FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA

GRID APPLICATION

YAP CHIN WAI
WEK020291

Under the supervision of
Mr Ang Tan Fong

Moderator
Mr Liew Chee Sun

Session 2004/2005

A Project proposal and report in partial fulfillment of the requirement for the degree of bachelor of Computer Science
University of Malaya

Abstract

Since 20 century, Grid application became more popular in the section that required complicated computing system. Science is the most widely section that involved in the Grid technology for computational power and also flexible management for intensive data produced during researches. Until now, the Grid technology has evolved into calculation system for analyzing the data for natural phenomena like Earthquake. By using the Data Grid technology, can build up an efficient management, discover, and allocation system of distributed data resources over the network within the Grid environment.

This project is to develop a Data Portal that is residing in the FSKTM server. This portal provides a simple web-based user interface that allows users for searching data from multiple databases that located in Windows and Linux platforms separately. OSGI.Net and C#.Net programming language will be used for develop the system. The databases involved are Microsoft SQL Server and MySQL.

In the back-end of this system, the databases will have a replication function that ensures performance and reliability of the system. Replication allows data replicated from one database to another. When a database is overhead due to too many concurrent accesses, some of the files currently accessed can be replicated to one less working database for better running. This also makes sure the connection between the users and databases is still running without hang out.

Acknowledgement

For develop the FSKTM Data Portal, I have got a lot of helps, advices, and contributions from many people. In here, I would like to express my appreciation for these persons. First at all, I was very thank you to my supervisor, Mr Ang Tan Fong that let me doing this project and guides me through the process of identify the scope of this project and writing the documentation.

Secondly, I would like to express my gratitude to my project moderator, Mr Liew Chee Sun. He had given me an inspiration to propose the FSKTM Data Portal concept and also gave me plenty of useful advices to implement the system.

I also wanted to thank my project partner, Tan Kien Peng for sharing his knowledge and information with me and accompany me until the project finished. Same gratitude also been expressed to my other course-mates that helped me in my other assignment when I was busy with my thesis project.

CHAPTER 2: LITERATURE REVIEW

CHAPTER 5: SYSTEM DESIGN

CHAPTER 6: IMPLEMENTATION

CHAPTER 7: SYSTEM TESTING

CHAPTER 8: SYSTEM EVALUATION

List of Figures

List of Tables

CHAPTER 1 - INTRODUCTION

1.1 Overview of Grid Technology

The immediate ancestor of Grid is metacomputing at around year 1990. The metacomputing is a project that combined several interconnected supercomputer in US to provide powerful processing capability. In 1995, two metacomputing projects called FAFNER and 1-WAY were being conducted and somehow their existing has influenced the evolution of some kind of Grid technology today.

Grid technology is implemented based on the concept of connected dispersed computer resources in world wide for collaboration purpose and for solving complex problems with impressive computing power within Grid. Grids are built from collections of different independent services/applications. These services are uniformly available throughout the distributed environment of the Grid. A set of grouped services is refers to what we called middleware. Available middleware included Globus toolkits, MyGrid, MS.Net Grid, Achelmi and others.

Until now, Grids technology implementations are focused at scientific, engineering and biology aspect. The final goals of Grid are to increase the processing power and use efficiently of resources and to reduce the time of execution, besides allows the applications or resources can be accessed throughout the geographical boundary.

Science Portal is one of the most exciting areas of Grid implementation. Science Portal allows scientists, researchers or high-performance application users to execute some large-scale process and access resources through a users-friendly web page browser. The research of Grid Portal has been carried on at the late 1990s that initially to design a web based front end for the Grid applications and resources. Science Portal is the derived portal from Grid Portal that specified on solutions for computational scientific problems that required high processing power to solve and related resources usages during the execution.

There are several architectures for building Grid Portal available on the internet. Open Grid Services Infrastructure (OGSI) is one of the most standardized models for Grid Portal development. OGSI contains a lot of mechanisms for develop and manage the Grid services within a Grid Portal.

1.2 Project Objectives

Here we will list out the main objectives to be achieved in this project.

- Develop a Data Grid Portal for Faculty of Computer Science and Information Technology, University of Malaya.
- This application can run on 2 system platforms: Linux and Windows.
- Users can access this application from a standard web browser.
- Provides a reliable search engine for users to gather related resources.

1.3 Project Scopes

The FSKTM Data Portal is a web interface that developed in FSKTM server and consists of 2 kinds of databases that run on different platforms (Linux and Windows). The following are the scopes to be reached:

- Provides a standard and user-friendly interface that allow user to gather resources easily.
- Allows users download resources from multiple sites.
- Allows users to share their resources through uploading resources to databases.

1.4 Problems

- **Inexperience to Grid application**

  Grid can still be defined as a new technology and not yet become common for everyone but for several specified community. In personal, the Grid concept and implementation are a little complicated for me and its applications are always large in size.

- **Lack of programming skills**

  C#.net and Java are 2 powerful programming for writing the Grid applications. However they are new for me as I never touch with them before and now have to use them to write a whole system.

- **Times**

  Due to the about 2 factors, I need more time to understand the Grid concept first and then have to deal with the new programming language to build a system.

- **Resources**

  Although we can find out various of information through internet about Grid technologies, concepts, applications and tools for develop Grid, they are mostly complicated and difficult to read.

1.5 Report Organization

This section will give a briefly introduction of every chapter in this report for better understanding of the readers.

1.5.1 Chapter 1 "Introduction"

This chapter gives a brief introduction of the whole project including project overview, project objectives, project scopes, limitations, and project schedule chart.

1.5.2 Chapter 2 "Literature Review"

This chapter consists of all the researches on the project. This included the studying of the Grid definition, architecture, and implementation from other Grid research centers. Besides, results from studying the existed Grid Data Portal applications are listed here and the tools (or programming languages) also recorded in this chapter.

1.5.3 Chapter 3 "Methodology"

This chapter is discussing about what development method (waterfall, waterfall with prototyping …) we are using. Technique that used to find resources also discussed here.

1.5.4 Chapter 4 "System Analysis"

This chapter is shows the details of the system requirements including functional and non-functional requirements, hardware and software requirements, and also the reasons for choosing the selected programming language and middleware.

1.5.5 Chapter 5 "System Design"

This chapter is shows the details of the system development including system architecture, design, database design, and interfaces design.

1.5.6 Chapter 6 "Implementation"

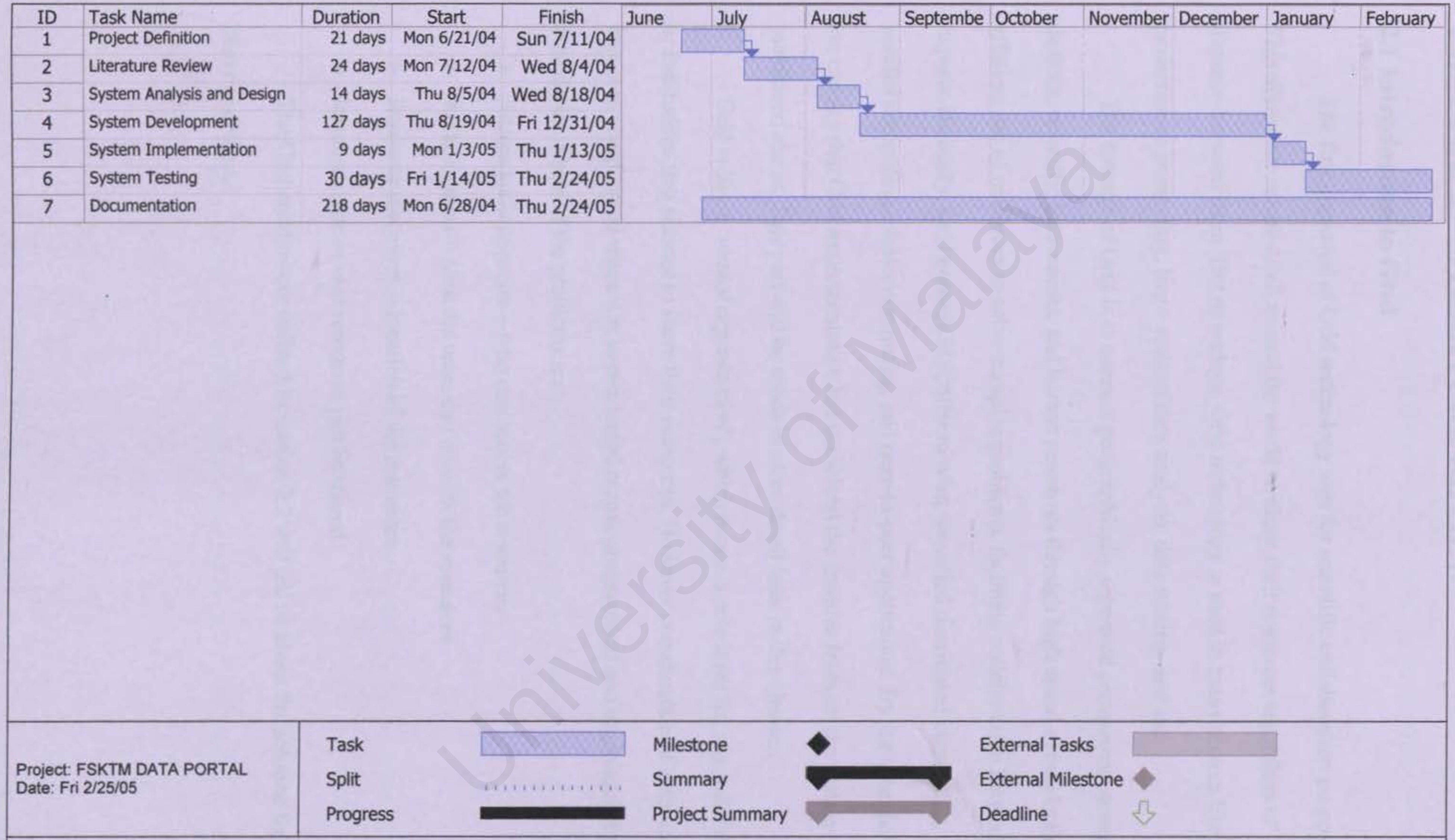
This chapter focuses mainly on the implementation phase of the entire project which consists of the coding process done to convert the proposed project into a fully functional system.

1.5.7 Chapter 7 "System Testing"

This chapter describes the testing process and methods carried out to verify and validate the system to make sure it fulfills its requirements. It is an important phase in determining the errors, bugs and faults of the system and the subsequent action taken to overcome it.

1.5.8 Chapter 8 "System Evaluation"

This chapter will touch on the evaluation done on the finished system. It will include the problems encountered during the system development, the system's weaknesses and strengths, etc.



CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction to Grid

The first approved of Grid technology was for scientific collaboration purpose. This allows the scientists all around the world to share their resources regardless of distance between them. But nowadays, Grid technology is used in many sectors like high-performance computing, large-scale of data analysis, data mining, and etc.

The concept of Grid is to connect geographically separated computers, storage devices, scientific instruments, and human resources through high speed networks to efficient use of resources to solve complex problems, facilitate collaboration between experts. Basically, Grid concept is similar to what we called distributed computing, parallel computing, cluster computing, and peer-to-peer application. By the other hand, we can say that Grid implementation has combined the features from all the system mentioned above. This part will be touch in more detail later in this chapter.

Grid is like a '*virtual organization*', which means a rule is set for any individuals or institutions that attempt to share their resources. However coordination of resources becomes complicated when it is implemented across geographical and organizational boundaries. Some of the problems are:

- Access authentication – who can access the resources
- Authorization – what the users can do with the resources
- Resource discovery – location of the resources
- Resource type – what resources can be shared

The Grid architecture defined in section 2.2 will tell us about the solution for the above problems.

Recently, there are 2 main types of Grid computing: Computational and Data Grid. The following table shows the explanation for the 2 types of Grid and also the differences between them.

