Tuning'**

If the user indicates any unwanted condition on their computer indicates by the Computer Monitoring System, they may proceed in doing self-tuning to the computer through BIOS configuration, hardware check, optimizing computer process and more. If they did not plan for any tuning yet for the current time, they might be considering tuning in during the next time they run the computer or tuning it after the computer is shutting down.

Description for **'Obtain New Configuration'**

If the user does the tuning at the same time the program was running, there might be changes in monitoring attribute setting within computer system, and therefore, the system need to obtain the new configurations of the computer system.

Description for **'System Shutdown – Program Shutdown'**

User turn off the computer as well as closing the monitoring program.

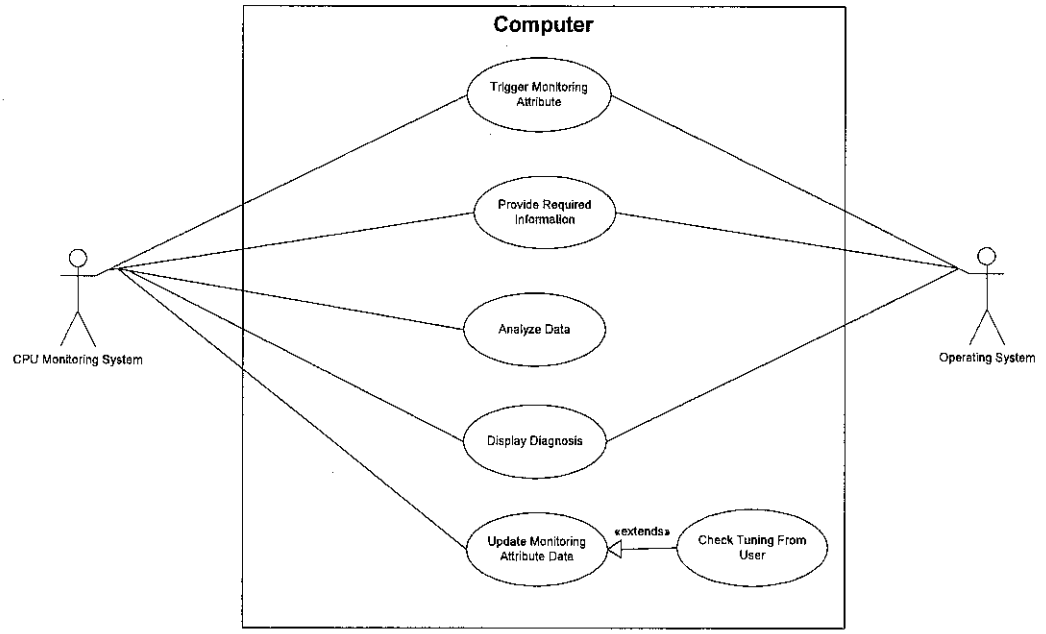


Figure 4 Internal System Use Case Diagram

Those system behaviors and functionality could be concluded into series of step which are :

Step 1 : Installing CPU Monitoring System
Step 2 : Run the system
Step 3 : Obtained monitored information
Result : System software and hardware details
Step 4 : Do self-tuning upon inefficient data
Step 5 : Go back to Step 3

Next step is to identify required process involved in Computer Monitoring System in order to provide a clear view and understanding on how the system works. This includes identification of the data, process executed within the system and assessment made throughout the processes. Computer Monitoring System process consists of six main processes starting with system installation on the respective computer until the system enable to display the information on the monitoring attribute and the system hardware and software information.

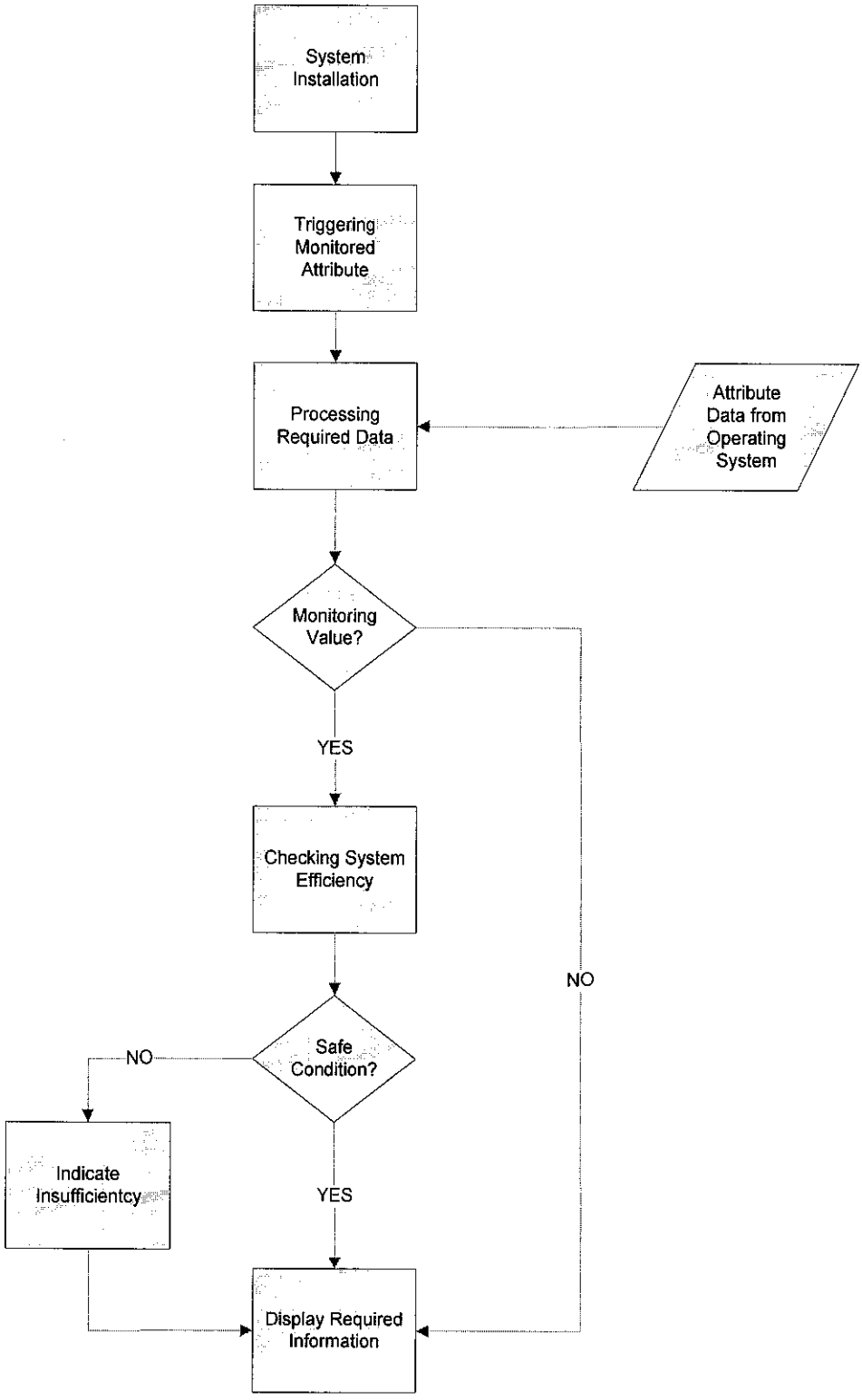


Figure 5 Computer Monitoring System Process Flow

Computer Monitoring System processes begin with the installation of the software by the user into the respective computer system. The installed location should be occupied with Windows XP Operating System as the product only support Windows XP platform. Different operating system platform required/provide different measurement value to the monitoring system, thus if the monitoring system is implemented on the other operating system platform, the system cannot be functioning and provided precise and effective information as well as it might be a problem of information unavailability.