Computational Grid	***Data Grid***
Focused on processing capability	More concerned on network bandwidth
To handles large computing processes	Provides an efficiency management for large scale of data
Processes are executed through distributed resources	Data can be accessed from geographically dispersed sites.
More useful when many raw data to be reduced and their processing is shared between sites	More useful when reduced data already exist and needed to be accessed by may users through remote sites

Table 2.1: Computational Grid vs Data Grid

However, there is one common between them: No need to know resources location.

- Computational Grid: users no need to know where the processes are performed.
- Data Grid: while searching for data, users no need to select which database or sites to search.

Here are some extra details about the Data Grid. The main purpose of Data Grid is to deals with efficient management, placement and replication of large collections of data. Characteristics of Data Grid included:

- Heterogeneity Transparency

  Access mechanism must be independent of the actual implementation of the data source
- Location and Name Transparency

  Application can access data without knowing its actual location

- Distributed Transparency

  Application should be able to manage, query or update the distributed data in a unique implementation

- Replication Transparency

  Data can be replicated to many sites for performance and availability. Application allows data accessed from the most suitable replica.

2.2 Grid Architecture

Before go through the Grid architecture, let take a look on the Grid's characteristics.

- *Coordination of resources without centralized control.*

  Grid allows a secure, flexible and coordinated resource sharing among dynamic collections of individuals, institutions, and resources. Everyone is act like an administrator that maintains his own resources in their desktop.

- *Using standard, open, general purpose protocols and interfaces.*

  A Grid is formed using multi-purpose protocols and interfaces that define different functions like resource discovery, resource access, authentication, authorization, and so on. A standard and open of these protocols or interfaces should be decided.

- *To deliver nontrivial qualities of service.*

  A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts.

- *Interoperability on heterogeneous system*

  Sharing relationships is one of the important considerations in Grid design. Sharing relationships can be built among existing system or new developed system especially with different platforms, language, and programming environment. So with the interoperability, different type of system can communicate and share different type of resources.

Due to the Grid is implemented across the network, its architecture can be divided into several layers like the TCP/IP or OSI model.

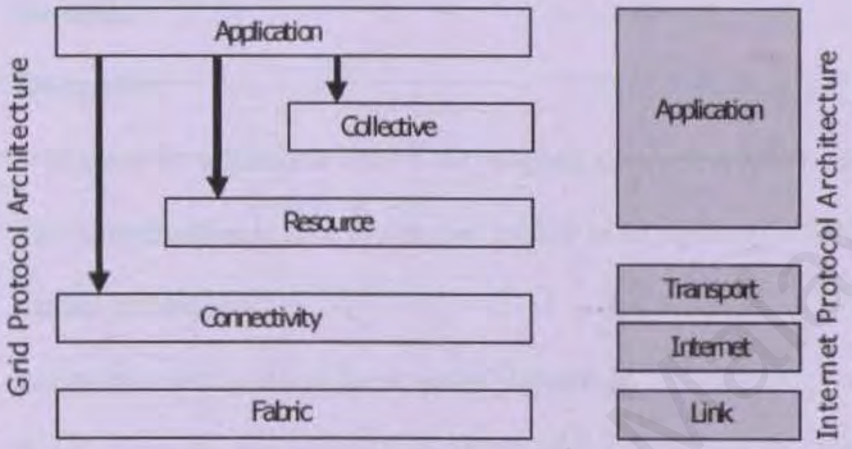


Figure 2.1: The layered Grid architecture and its relationship to the Internet protocol model.

2.2.1 Fabric layer

The Fabric layer contains the Grid protocols and interfaces that provide access to the shared resources. Fabric components will perform a sharing operation on specific resources. Enhance the Fabric functionality enables higher-level of sharing operations.

2.2.2 Connectivity layer

The connectivity layer defines core protocols required for Grid-specific network transactions. There are 2 core protocols: communication and authentication protocol. Communication protocols used to manage the exchange of data in Fabric layer, when the authentication protocols provide uniform of security mechanisms to identify the users and resources accessibility.

Characteristics of the authentication methods for virtual organization are:

- *Single log in*

  Users only need to sign in once and then they allowed accessing multiple grid resources

- *Delegation*

  User must be authorized before the program can access to the required resources. The program should able to delegate partial of its rights to other program under certain conditions.

- *Integration with various local security methods*

  Every resources owner may have their local security tool like Kerberos and Unix security. So, the Grid security must able to interoperate with the local security solutions.

- *User-based trusted relationships*

  Users should be allowed to access resources from multiple providers without requiring each providers' security interact.

Examples of protocol:

- *Public-key based Grid Security Infrastructure (GSI)*

  Used for authentication, authorization, and communication protection. It provides single login protection capability. Public-key technology X.509 is used as the formal identity certificates.

2.2.3 Resource layer

The Resource layer defines protocols required to access and control the sharing of individual or local resources. These protocols only concentrate in the local resources and ignore the controlling over global network sharing.

There are 2 main classes of Resources layer protocols:

- *Information protocols*

  These protocols are used to get information about the state and structure of a resource.

- *Management protocols*

  These protocols are used to control access to the shared resources and the operations performed, such as data access or process access.

Examples of protocol:

- *Grid Resource Information Protocol (GRIP)*

  A protocol based on the Light Weight Directory Protocol (LDAP) that used to control access to structure and state information.

- *Grid Resource Access and Management Protocol (GRAM)*

  A HTTP-based protocol used for remote allocation of resources and for monitoring and controlling the use of resources.

- *Grid File Transfer Protocol (GridFTP)*

  A protocol builds on and extends the File Transfer Protocol for high-speed data access.

2.2.4 Collective layer

The Collective layer defines protocols required to control the interaction of resources across world wide. This layer protocols provide a meta-directory services that allow users to view the resources, request for resources, monitoring of resources failure, and etc.

2.2.5 Application layer

The final layer, Application layer defines the user applications that run at on the top of Grid computing environment.

2.3 Types of Grid Application

Grid applications include:

a) **Science Portal**

Science Portal is a problem-solving environment that enables users to access and execute Grid application by provides a convenient remote web browser interface. With this application, scientists can efficiency solve their problem through the Grid without install some new software. Until now these portals are implemented in physic, chemistry, biology, mathematic, and other sectors.

b) **Distributed Supercomputing**

Like the word 'distributed', this application consists of multiple resources. Within a Grid, dispersed resources over the world can be combined through high-speed network to provide a high-performance computing capability. For example, the Entropia Inc's FightAIDSAtHome system use more than 30000 computers to analyze AIDS drug candidates.

c) **Data-intensive Application**

This type of application focused on analyzing of large amount of data. The distributed environment of Grid can provides impressive storage spaces for holding the intermediate results. Parallel mechanisms in Grid analyzing processes also ensure the use of distributed resources efficiently.

d) Collaborative Work

High computing power is not the only function for researchers but they also want to share their works with other community. They can help each other to carry on a testing for results on the same applications or simulations and in the same time discuss about encountered problems.

e) High-Throughput Application

A large number of independent tasks can be scheduled in Grid and performed by any available unused resources. For example, RSA keycracking is one of the high-throughput applications.

2.4 Another Computing Technology

There are many computing architectures other than Grid. For examples, parallel computing, serial computing, distributed computing, clustering, peer-to-peer application, high-performance computing, and others. Some of them are connected to the Grid architecture because they have concept in common. Here we will discuss the about the 5 similar concepts.

2.4.1 Parallel Computing

In Parallel computing environment, it simultaneously uses of multiple compute resources to solve a computational problem. The multiple compute resources can be:

- Single computer with multiple processors.
- Multiple computers connected through network.
- Combination of both.

For parallel computing, instructions are executed concurrently within 1 clock cycle. There are 2 main type of instructions execution:

- Single Instruction, Multiple Data (SIMD)

  1 same instruction is being performed by multiple resources in 1 clock cycle, but each instruction can executes different data. This is a synchronized and deterministic process.

- Multiple Instruction, Multiple Data (MIMD)

  Multiple different instructions are performed by multiple resources in 1 clock cycle. Each instruction also can execute different data streams. This process can be synchronous or asynchronous.

2.4.2 Distributed Computing

Distributed system is formed by a group of interconnected cooperating computers. The algorithm of distributed computing is processing a set of data by allocating the data among the cooperating computers. Distributed computing is developed to replaced supercomputer due to its same power of processing ability but cheaper in cost. Common software has to be installed on each personal computer to handle the jobs processing in a distributed environment. The software will define what to be processed and how to divide the data among the computers.

2.4.3 Clustering System

Characteristics of clustering system:

- Formed by multiple same platform computers (homogeneous).
- Use shared resources.
- All computers are connected through network.
- All computers are trust each other and therefore no password required.
- Use same software so that application can run on every component.

All computers in cluster system must operate using the same platform (PC/Unix). The size for clustering systems is always smaller as only consist fewer than 100 computers. The cause is the interconnection and programs for cluster are less mature.

A general use of clustering system is for load-balancing purpose. The process of load-balancing is partition the jobs of a computer to others computer so that more jobs can be done in the same amount of time. The goal of this application is can serve users' request faster.

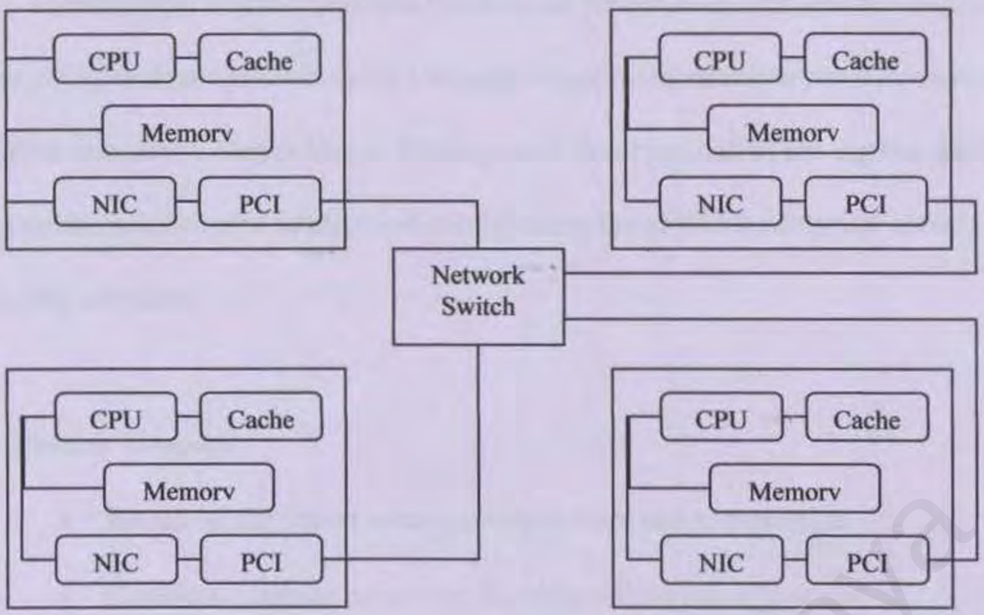


Figure 2.2: Structure for 4 nodes cluster

2.4.4 High-Performance Computing

High-performance computing is a technology developed to solve computational problem that need powerful processing capability, save time, and process large scale of data in once. It is also called supercomputer. With the power of supercomputer, researchers can reduce the time required to solve a complicated computational problem. There are 2 types of high-performance application:

- Serial type
- Parallel type

Serial type high-performance application uses only 1 processor to execute multiple instructions in sequence. This is the traditional computing application method and hence became difficult to speed up the performance due to serially instructions execution.

Parallel type is using multiple numbers of processor to perform related tasks, neither using multiple processors in 1 computer nor using multiple processors across separated computers. Depending to the degree of communication among the application components, parallel type application can be categorized into 2 subtypes: loosely coupled and tightly coupled.