Running the monitoring system will enable the system to trigger the required information needed to manipulate it into computer status, system software and hardware details and others overall monitoring subject; a useful information that can be understood and used by the user to perform system tuning. This could be clearly illustrate using the below diagram :

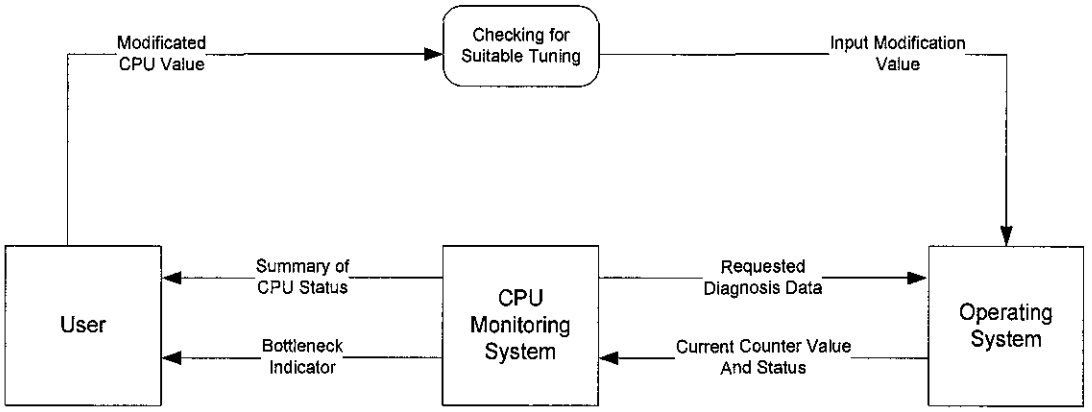


Figure 6 Computer Monitoring System Dataflow Diagram.

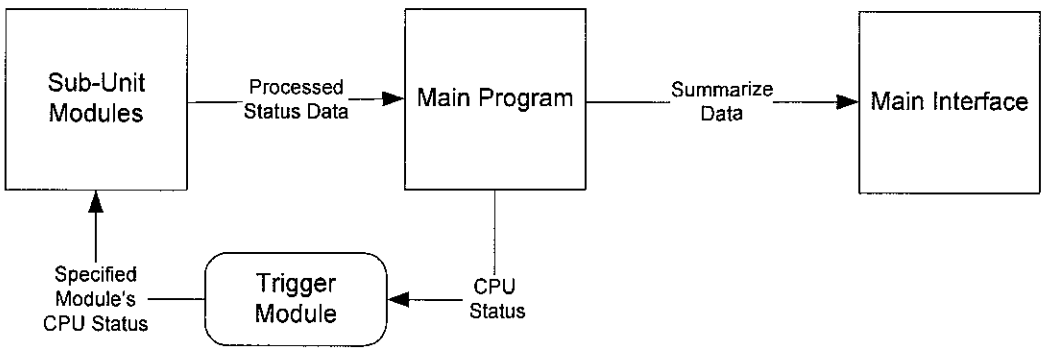


Figure 7 Internal Data Flow.

4.3 Physical Design Result

4.3.1 Computer Monitoring System : Physical Design

The implementation of the logical design into the physical design basically involved the development of system hardcode and the interface of the application. This process typically divided into three major areas of development; construction of user interface using HTML coding (development of HTML application) and synchronization of the application layout through implementation of cascading style sheet (CSS), scripting using VBScript and the integration of both three elements; CSS, HTML application and the script.

Cascading Style Sheet Construction

CSS is basically a HTML style sheet standard set by W3C and Internet Explorer that allow the user to attach styles to HTML tags to revolutionize the document. The author has used this approach and embedded the technique into my project so that the author can easily manage the layout of the interfaces especially when dealing with changes that involved tremendous modification of the codes. With CSS, color, fonts, style and spacing of the application interface could be standardized thus providing superior navigation and user-friendliness of the application. Moreover, margins, line spacing and placement of the design element could be controlled easily particularly when the author dealt with great numbers of sub tables, link and references.

Linked style sheets allow me to apply standard style sheet to different document that the author work with, giving the application a standard looked. Additionally, if the author wants to change the style throughout the document, the author will just need to update the CSS in one place in order to apply changes throughout the application.

HTML Application Development

Tagging and Scripting

Foremost, in developing HTML application, the process of creating HTML tag in the code document is important, as it always be the first line in any code. Tag will signal the browser that the subsequent txt should be treated as an HTML document. Fundamentally, <HTML> tag is included to let the browser know how to interpret what follows. This tag is attached at the beginning and at the end of the coding to terminate tag container as well as indicating that the whole document has been received.

While for <HEAD> tag, on the whole used as holder for the code document information whereby it usually hold the title page through <TITLE> tag, keyword for search engines and script using <SCRIPT> tag and <META> tag. The last important tag that the author also includes was <BODY> tag that marks the start of the main body of the HTML document. It controls how the application developed appears on the screen. The <BODY> tag part is also played important role in author's script development, as it create framework on which the script is build. Before the author used VBScript to create my active content, the script should be associated with the item on the HTML application (HTA). In the <BODY> section, it serves the application by tying together the script and the objects because it names everything in which the script embedded need to access.

Script Development

This is the basic development of VBScript in Computer Monitoring System as it defines and identifies the current computer operating system.

```
strComputer = "."

Function GetWMIServices()
      Set GetWMIServices = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" & strComputer &
    "\root\cimv2")
End Function
```

The next step required the correlation between VBScript that the author currently build with the previous HTML codes and tag used in HMA developed. This part is constructed using functions which will synchronize the layout of the application between what have been declared in CSS file, HTML file and VBScript file.

```
Function DisplayOutputHeader(strHeader)
      document.all.headOutput.innerText = strHeader
End Function
Function DisplayOutput(strOutput)
      document.all.divOutput.innerHTML = strOutput
End Function
Function GetTableHeader()
      str = "<TABLE class='tblOutput'>"
      str = str & "<THEAD><TR><TH width=30%>Property
      </TH><TH>Value</TH></TR></THEAD>"
      str = str & "<TBODY>" & vbCRLF
      GetTableHeader = str
End Function
Function GetTableFooter()
      str = "</TBODY>" & vbCRLF & "</TABLE>" & vbCRLF
      GetTableFooter = str
End Function
Function GetRow(PropName, PropValue)
      str = "<TR>"
      str = str & "<TD class='PropName'>" & PropName &
      "</TD>"
```

```
        str = str & "<TD>" & PropValue & "</TD>"
        str = str & "</TR>" & vbCRLF
        GetRow = str
End Function
```

The construction of each areas of information retrieval, in this case each of the classes that offered by the HTML application links, was created by separating each area with different function so that it will be more manageable and easy to understand.

Problem Faced

- Setting up interface is quite difficult since constructing HTML application did not provide design layout thus required the developer to run and test the codes during editing.
- A little tweak in coding especially during integration process between CSS, HMA and VBScript may cause big changes and sometimes the codes not executed and error occurred.

The basic layout of the application could be divided into three main pane; the upper left pane consist of the list of drop down option that is classified according to the areas of monitoring attributes such as motherboard and drives/disk, the lower left pane contains various user control commands that enable the computer user to remote the system using the application and the left pane in which the main output panel that will display the information according to the selected function triggered from the selection option.

Information will be displayed in the main output panel using table view. The table is divided into two column named as the property column and the value column. Property column used to display the attribute's name while the details about the attribute will be displayed in the value column according to the respective and type of value that the WMI database holds.