Loosely Coupled

- Required minimum communication between components.
- Contains a master processor that controlling other processors' (slave/workers) processing processes.
- Slaves perform tasks based on master request. Slaves will inform the master when finish processes the tasks.
- Master will then collect the output results from the slaves.
- Master and slave are connected by network.

Tightly Coupled

- Processing components communicate with each other frequently.
- Consists of 2 kinds of system architecture: scale-up and scale-out
- Scale-up system can be extended by adding more processors and memory units to provide more processing power when load increases. This system uses a shared memory for communication of data.

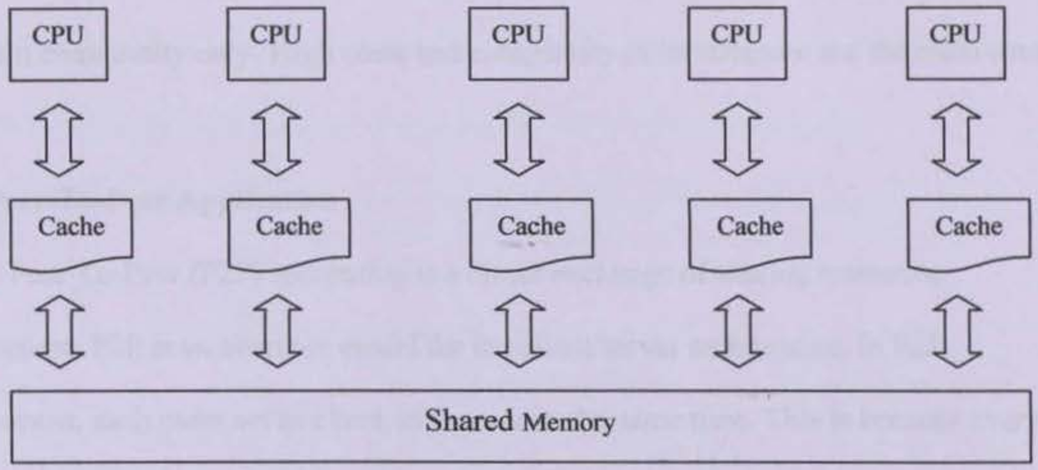


Figure 2.3: The architecture for scale-up system.

- Scale-out system is a set of computers connected through network that enable an application to be scaled by divided the processes and running it separately by the set of computers. When the load increases, additional computers can be added to the set and provide more processing power and memory.

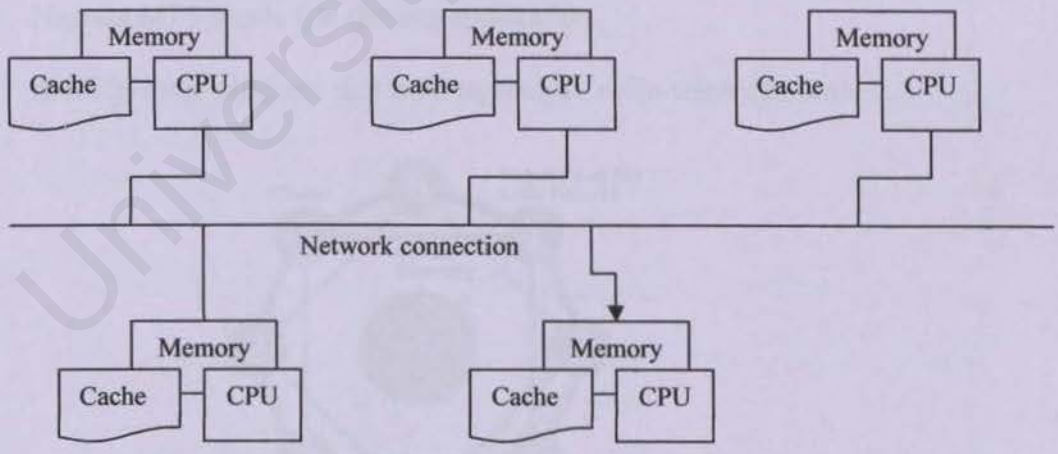


Figure 2.4: The architecture for scale-out system

However high-performance computing technology is not widely used but limited to certain community only. High costs and complexity of its structure are the main causes.

2.4.5 Peer-To-Peer Application

Peer-To-Peer (P2P) computing is a direct exchange of sharing resources environment. P2P is an alternate model for the client/server architecture. In P2P environment, each users act as client and server in the same time. This is because every peer (user in P2P environment is called peer) can request access to other peers' resources and in the same time shares their own resources to others. P2P is like communities where everyone involved can share resources, communicate, and collaborate. So the 2 main features of P2P are:

- Users act as client and server.
- Sharing of resources by direct exchange.

Examples of P2P computing:

- Napster MP3 music file sharing application.
- SETI@Home program that used to analyze radio telescope data.

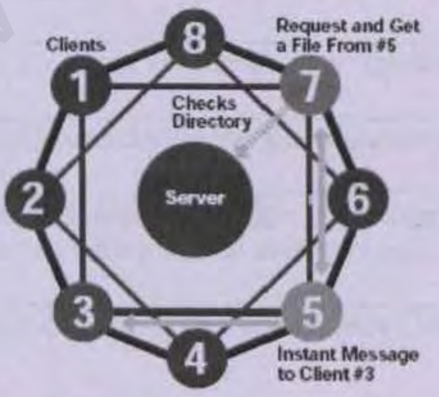


Figure 2.5: P2P Model

2.4.6 Comparison between Grid with the other computing system

Actually Grid application has combined all the features of the computing models listed above. The following are the features from every computing model:

Distributed Computing / Cluster	Compute processes in Grid are run by disperse resources to speed-up the operations. Now, we can define distributed computing as subset of Grid.
High-Performance Computing	One of the purposes of Grid is provide powerful computing capability for solving complicated scientific problems.
Peer-to-Peer	Grid also focused on resources sharing.

Table 2.2: Grid features from other computing system

However there are some differences between them.

Grid	Implemented through cross platforms (heterogeneous).
Cluster	Consists of same platform of operating components (homogeneous).

Table 2.3: Grid vs Cluster Computing

Grid	Focus on resources coordination. Information searched through Grid is resources from other individually accessible computers.
Peer-To-Peer Application	Focus on direct exchange of data sources between 2 nodes connected in P2P environment.

Table 2.4: Grid vs Peer-to-Peer Application

Grid	Can be implemented over the global network connection.
High-Performance Computing	Always implemented by a single supercomputer or a set of resources interconnected within a local area network.

Table 2.5: Grid vs High-Performance Computing

2.5 CASE STUDY

2.5.1 Case Study 1: CCLRC Data Portal

This is a project conducted under the e-Science Center, UK. There are temporally 4 faculty involved in this project for the Central Laboratory of Research Councils. Each faculty is have their own scientific data generated, so the aim of this project is to develop the means for a scientist to explore these data resources, discover and retrieve the data through one standard interface and independent of the data location.

Three challenges have been proposed in this project: data accessibility, data transfer and management of personal data. Data accessibility implies the capability to locate information without prior knowledge of its physical location or the form in which its contents is described. Data transfer relates to the problem of large scale of data need to be transferred across the Grid environment. Management of personal data is related to the growing of distributed data produced by scientists within the Grid environment.

Architecture of CCLRC Data Portal

The CCLRC Data Portal consists of three main components:

- A web-based user interface including a security subsystem.
- A metadata catalogue
- Data resource interfaces

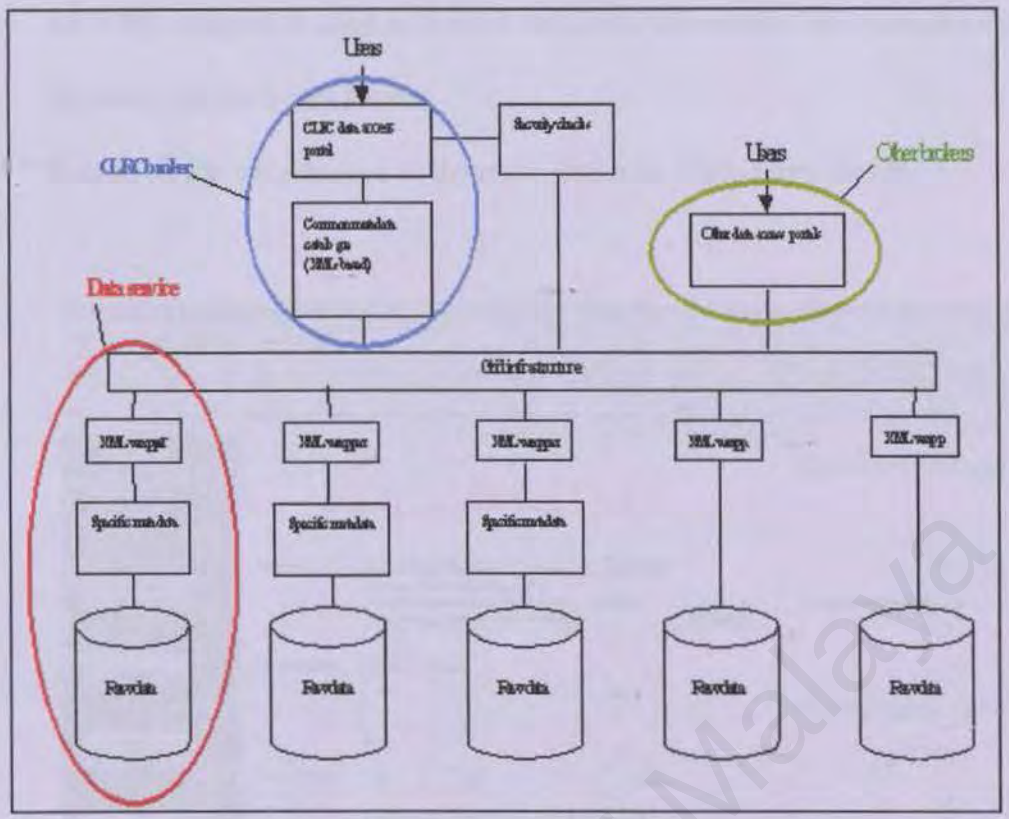


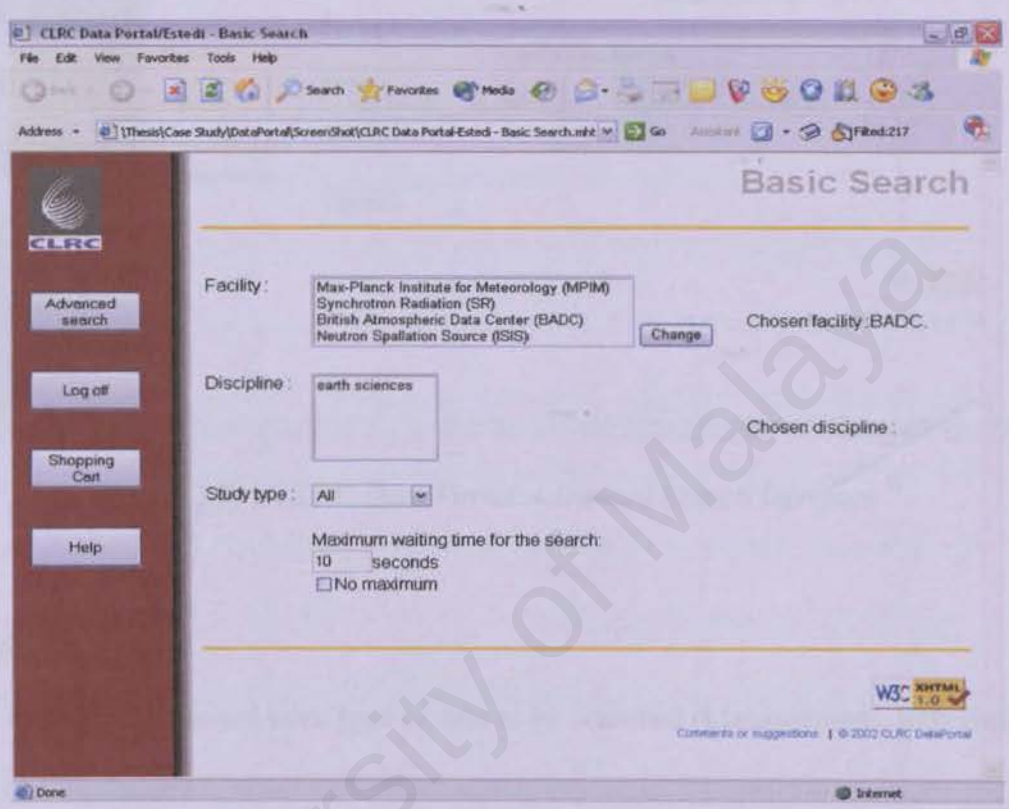
Figure 2.6: Architecture of CCLRC Data Portal

The communication between each databases or faculty is via SOAP for sending message. Web services technology and SOAP enable the Portal to be decentralized into modules that represent an area of functionality. These services also are platform and language independent. Other services can easily be integrated into this Portal in the future.

How CCLRC Data Portal work

- User selects what kind of data to be search.
- Request from users will be interpreted by local server and a query is formed.
- The query will be transmitted to the Central Metadata Catalogue or connected Metadata Repositories in XML form.

- An XML wrapper is used to convert the query statements into corresponding database unique query format.
- Search results are returned to the main server in XML form via http.



CLRC Data Portal/Estedi - Basic Search
File Edit View Favorites Tools Help
Search Favorites Media
Address \Thesis\Case Study\DataPortal\ScreenShot\CLRC Data Portal-Estedi - Basic Search.mht
Go
Filtered:217
CLRC
Advanced search
Log off
Shopping Cart
Help
Change
Comments or suggestions | © 2002 CLRC DataPortal
Done
Internet

Basic Search

Facility: Max-Planck Institute for Meteorology (MPIM) / Synchrotron Radiation (SR) / British Atmospheric Data Center (BADC) / Neutron Spallation Source (ISIS)

Chosen facility :BADC.

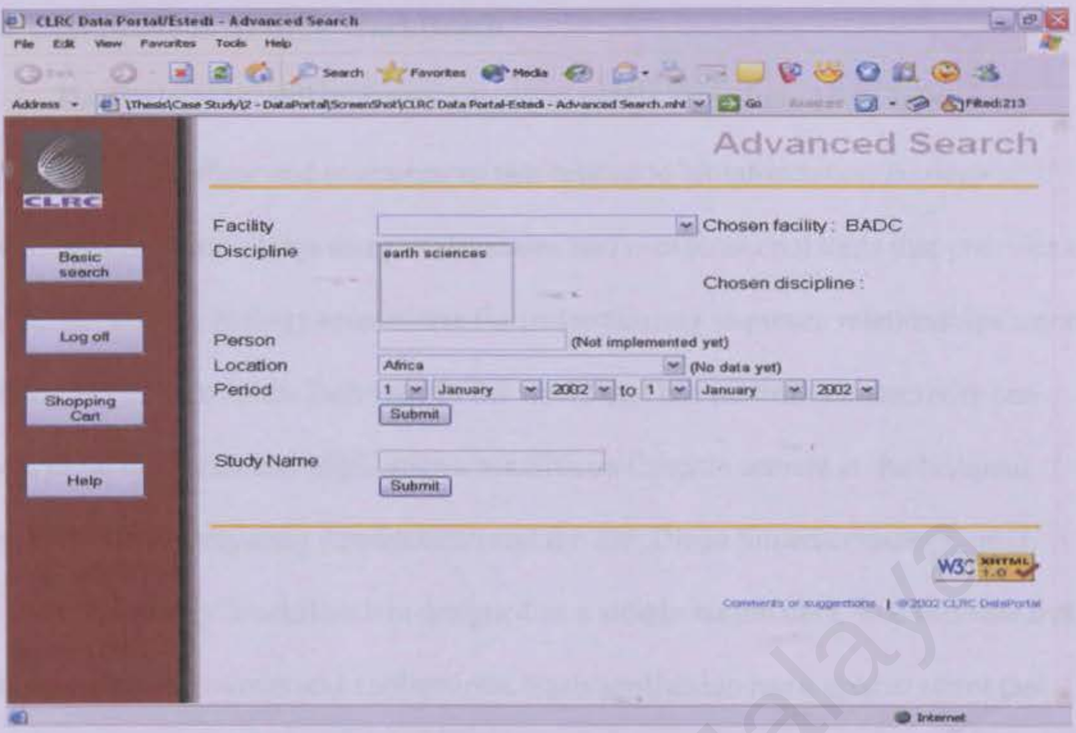
Discipline : earth sciences

Chosen discipline :

Study type : All

Maximum waiting time for the search:
10 seconds
□ No maximum

Figure 2.7: CCLRC Data Portal Basic Search Interface


CLRC Data Portal/Estedi - Advanced Search
File Edit View Favorites Tools Help
Search Favorites Media
Address
\Thesis\Case Study\2 - DataPortal\ScreenShot\CLRC Data Portal-Estedi - Advanced Search.mht
Go
Filted:213
CLRC
Basic search
Log off
Shopping Cart
Help
Advanced Search

Facility Chosen facility : BADC

Discipline: earth sciences

Chosen discipline :

Person (Not implemented yet)

Location: Africa (No data yet)

Period: 1 January 2002 to 1 January 2002

Submit

Study Name

Submit

W3C XHTML 1.0
Comments or suggestions | ©2002 CLRC DataPortal
Internet

Figure 2.8: CCLRC Data Portal Advanced Search Interface

Strengthens:

- Allow user to select what type of data to be searched (Measurement, Experiment or Simulation)
- Allow user to set maximum search time
- Allow user to specify the which period of data to be searched

Weaknesses:

- No search keyword

2.5.2 Case Study 2: Biology WorkBench

The Biology WorkBench was opened to public since June 1996. It is a computational interface and environment that related to bioinformatics. Biology WorkBench contains a large array of databases and computational tools that provides a convenient way for biology researchers for understanding sequence relationships among proteins and nucleic acids. Everyone in the world that has network connectivity can access these databases and applications, via Silicon Graphic servers at the National Center for Supercomputing Applications and the San Diego Supercomputer Center.

The Biology WorkBench is designed as a simple button click web browser that links to variety of sources and applications. Each application has a unique script that control connection to the users' required sources. Functionally the scripts change the interface into a standard web page that permits them to be seamlessly interconnected. The scripts also work closely with the interfaces ensure that all operations can be done as fast as possible. Users can easily use the services without knowing details about the structures behind due to the simplified into point and click web interfaces.

Important features of Biology WorkBench:

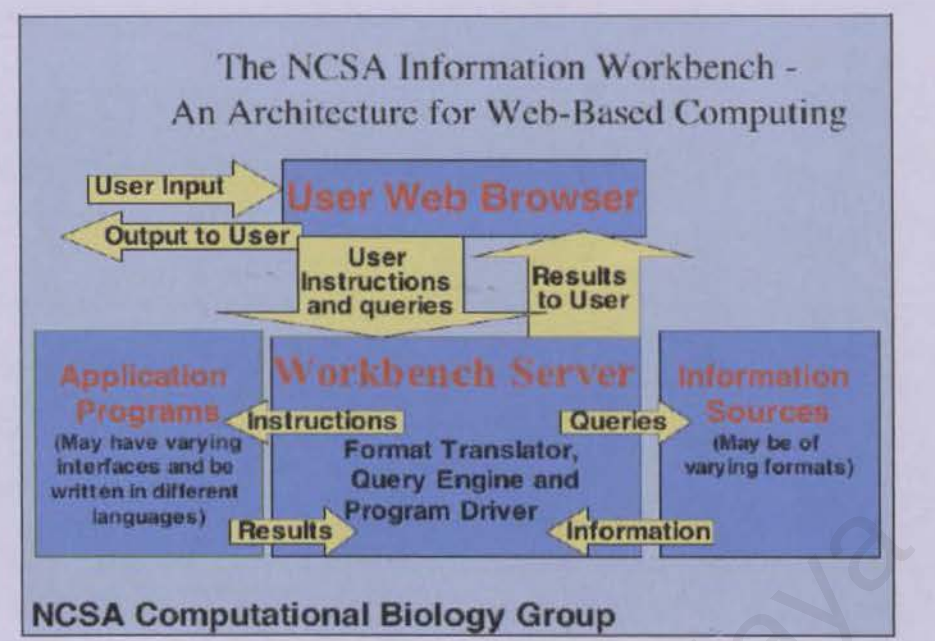
- Provides a unified interface to access various analysis tools and data sources.
- Can be accessed with a standard web browser.
- Free accessible by anyone.

Among the variety of services provided in this WorkBench, one of the services is called Ndjinn-Multiple Databases Search. This service allows users to search for related protein sequence from multiple dispersed databases. There are more than 50 databases available.

How Biology WorkBench work

Many programs and data sources available through Biology WorkBench are public domain resources from separated places. Biology WorkBench acts as a database manager that collects all the distributed information and provides a common web interface to access them. Its interface somehow is on the NCSA's SGI POWER CHALLENGEarray. The operation behind is that the interface has a translation libraries that convert a generic query language into the query language unique to every database. Besides, each programs or database sources have a script inside them that will convert them from 1 format to another.

The databases within the Biology WorkBench are always up to data because it will send an intelligent agent to search the mirror sites on the internet for any updating information every night. It also has been considered about the overloading of the servers that they use Java as one possible solution. On the other hand, massive of upgrade to the databases and tools is another problem. In here, the developers of Biology WorkBench use Perl5 (an object-oriented programming language that easy to upgrade) to write the communications gateway.

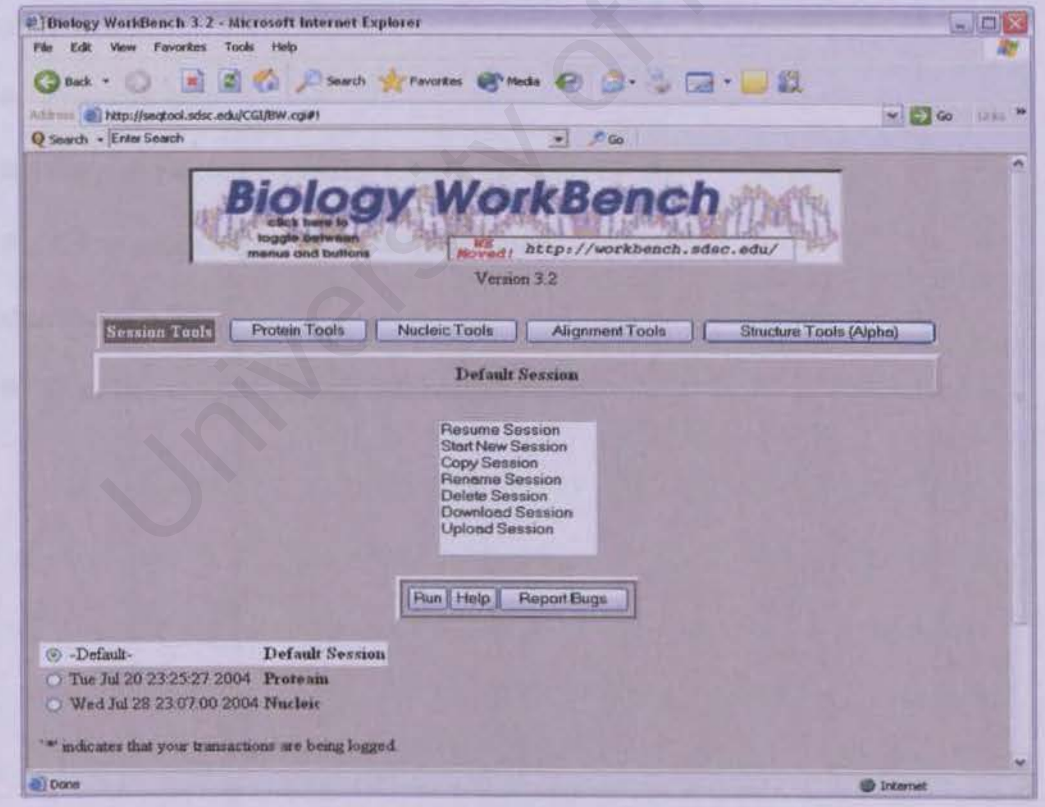


The NCSA Information Workbench -
An Architecture for Web-Based Computing

User Input
Output to User
User Web Browser
User Instructions and queries
Results to User
Application Programs
(May have varying interfaces and be written in different languages)
Instructions
Results
Workbench Server
Format Translator, Query Engine and Program Driver
Queries
Information
Information Sources
(May be of varying formats)