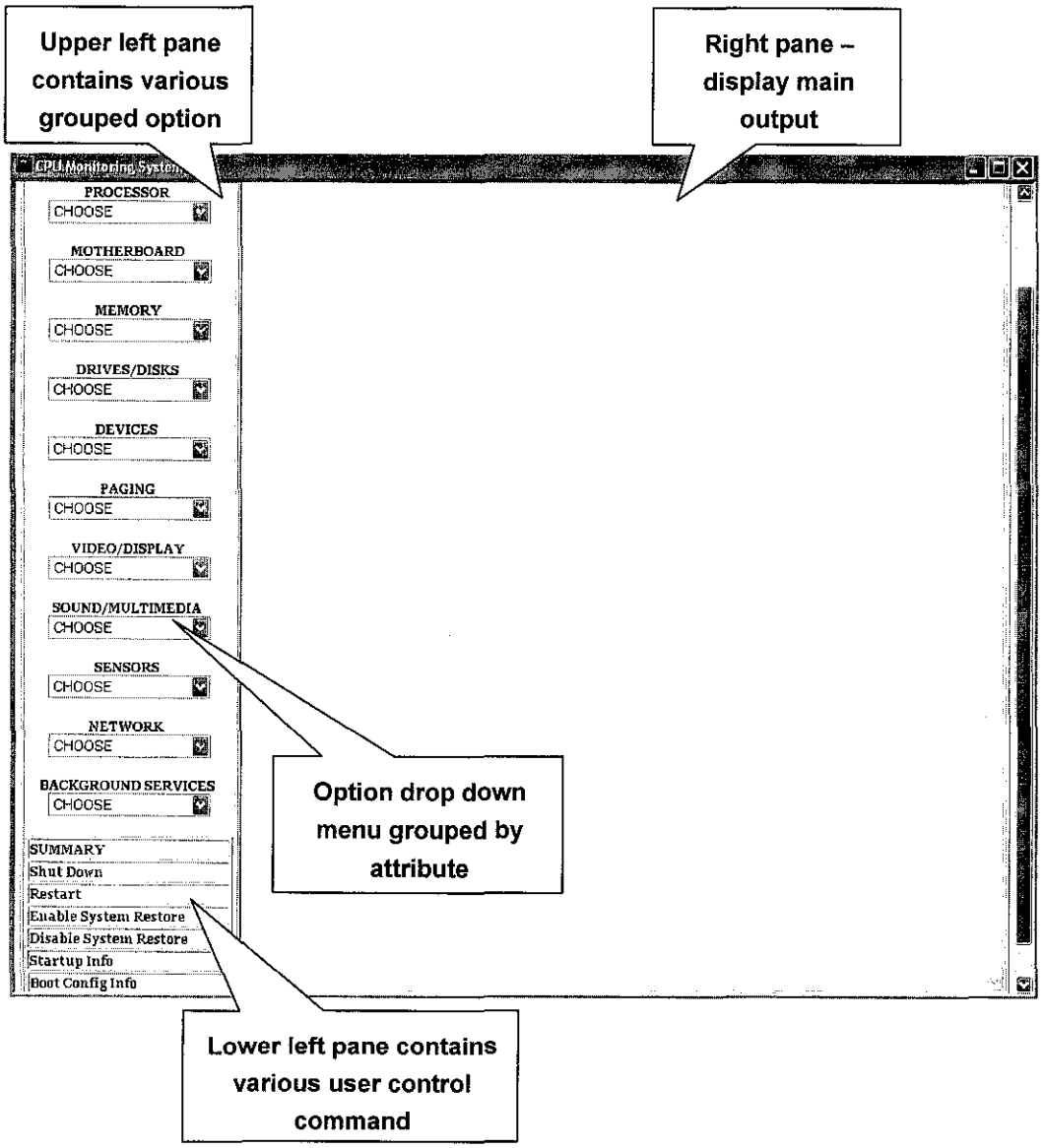


Figure 8 Main Interface

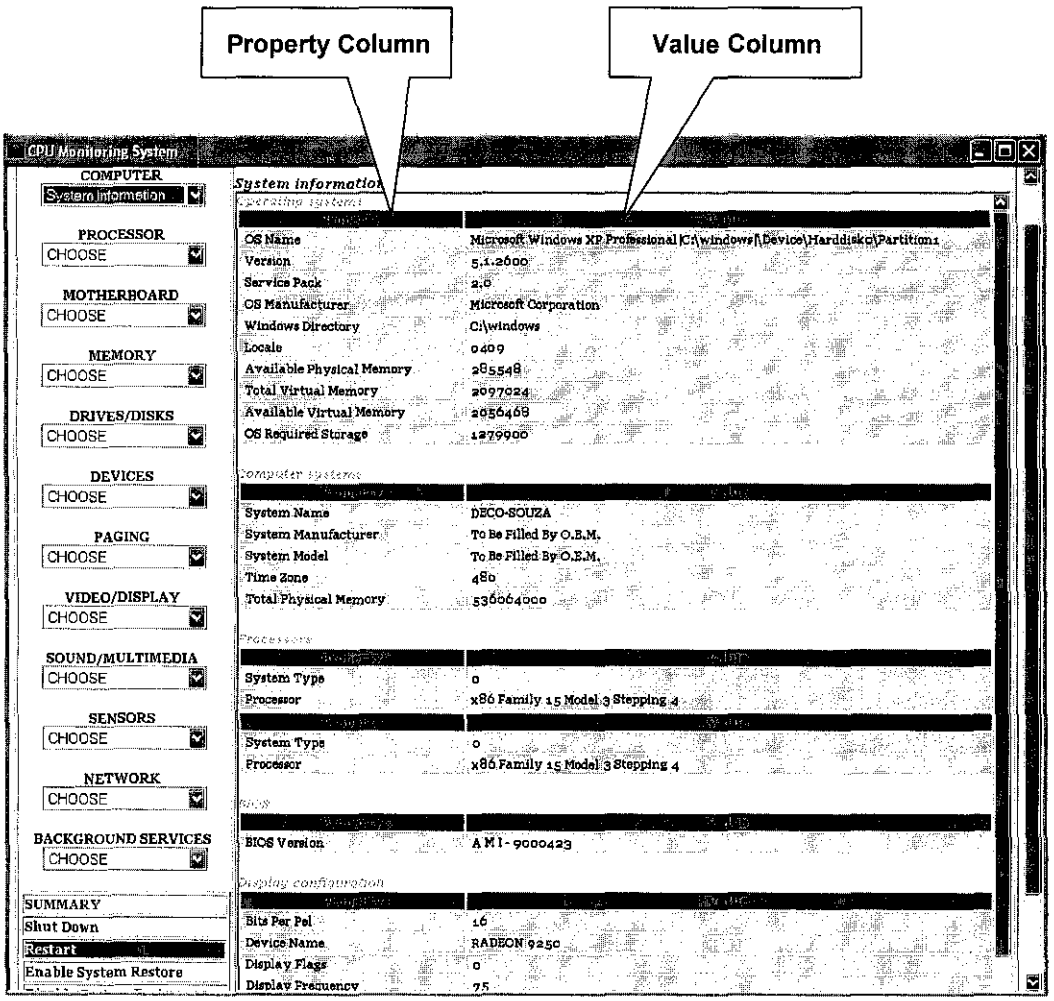


Figure 9 Executed Prototype Interface

4.4 Evaluation (User Acceptance Test)

The user acceptance test was performed to verify the usability and efficiency of the developed system as well as other similar CPU monitoring software in assisting the computer user gaining information and monitoring the computer current condition. This test was conducted to measure the effectiveness of the system's provided information to the user.

The evaluation was conducted by ten selected computer user; in which the subject must be familiar with computer monitoring software such as Everest, CPUIZ, Motherboard Monitor and etc. With the familiarity in using the off-the-shelf CPU monitoring application, this will provide a better view on what this system is all about and enable them to test and manipulate the system in order to test the usability and the efficiency of the system.

The test will require each of the evaluator to run the application on separated computer (basically computer user using Windows-based operating system) so that they can give their individual opinion. The areas in which the system is evaluated consist of the realism of prototype, user control and freedom, aesthetics and minimalist design and the usability of the system.

All evaluators are asked to test the system and fill up a form (User Acceptance Form) as embedded in the appendices.

Each of the evaluation characteristics are explained to the evaluators. They are :

- Realism of prototype : The overall performance of the prototype.
- User control and freedom : Navigation, user-friendliness and interface layout
- Aesthetics and minimalist design : The system content. Does not contain information and irrelevant information.
- Usability of prototype : System functionality and area of coverage.

4.4.1 Evaluation Result

Table below summarizes the evaluation that has been made by all ten evaluators. Please note that the score are on the scale of 5.

Criteria	Poor	Below Average	Average	Meet Expectation	Excellent
Realism of Prototype	0	0	2	7	1
User Control/Freedom	0	0	1	6	3
Aesthetics and Minimalist Design	0	0	4	6	0
Usability of Prototype	0	0	2	6	2

Table 4 User Acceptance Evaluation Result

4.4.2 Evaluation Result Analysis

Once evaluators have completed filling the evaluation form, they were asked some opinions on evaluation they have made. The full score for each criterion is 5. Figure below summarizes the evaluation result.