NCSA Computational Biology Group

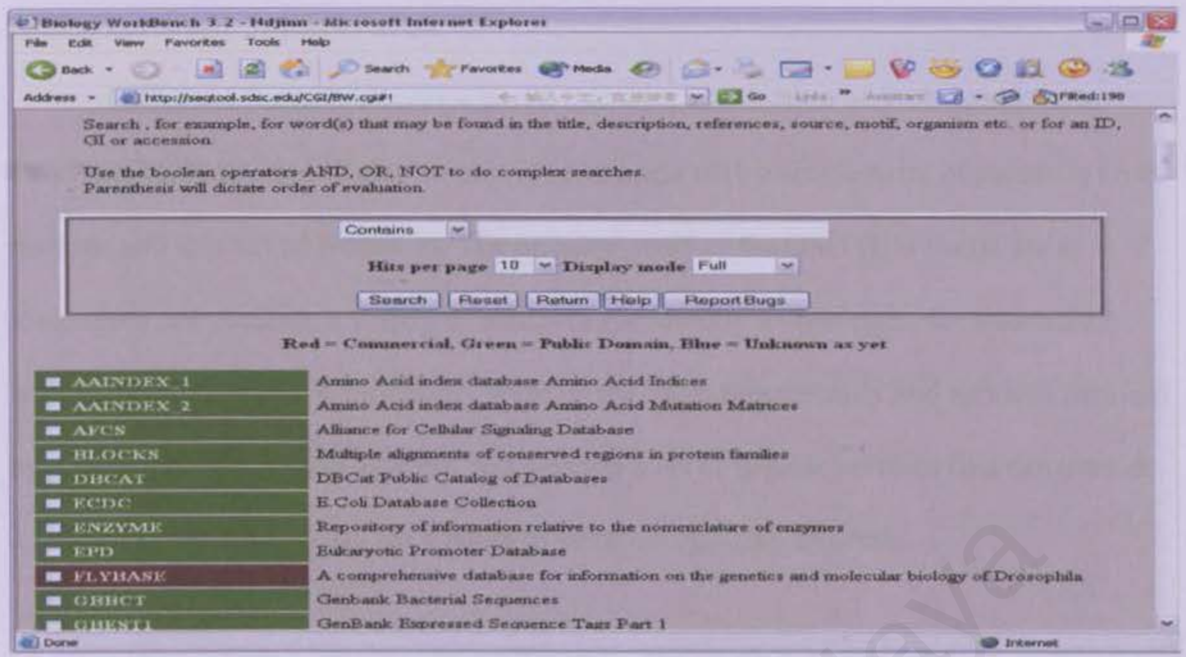
Figure 2.9: the workflows for the operations of WorkBench.



● -Default-	Default Session
○ Tue Jul 20 23:25:27 2004	Proteain
○ Wed Jul 28 23:07:00 2004	Nucleic

'*' indicates that your transactions are being logged

Figure 2.10: Application Interface for WorkBech and the available tools


Biology WorkBench 3.2 - Ndjinn - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://seqtool.sdsc.edu/CGI/BW.cgi#1 Go
Contains
Search Reset Return Help Report Bugs
Done
Internet

Search , for example, for word(s) that may be found in the title, description, references, source, motif, organism etc. or for an ID, GI or accession.

Use the boolean operators AND, OR, NOT to do complex searches.
Parenthesis will dictate order of evaluation.

Hits per page 10 Display mode Full

Red = Commercial, Green = Public Domain, Blue = Unknown as yet

AAINDEX_1	Amino Acid index database Amino Acid Indices
AAINDEX_2	Amino Acid index database Amino Acid Mutation Matrices
AFCS	Alliance for Cellular Signaling Database
BLOCKS	Multiple alignments of conserved regions in protein families
DBCAT	DBCat Public Catalog of Databases
ECDC	E.Coli Database Collection
ENZYME	Repository of information relative to the nomenclature of enzymes
EPD	Eukaryotic Promoter Database
FLYBASE	A comprehensive database for information on the genetics and molecular biology of Drosophila
GBBCT	Genbank Bacterial Sequences
GBEST1	GenBank Expressed Sequence Tags Part 1

Figure 2.11: Ndjinn-Multiple Databases Search interface.

Strengthens:

- Provides help
- Support Boolean operator (AND, OR, NOT)
- Allow user to specify the display mode

Weaknesses:

- User has to select which database to search

2.5.3 Case Study 3: DSG Portal

DSG Grid Portal is a service portal that uses a collection of Web-based tools, GridPort, MyProxy and Globus to provide end users with a single point of access to Grid services and networked resources. The primary goals of the DSG Grid Portal are to investigate and evaluate a variety of different commodity technologies for interacting with Grid-based services, to identify some of the basic functionality and services required by a particular application domain, and develop a set of generic services that can provide a portal's core functionality for a range of different application needs.

Architecture of DSG Portal

The DSG Grid Portal is structured into the classic three-tier architecture. These tiers separate functionality and responsibility. The Web browser serves as the first-tier, which provides a uniform and familiar interface for users to access Grid and other services. The middle-tier, which acts as a service broker, provides the functionality for the first-tier. The services can be implemented directly by the middle tier software, or the middle tier can act as a proxy for accessing backend services. Backend services can be, for example a queuing systems for a high performance computer, a database, a mass storage systems, or meta-computing services such as Globus.

The implementation of the DSG Portal consists of a vertical layer of software and hardware components. The DSG Web server is a secure Apache Web server capable of handling standard HTTP/HTTPS requests from client browsers and producing either statically or dynamically generated HTML pages to clients. The next layer consists of a well-defined set of core services. Currently, the DSG Grid Portal's supported the following core services:

- Security service – allow access only to authenticated users to computational resources.
- Information service – allow users to discover resources and monitor system's queues and usages.
- Resource monitoring service – allow users to discover and monitor Grid sites as well as their resources.
- Desktop service – allow users to interact through a single and familiar interface (their Web browser) with a desktop environment.
- Database services – allow users to store, cache and retrieve data via standard relational database systems.

The core services are implemented using a combination of Java applet's, JavaScript, PHP, Perl-based CGI scripts and a single instance of GridPort toolkit. GridPort is in turn implemented using Grid enabled software modules such as Grid Security Infrastructure (GSI), GSI-FTP, Globus Resource Allocation Manager (GRAM), Grid Information Services (GIS) and MyProxy. Since these software modules constitute the major part in the design and the development of the DSG Grid Portal, we will briefly describe them in the following sub-sections.

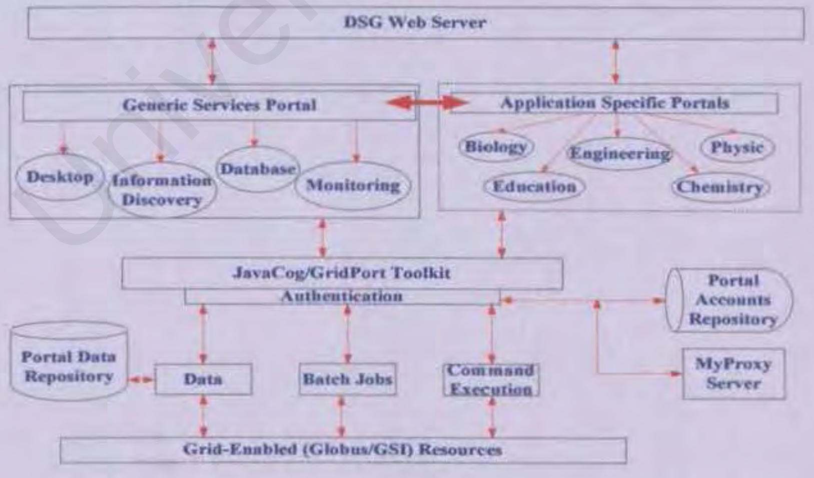
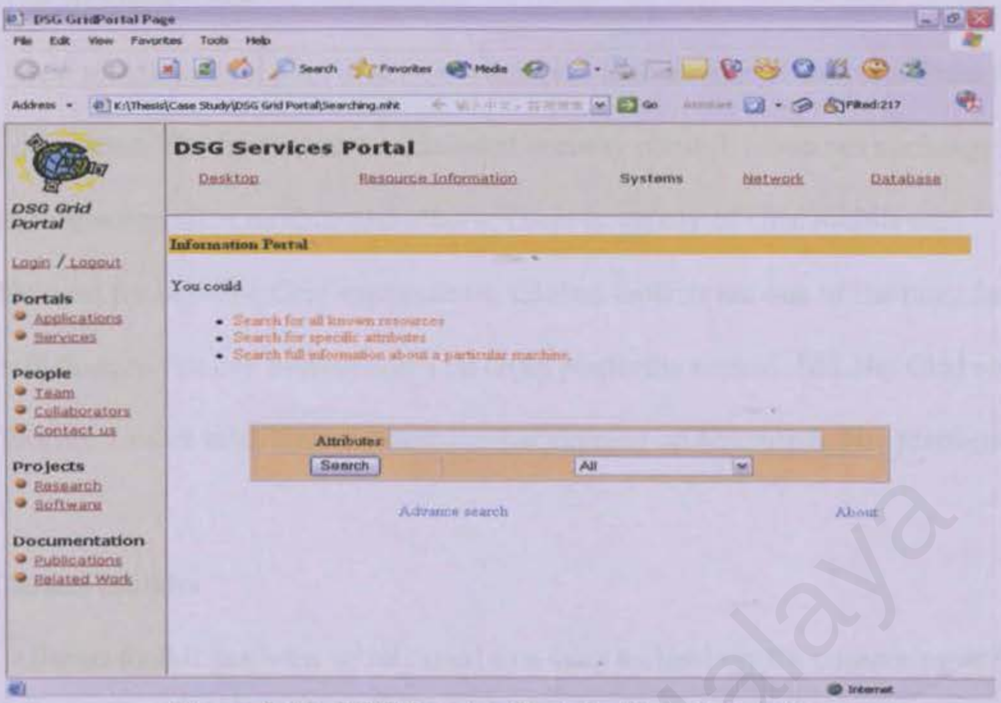


Figure 2.12: Three-tier Architecture of DSG Portal



You could

- Search for all known resources
- Search for specific attributes
- Search full information about a particular machine.

Figure 2.13: DSG Service Portal – Search Interface

Strengthens:

- Easy to use

Weaknesses:

- No additional functions

2.6 Grid Middleware

Grid middleware is a set of services that provides various of functionalities for the Grid applications. The functionalities included security control, resources exchanged process, jobs submission module and others. There is variety of Grid middleware available used for building Grid applications. Globus toolkits are one of the most famous Grids middleware that can implemented on cross platforms system. MS.Net Grid and OGSI.Net are 2 other middleware examples that running on Microsoft .Net platform.

2.6.1 Globus toolkits

Globus toolkit has been widely used as a Grid technology for computing and scientific purpose. It is a community-based and open source set of services that used for building Grid applications. The functions included in Globus toolkit are security, information discovery, communication, data management, fault detection and portability.

The Globus toolkit is developed to be compatible with the Open Grid Services Architecture (OGSA). The components included in toolkit are Grid Resource Allocation and Management (GRAM), Meta Directory Service (MDS-2), GridFTP and Grid Security Infrastructure (GSI).

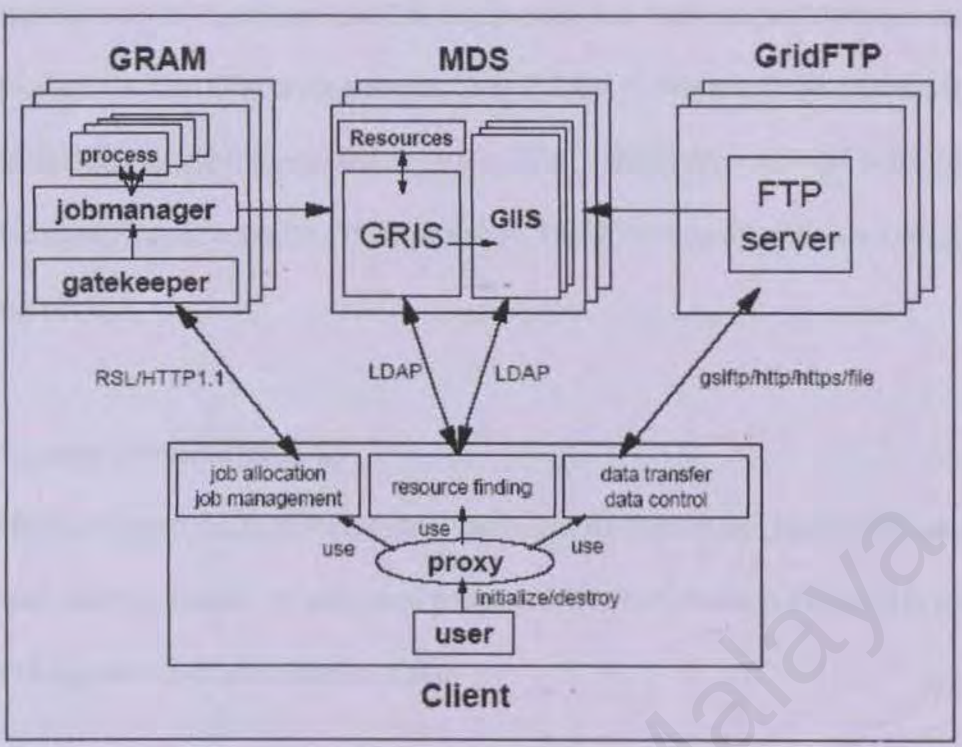


Figure 2.14: The System Overview of Globus toolkits

Grid Resource Allocation and Management (GRAM)

This protocol provides the secure remote execution and management of the execution. GSI protocol is used to authenticate, authorize and delegate credential for access remote Grid application. Jobs submitted by users are processed by the 'gatekeeper' and it responsible for begin and monitoring the job. Status of the jobs will be send back to users and the jobs are terminated when finished.

Grid Security Infrastructure (GSI)

- GSI is a public-key based security mechanisms. It provides single log in secure authentication, access restricted control and communication security. Single log in security enables users just need to log in once and then can access the resources available within Grid environment. The GSI use X.509 proxy certificates for controlling single log in and delegation. The authentication of GSI is based on the Transport Layer Security (TLS) protocol. The X.509 certificates are sitting on the top of TLS.

Meta Directory Service (MDS-2)

MDS-2 is used for monitoring the system configuration and status such as server settings and network traffic. A soft-state protocol, Grid Notification Protocol is used for life-time management of resources on Grid.

GridFTP

GridFTP is similar to normal network FTP that provides a secure and reliable data transfer among Grid nodes. The GridFTP is divided into 2 parts: GridFTP server and GridFTP client. There are two types of file transfer supported by GridFTP

- Standard file transfer

  Clients are send the local file to the remote FTP server machine.

- Third-party file transfer

  Clients are copies large files in a remote storage to another remote server.

2.6.2 MS.Net

MS.NET Grid is a collaboration project between EPCC, Microsoft Research Limited and the National e-Science Centre (NeSC). It is implementing Microsoft .Net web services on the Open Grid Services Infrastructure (OGS1). Globus Toolkit 3 implementation of OGSI and the design of OGSI .Net inspiring the development of MS.NET Grid. The development team gain features in these two toolkits to shorten the development time. ASP.Net is used to intercept and serialize incoming HTTP/SOAP messages and mimicking the service deployment model.

Architecture of MS.Net Grid

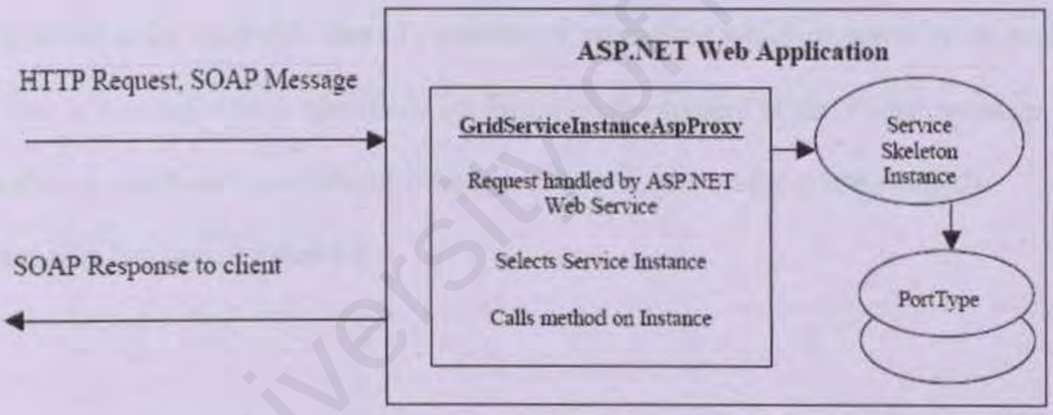

HTTP Request, SOAP Message
SOAP Response to client
ASP.NET Web Application
GridServiceInstanceAspProxy
Request handled by ASP.NET Web Service
Selects Service Instance
Calls method on Instance
Service Skeleton Instance
PortType

Figure 2.15: Architecture of MS.NET Grid

The OGSI container is implemented as an ASP.NET web application. Components deal with service instance management, managing communication with clients, providing OGSI portType-related functionality and allowing developers to deploy services within this application. The architecture of MS.NET Grid is shown in figure 4.5 above.

Service Lifetime

There are two types of service lifetime in MS.NET Grid. They are 'transient' service lifetime and 'persistent' service lifetime. The 'transient' service time applies to the service instances which are spawned by other services. This type of service instance will be terminated when termination time has passed. It do not respond to operation invocation requests. 'Persistent' service lifetime is the service instances are initialized when the container starts up. The service instance lives as long as the containing web application.

Maintaining State

SOAP message is sending to a network endpoint at somewhere. The message is interpreted at the endpoint. One of a number of operations which exposed by the web service is invoked. These operations are based on the content of the SOAP message. The results are serialized into a SOAP message and send back to the client when the processing has been carried out.

2.6.3 OGSI.Net

OGSI .Net is a grid middleware which operate on Microsoft .Net framework. There two highest design goal in developing OGSI .Net are to support the dynamic creation of grid service instances that persists between client invocations and to use IIS to receive requests from clients.

The container entity is created to hold all the services instances which running on a host. A collection of ApplicationDomains or AppDomains and Microsoft's mechanism for intra-process memory protection are the modules of the container entity. Each service instances execute on its AppDomain. An additional domain is added for container's logic.

Users can submit jobs on OGSI .Net architecture by online. IIS web server is ready for receiving messages which sent by users. OGSI .Net uses an ISAPI filter to intercept request. The purpose of using ISAPI filter is to support arbitrary names of grid services. It only intercept request at an early stage in the IIS request chain.

Components of OGSI.Net

There are several components in the OGSI .Net to implement a grid application. The dispatcher, the service wrappers, factories and message handlers are the major components. The overview of components of OGSI .Net is shown in figure 4.4.

Dispatcher

Dispatcher is the interface between the client request and the service instances the request. The main function of dispatcher is to route request message to the appropriate service instance. Besides that, it also needs to return the result to the client.

Grid Service Wrapper (GSW)

The various functional units of the grid services instances are encapsulated by grid service wrapper. Each of the AppDomain container has a GSW which only wraps a grid service instance.

Light-Weight Wrapper

Light-weight wrapper depends on the functionality of the container process in order to handle requests and dynamically create or destroy service instances. It allows entities with the limited functionality to present OGSI-compliant interfaces. These entities do not need full container underneath them. Light-weight service wrappers (LWSW) are used to encapsulate these services. There are some restrictions of LWSW:

- Cannot create other services
- Have no SDEs
- Services are terminated when client terminates
- Do not have the configurability of grid services

Factories

Factories create instances of other services in OGSI .Net. A new AppDomain and a new Grid Service Wrapper are created in that domain. A factory stores a reference to the GSW along with the published name of that instance. The mapping is also sent to the dispatcher.

Message Handles

Message handlers perform message format specific processing on a service instance's incoming and outgoing messages. There are two types of message handlers. One of the message handlers is for SOAP and another is for remoting messages format.

A message handler deserializes the request message which arrives from the dispatcher. Creating any parameter objects and processing any message headers are perform in this stage. When the request is completed, the results will be serialized by message handlers. Messages handlers pass the results to the dispatcher to be returned to the client. Service authors are allowed to write services independent of messaging issues.

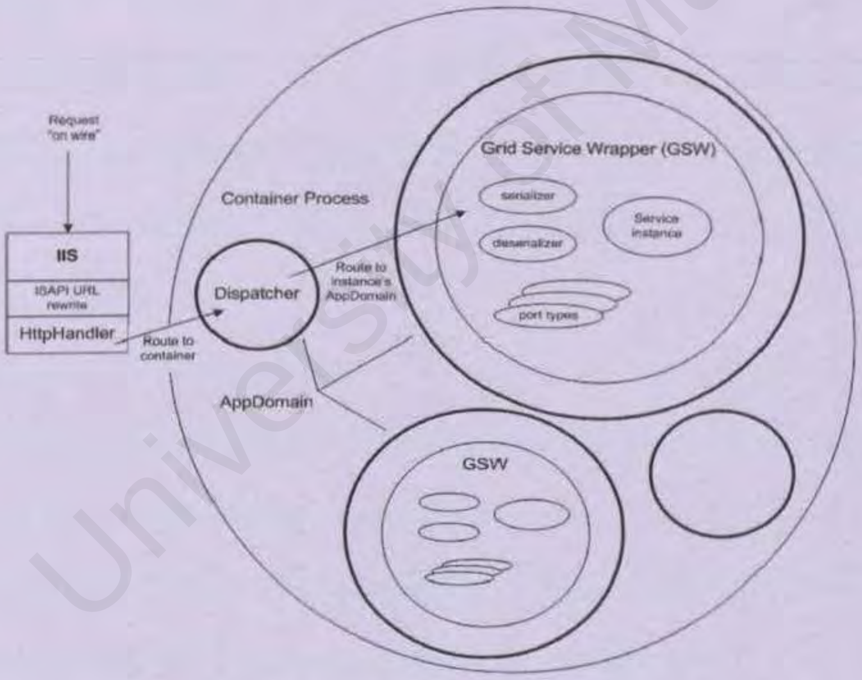


Figure 2.16: Component of OGSI .Net

Security

A standard-based message layer security is provided to service instances via the Web Service Extensions (WSE) pipeline run by message handlers in the Grid Service Wrapper. Besides that, OGSI .Net allows each service instance to live in its AppDomain. This will provide protection for the other services in the container.

2.7 Operating Systems

Windows 98

The first version of Windows 98 was released at the end of year 1996 as the ultimate upgrade to 95 (DOS-based Windows system). The Windows 98 was developed under the consideration of big influence from web technology to the current system. So the Windows is transformed into a web-based system that has Internet Explorer 4.0 integrated within.