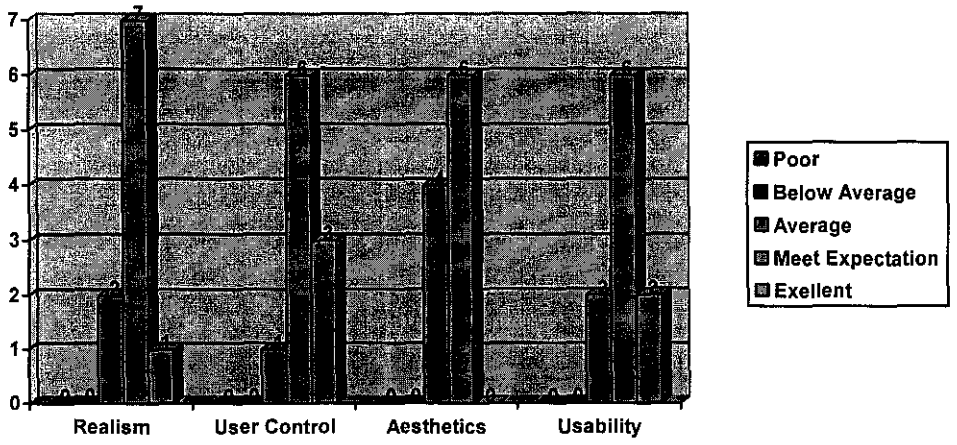


Figure 10 Evaluation Result Analysis

Realism of Prototype

The average score for this criterion of evaluation is 3.9. Based on the informal discussion with the evaluators regarding the overall performance of the system, most of them agreed that the system is presentable and suitable to be used either by expert or the beginners. Some of them however say that the system is presentable but still need some improvement especially in distinguishing what is suit for the beginners and what is suits for the expert users.

User Control and Freedom

The second criteria scores 4.2 in average. All of the evaluators agreed that the system provides user with control and freedom in navigating from one function to another as well as providing the direct access on computer control like enabling and disabling the system restoration. They also pleased with the simplicity of the navigation between one function to another as well as the less-complex step required to obtain necessitated information.

Aesthetics and Minimalist Design

The average score for this criterion is 3.6. Although all the evaluators agreed that the information provided in the system relevant to the scope and the areas of project coverage, some of the information might be irrelevant to the beginners as these information might hard to be understood for the user with only basic knowledge about the computer.

Usability of Prototype

For the last criteria of evaluation, the analysis shows the average score of 4.0. Most of the evaluators agreed that the system prototype covered mostly all computer aspects that need to be monitor in order to achieve optimization in computer usage. They also suggest that in the future, this system could improve its functionality by providing the troubleshooting and recommendation for computer tuning.

CHAPTER 5
RECOMMENDATION AND CONCLUSION

5.1 Summary of Project

By incorporating scripting technologies, HTML application and manipulating Windows Management Instrumentation functionality, it is believed that the author had achieved the initial aim and objective that are specified earlier. This project could assist computer user in understanding the requirement needed in order to maintain a good computer configuration. The author also had learned new lesson regarding computer configuration and programming techniques during the development and research phase of the project, which can be useful in the future.

Towards the completion of the prototype, the methodology has provided the author as a powerful guideline of the project. The research and construction phase is very crucial in completing the project in order to meet the scope defined.

5.2 Suggested Future Work for Expansion and Continuation

There are several suggestion and recommendations that can be done to this project toward future improvement of efficiency, functionality and flexibility.

5.2.1 Provide troubleshooting functionality

Currently, the prototype only capable in providing the information regarding the computer condition based on the triggered computer area. Therefore, if the user stumbles upon insufficiency in any part of the computer configuration, they might have no idea in what should they do and react. By adding this functionality, user might have guidance to figure out themselves on the potential problems that lead to the insufficiency as well as on how to overcome those problems. This troubleshoot function will be a 'nice-to-have' function to the system.

5.2.2 More clear and comprehend information provided

Since the prototype covered most of the computer component areas that related to the performance of overall computer, it provides the user with massive information. The author believed that this massive information only reliably and applicable to the understanding of computer expert user, not for the novice or beginner. As a result, some information might be useless if it is not represented in more understandable ways. In order to make this information more comprehend to this type of user, the prototype should be improved in such way that it will provide a clear description on the monitoring areas – what it is all about and which area the attributes might affect.