The Windows 98 was designed to overcome some inexperienced problems in Windows 95 when upgrading or improve the system. There are 4 main improvements for Windows 98:

- *Provide better Internet access*

  The Internet Explorer 4.0 integrated within Windows 98 provides a convenient way for users to search the internet. Implementation of Active Desktop adds a HTML layer over the Windows desktop that enables live information through internet access.

- *Enhanced computing environment*

  Windows 98 is more stable and faster than Windows 95. Programs are also running faster under Windows 98 computing environment. A new file system (32-bits file allocation table FAT) introduced in Windows 98 adds more disk space that support up to 9 monitors and reduce shut-down time.

- *Making computing more entertaining*

  Improvement to the multimedia included latest version of Direct X, DVD support and MMX support.

- *Compatible with many hardware accessories*

  Windows 98 is support many newest hardware accessories included Universal Serial Bus (USB), TV tuner, 3D sound card, scanners and digital cameras.

Windowss 2000 Pro

The 2000 Pro is the business mobile and desktop operating system that developed based on the NT technology. This Windows is designed specified for business computing usages instead for home users. The hardware requirement for installation also higher due to it is used for commercial purpose.

Compare to Windows 98 and NT, Windows 2000 Pro is more stable and reliable. It can be running for a long time without any crash. Simplified interfaces in Windows 2000 Pro make it easier to use but with enhanced graphic display.

However Windows 2000 Pro has a weakness, lack of hardware and software compatibility. Hardware Compatible Lists (HCL) is used to define which hardware are compatible with Windows 2000 Pro, where software checked by Application Compatible Lists.

Windows Me

Due to users confused about the actual implementation of Windows 2000 Pro, Windows Me is released for home users. Windows Me somehow has higher hardware and software compatibility than Windows 2000 Pro. Other features of Windows Me are:

- *Removal of 'Real Mode DOS'*

'Real Mode DOS' is a running environment for older DOS applications and enables computer to be boot/reboot in DOS mode. Canceling of 'Real Mode DOS' make the operating system became more stable and reliable because 'Real Mode DOS' is one of the reasons of reliability problems encountered by previous version of operating system. Replacement of 'Real Mode DOS' is called 'Protect Mode' that allows operating system fully access to the power of hardware.

- *System File Protection (SFP)*

  SFP is a protection method that prevents important system files from being deleted by users. Whenever a system file has been overwriting by different file, the system will automatically put back the correct version.

- *System Restore*

  System restore is a unique service for Windows Me. This feature enables roll back the system to previous fine state when users wrongly configured some settings. However more hard disk spaces are reserved for this setting.

- *Auto update ability*

  Updating the old version of operating system is done manually by users to access the web sites and download them. But with auto update feature, if any updates are available, system can automatically download it and install it.

- *Help and Support service*

  This is a new HTML-based activity center that provides interaction between local help files with the online updates from the organization or 3rd parties' sites.

- *Improvement multimedia applications*

Some new or new version of applications are added included Image Acquisition, Movie Maker and DirectPlay Voice. Image Acquisition enables system to acquire images from scanners or digital cameras through a direct and simple interface, while DirectPlay Voice allows game players to chat with each other by using headset microphone that connected to their sound card.

- *Improvement of networking tools*

  Improved Home Networking Wizard (HNW) that help users to easily set up a home network, besides helping in file sharing, printing and connect to web.

Negative sides of Windows Me:

- Only 1 new update available in Internet Explorer 5.5 compare to the previous version IE 5.01
- Some integrated applications like Windows Media Player and Windows Movie Maker cannot be remove from system.

Windows XP

Windows XP is the newest and most important operating system released by Microsoft since Windows 95. Windows XP has a new user interface technology called 'Visual Style' that allow users to change the skins of the interfaces, buttons, web view or others. A CD-R burning utility also integrated in Windows XP that enables users to copy files to CD-R without install 3rd parties' software. There are two versions of Windows XP: Home and Professional Edition. Home edition is for consumer home users while Professional edition is for business or power-users. Features included in XP Professional that are not covered in XP Home are:

- Multiprocessor support
- Dynamic disk support
- Encrypting File System (EFS) used to encrypt individual files or folders for local security
- File level access control that each user is authorized to access specified applications.
- Multi-language support.

Unix

Unix is the first operating system written in C programming. Unix is not a single operating system like owned by anyone, but formed by a class of similar system. It is an open source and standard operating system that everyone can contribute to enhance or improve its functions. There are many different implementations of Unix system like Solaris, AIX, HP-UX, BSD/OS and MacOS.

Unix is a multitasking computing environment. It cans simultaneously running different independent processes. Unix also provides the multiuser capability that enables multiple users using a individual Unix system or access and running the same programs/files in the same time. As an open source system, Unix consists of a powerful programming environment for C, C++, Fortran and Java compilation. Free development tools also available for anyone wishes to design for the Unix system. Portability is another feature of Unix system. It can easily be moved from one PC to another without large changes to the codes.

Linux

Linux is a Unix-like operating system developed by Linus Toryalds. It is also an open source system that enhanced through contribution from the public. Similar to Unix, Linux also released for many different version: RedHat, Mandrake, Lycoris, Lindow and others.

The central part of Linux is called kernel that is developed and upgraded from public's contribution.

Advantages of Linux:

- *Low cost*

  Linux is licensed by General Public License (GNU) that means everyone is freely to use the software and will not expire.

- *Stability*

  Linux is an ultra stable operating system ever been developed. It can operate for a long time without facing any hang or slow down phenomenon.

- *High performance*

  Linux provides high performance computing for workstation or network. Like Unix, Linux can handles large number of processes in the same time.

- *Strong network support*

All contributions for enhancing Linux is done through network make it become strongly support for network connectivity and always compatible with the client and server architecture system.

2.8 Development Platforms

Microsoft.Net

Microsoft .NET is a set of Microsoft software technologies for connecting information, people, systems, and devices. It is a XML Web services framework. XML Web services allow applications to communicate and share data over the Internet, regardless of operating system, device, or programming language. Web services is the most important technology used in develop the Grid application.

→ Web Services Technology

A web services are self-contained and modular applications which support interoperate machine-to-machine interaction over a network. A interface is provided by web service to let other system to interact with web services. The interaction between web services and other system involved SOAP messages which typical conveyed using HTTP with an XML serialization in conjuction with other web-related standards. The major industry intiatives are Microsoft (.Net), IBM (Dynamic e-Business) and Sun Microsystem (Sun One).

Open Grid Services Architecture (OGSA) proposes the three main standards to implement web services in grid application. The three standards are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Web Service Inspection (WSI). The folowing is a briefly description of these standards.

- SOAP

It used to provide messaging between a Service Requestor and a Service Provider. It is a simple enveloping mechanism for XML payloads and defines a Remote Procedure Call (RPC) mechanism and conventions.

- WSDL

It is a XML document. A set of endpoints on messages is defined by WSDL that containing either document-oriented (messaging) or remote procedure calls (RPC) payload. WSDL is extensible to allow description of endpoints and the concrete representation of their messages fro a variety of different message formats and network protocol.

- WSI

It is a simple XML language. Related conventions for locating service descriptions which published by a service provider is did by WSI. A collection of service descriptions and links to other sources of services descriptions can be a content of a WS-Inspection Language (WSIL) document.

The mechanism which used to exchange messages is documented in a Web Service Description (WSD). WSD which written in WSDL is machine-processable specification. Messages format, datatypes, transport protocols and transport serialization formats are defined by WSD. All these information are used between the requester agent and provider agent. Besides that, some other related information can be also defined by WSD. The example is the network location at which a provider agent can be invoked.

The service description represents an agreement which govern the mechanics of interacting with that service. Hence, URL to a WSDL document is contained in a service description. Service description also consists of the URL to another WSIL document.

A concrete agent which is a computational resource must be implemented in the web services. Agent is a pieces of software or hardware in the web services. The purpose of implementing agent is to reliaze or request zero or more web services. The agent actually act as person or organization that perform the task.

→→Overview of engaging web services

The two entities which operate in the web services are the provider entity and the requester entity. The provider entity is the person or organization which provides an appropriate agent to implement a particular service. To make use of a provider entity's web service, the requester entity will use requester agent which can interact with provider agent. The figure below shows the relationship between provider entity and requester entity.

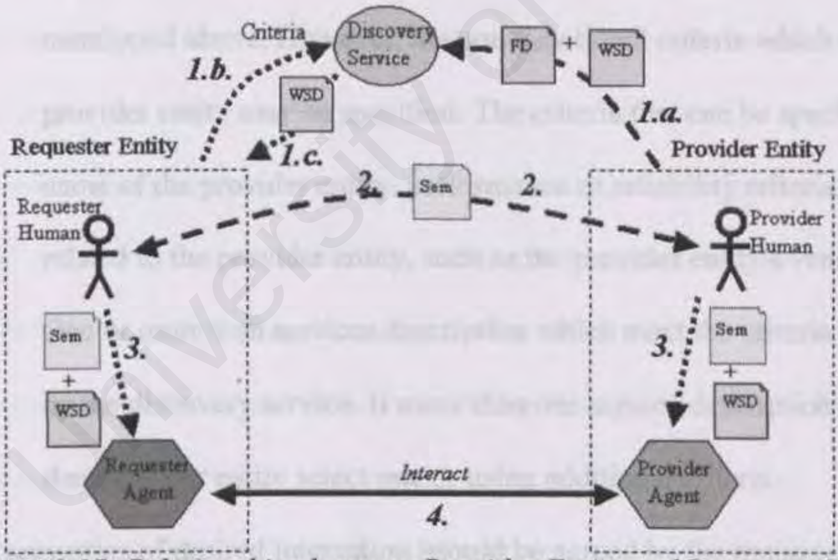


Figure 2.17: Relationships between Provider Entity and Requester Entity

A discovery service is needed when the requester entity which wishes to intiate an interaction with provider entity does not know what provider agent it wishes to engage.

The discovery is a service that facilitates the process of performing discovery. Either the requester agnet, the provider agent or some other agent can perform this service.

There are several steps that should follow to engage a service after discovery services performed.

1. The requester and provider entities " become known to each other"
   i. Both web service description and an associated functional description of the service are obtained by discovery service. The functional description is a description of the functionality of the service that offered by provider entity.
   ii. Some criteria should be supplied by requester entity to the discovery entity to select web services description. the associated functional description, capalities and potentially other characteristics is the criteria which mentioned above. However, the non-functional criteria which realted to provider entity may be specified. The criteria that can be specified are name of the provider entity, performance or reliability criteria, or criteria related to the provider entity, such as the provider entity's vendor rating.
   iii. One or more web services description which meet the criteria are returned by the discovery service. If more than one service description are returned, the requester entity select one or using additional criteria.
2. The semantics of desired interaction hsould be agreed by the requester and provider entities. Normally, the semantic is defined by provider entity. The requester just follow it. Besides that, the semantics can be defined in other ways. Example, the requester and provider entities can use the semantic which is

defined by some industry standards. Step 2 requires the parties agree on the service description.

3. The service description and semantics are input to both the requester agent and the provider agent.
4. SOAP messages exchanged between the requester agent and provider agent on behalf of their owners.

There are two types of discovery process which state above. The two type of discovery process are manual discovery and autonomous discovery. Manual discovery defined that a discovery service which used by a requester *human* to locate and select a service description that meets the desired functional and other criteria. Manual discovery typically is perform at the design time. Under autonomous discovery, a requester agent uses discovery service to perform this task.

A number of condition must be establish to ensure computer programs can interact with each other successfully.

- ➢ A physical connection must exist between the computer program
- ➢ There must be agreement on the form of data. This agreement will define whether data is lines of text, XML structure and etc.
- ➢ The aggreement must be shared by two or more programs as to the intended meaning of data
- ➢ There must be agreement as to the implied processing of messages exchanged between the programs.