5.2.3 Capability in using at different platform

The prototype hardcode is basically build based on the latest Windows-base platform, which means Windows platform that support the Windows Management Instrumentation functionality. This hardcode will directly extract computer information from this WMI. As the prototype is builds for Windows-base platform, the application cannot be executed under any other operating system platform. By adding this portability capability in the system, this will provide more value as well as enable various computer users to be able to use the application.

5.3 Conclusion

Performance Monitoring is good as it will continuously or daily monitor the health of the CPU or Central Processing Unit which is indeed the brains behind any computer system. The bottleneck and computer overloading could also be identified more occasionally, reducing computer system crashing possibility. When dealing with the component problem, this system could beneficially provide troubleshooting feature to the user in several areas, thus reducing the cost and time if the user directly send the computer to service centre. For a long term benefits, the monitoring system could become a good supporting tools in providing the requirement for future upgrade especially in determining the CPU load, memory requirement usage and disk space. Improvement could be made for the existing monitoring software as well as the monitoring system project in the future whereby additional functionality that will enable the user to directly make system tuning and variable modification such as over clocking and changing parameters value through the system could also be included. Others, the developer maybe could consider developing computer monitoring system that compatible in various operating system platforms without need to make changes in system physical design development.

REFERENCES

[1] *Performance Monitoring,* Ronald Rose. Retrieved 10th October 2005 from the World Wide Web : http://www.emu.edu.tr/english/facilitiesservices/computercenter/bookslib/Unix%20Unleashed/unx29.htm

[2] *System Monitoring in Multitasking Environments,* Dave Wallenberg. World Wide Web : http://www.os2ezine.com/v1n8/monitor.htm

[3] *Graphical Monitoring of CPU Resource Consumption in Java-based Framework,* Andrea Camesi and Jarle Hulaas Walter Binder. (Software Engineering Laboratory Artificial Intelligence Laboratory)

[4] *Performance Testing Overview*, J.D. Meier, Srinath Vasireddy, Ashish Babbar, Rico Mariani, and Alex Mackman. http://msdn.microsoft.com/library/en-us/dnpag/html/scalenetchapt16.asp

[5] *A Portable CPU-Management Framework for Java in IEEE Internet Computing,* W. Binder and J. Hulaas Vol. 8, No. 5, September/October 2004, pp.74-83.

[6] *Monitoring Tools Profile: WhatsUp Premium Gold,* Nelson King, October 12,2005, World Wide Web : http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd

[7] *Using JConsole to Monitor Applications,* Mandy Chung, December 2004

[8] *Look Inside For More Information About Computer Monitoring Sleep Setting,* http://www.energy-solution.com/off-equip/OE2_Monitor.pdf.

[9] *Dynamic Processor Reallocation and Dynamic Processor Resilience,* April 16, 2002, http://docs.hp.com/en/1119/dynamic.pdf.

[10] *Balancing Performance, Energy and Quality in Pervasive Computing*, Jason Flinn, So Young Park and M. Satyanarayanan, Carnegie Mellon University and Intel Research Pittsburgh, http://notrump.eecs.umich.edu/group/papers/icdcs.pdf.

[11] *Expert System for PC Overcloacking,* Mohammad Radzi bin Zohri, Universiti Teknologi Petronas, January 2005.

APPENDICES

CPU MONITORING SYSTEM EVALUATION FORM

This questionnaire is performed to evaluate the effectiveness and the usability of the prototype for the CPU Monitoring System project. The evaluation required the candidates to evaluate and rate the system according to four different attribute, namely realism of the prototype, user control and freedom, aesthetics and minimalist design and usability of the prototype.

Realism of prototype : The overall performance of the prototype.

Poor	Meet expectation	Average	Good	Excellent
O	O	O	O	O

User control and freedom : Navigation, user-friendliness and interface layout

Poor	Meet expectation	Average	Good	Excellent
O	O	O	O	O

Aesthetics and minimalist design : The system content. Does not contain information and irrelevant information.

Poor	Meet expectation	Average	Good	Excellent
O	O	O	O	O

Usability of prototype : System functionality and area of coverage.

Poor	Meet expectation	Average	Good	Excellent
O	O	O	O	O