→→ Microsoft .Net Framework

The Microsoft .NET Framework is the infrastructure for Microsoft .NET Platform. The Microsoft .NET Framework provides a common environment for building, deploying, and running web services and web applications. It also contains a lots of class libraries - like ADO.NET, ASP.NET and Forms and this enable the services to be integrated into a variety of computer systems.

Basically, Microsoft .NET Framework is language neutral and currently it supports C++, C#, Visual Basic, JScript and COBOL. Other third-party languages, for examples, like Eiffel, Perl, Python, and Smalltalk maybe will be integrated into it.

Java Platform

The Java platform is a software-only platform that runs on top of other hardware-based platforms. Hardware-based platforms are varying in their storage, memory, network connection, and computing power capabilities. So, specialized Java platforms are available to address applications development for and deployment to those different environments.

There are multiple of specialized platforms developed until now.

- Java 2 Platform, Standard Edition (J2SE)

  Provides an environment for Core Java and Desktop Java applications development, and is the basis for Java 2 Platform, Enterprise Edition (J2EE) and Java Web Services technologies. It consists of the compiler, tools, runtimes, and Java APIs that enable developers to write, test, deploy, and run applets and applications.

- Java 2 Platform, Enterprise Edition (J2EE)

  This platform defines the standard for developing component-based multi-tier enterprise applications. It is based on J2SE and provides additional services, tools, and APIs to support simplified enterprise applications development.

- Java 2 Platform, Micro Edition (J2ME)

  J2ME is a set of technologies and specifications targeted at consumer and embedded devices, such as mobile phones, personal digital assistants (PDA's), printers, and TV set-top boxes.

2.9 Programming Languages

Microsoft Visual C#.Net

Visual C#.Net is a modern and object-oriented programming language with flexible Integrated Development Environment (IDE) for creating XML web services and Microsoft .Net applications (interconnected applications for Windows and web). C#.Net is builds on strong C++ heritage. There is some significant improvement in C#.Net including a unified type system, "unsafe" code for maximum developer control and powerful new language that easily understood by developer.

C#.Net enables developer to take advantages of component-oriented language with inherent support for properties, indexers, delegates, and custom attributes. Developer can produces good source code documentation by using the XML comments. Besides that, an advanced inheritance model allows developers to reuse their source code from within any language that supports .Net.

With the C#.Net, developers can use the high technology for resources management and unified types. Developers also can gain the advantages of .Net memory management technology for seamless garbage collections and reduce program complexity. Another benefit from C#.Net is class library implementation. Using the class library, developers can gain power built-in functionality including a rich set of collection classes, networking support, XML schemas, multithreading, and ect Besides, with the Java Language Conversion Assistant (JLCA), developers can integrate their Java-based project into Microsoft.Net environment.

One of the most important technology in C#.Net is provides powerful web services construction. The power of this technology is a web applications can written using separate web services. Each module is one web services and they are independent with each other and also independent from platforms.

Java

Java is a powerful network programming language that can develop a secure, reliable, portable, and scalable application. Java enables developers to:

- Write software on one platform and run on another.
- Create program to run within a web browser.
- Develop server-side applications for online forums, stores, polls, processing HTML forms, and more.
- Write applications for cell phones, two-way pagers, and other consumer devices.

2.10 Databases

Microsoft SQL Server

Microsoft SQL Server 2000 provides an enterprise data management platform that can adapt quickly in a fast-changing environment. For the purpose of scalability, speed, and performance, SQL Server 2000 is a fully enterprise-class database product, providing core support for Extensible Markup Language (XML) and Internet queries.

Features of Microsoft SQL Server are:

- Rich XML Support

  The integration of back-end systems and data transfer across firewalls can be simplified by using XML

- Security

  Ensure the application is secure in any network environment by providing role-based security and network encryption.

- Replication

  Users can perform merge, transactional, and snapshot replication with heterogeneous systems.

- Full Text Search

  Enable users to manage their structured or unstructured data including searching through Microsoft Work documents.

- Data Transformation Services

  An alternate way for extract, transform and load data from heterogeneous sources.

MYSQL

MySQL is an open source database that provides a less complicated solution suitable for widespread application deployment. MySQL has several advantages:

- Reliability and Performance

  All the versions of released database server software are allowed for testing by the open source community before it is ready for production use.

- Ease of Use and Deployment

  MySQL's architecture makes it extremely fast and easy to manage.

- Freedom from Platform Lock-in

  Be an open source database, MySQL's approach ensures freedom and hence it can prevent from lock-in to a single company or platform.

- Cross-Platform Support

  MySQL is available on more multiple platforms including Linux distributions, Mac OS X, UNIX and Microsoft .

- Millions of Trained and Certified Developers

  MySQL is the world's most popular open source database, so it's easy to find high-quality, skilled staff.

MySQL Database Server is the main data management application. It is contains a multiple storage engines that provides full transaction support with commit, rollback, crash recovery and low-level locking capabilities. Users can set the range of storage engines from fast in-memory operation to full transaction support and clustering for high-availability. Query caching is a function that ensures performance and with database replication capability, MySQL enables slave servers to run on a single master server increasing both speed and robustness. Another integrated function is robust security system with advanced permissions and support for SSL transport-layer encryption provides robust application security. Besides, MySQL support full-text indexing and searching that provides rapid searching of text fields for words and phrases. This includes relevance rankings, exact phrase matching and Boolean search operators.

CHAPTER 3 - METHODOLOGY

3.1 Systems Development Process

The process of systems development is a set of methods, activities, deliverables and automated tools used to develop and maintain the system and software. A standardized systems development process help organizations achieve several convenient or advantages. First, using a consistent of systems development maximizes the allocation or shifting of resources within the project. Secondly, a common documentation is created that reduces the times and costs used to maintain the system.

The process involved in the life-cycle of system development included:

- *Identify the scope of project*

  Typically this is the phase of project. This phase helps developers to define the project objectives and limitations. Besides, the size and the boundaries of project can be established and also the expected budget and schedule.

- *Requirements and Problems analysis*

  Problem analysis is the study of developers on the existing system that helps developers to find out the weaknesses of the old system and therefore not to repeat it. Requirement analysis phase used to find out what are the functions that users expected from the new system and also what technology will be used. This is very important that it will directly affect the system implementation.

- *Systems design*

From the requirement analysis, developer can start to design the structure for the new system. System design involved transforms the business requirement into physical design specifications that used as guiding of the system construction.

- *Coding*

  Here is where the developers to start building the system using the appropriate programming languages.

- *Systems testing*

  Several of testing will be carried continuously on the system individual components or overall system to ensure the level of completed and also the reliability of the system.

- *Installation and delivery*

  After the testing phase, the system is ready for installation and delivery to the client. A proper user manuals and documentations must be released to help users to understand how to use the system.

- *Systems improvement and maintenance*

  This will be a long term processes until the system met its end time. The system needs several upgrades or modifications to overcome the changes in environment and requirement.

3.2 Methodology

A system development methodology contains a set of procedures, strategies and standards that provide an efficient structure to ensure the development of system meet the user's needs. The system development methodology also helps to minimize faults encountered during the development phases.

There are many life-cycle models in software engineering. For examples, waterfall model, evolution-tree model, rapid prototyping model, spiral model and others. Each model also has similar steps to implement: requirement, specification, design, implementation, integration and maintenance. Choosing the suitable life-cycle model is very important. There are 3 considerations when choosing life-cycle model.

- *Speed*

  Consider about if the system has to be delivered fast.

- *Size of project*

  Consider about how big is the system.

- *Uncertainty*

  Consider about the unclearly structures for building the system.

Each model has their own strengths and weaknesses. However iteration and incremental life-cycle model is considered for building the Grid Portal application.

3.2.1 Iteration and Incremental life-cycle model

This model can handle the changes on the under developed system. The development of system is divided into builds (increments) and each increment goes through multiple versions (iteration). Within each iteration is a waterfall process running. Multiple versions are constructed for each iteration enable developers to choose whether to make changes to the selected version or to left it remain unchanged.

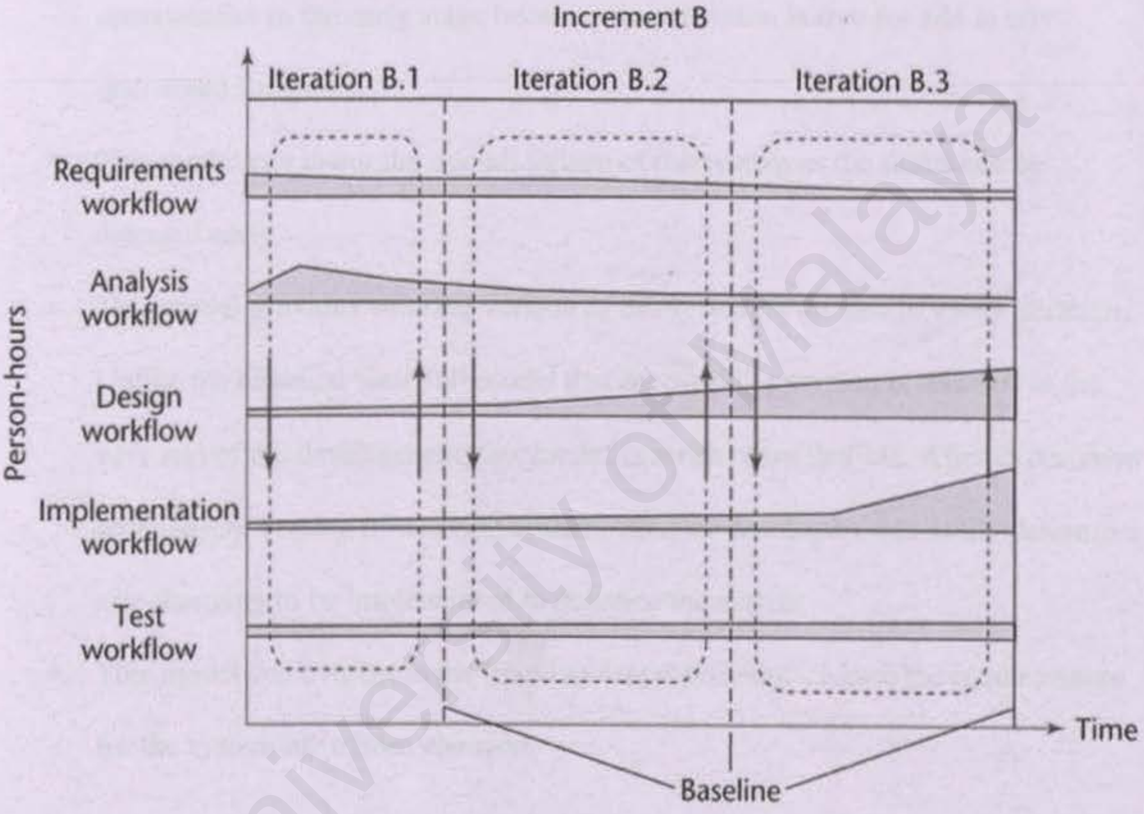


Figure 3.1: Iteration and Incremental life-cycle model

3.2.2 Why choosing Iteration and Incremental life-cycle model

Iteration and Incremental life-cycle model is chosen based on several advantages:

- Each iteration consists of checking that tests the work flow to make sure it is correct. So, faults can be detected early during the checking and this can reduce the time and cost required to develop the system.
- This approach is suitable for system development that contains a lot of uncertainties in the early stage because each iteration is free for add in any uncovered functions.
- This model can lower the overall failure of the system as the faults can be detected early.
- This model provides working version of the system at the end of every iteration. Unlike the classical waterfall model that the working version is released at the very end of the development, this model is much more flexible. After experiment with current version of working system, users or developers can easily determine any changing to be implemented to enhance the system.
- This model can overcome the 'moving-target problem', where the requirements for the system have been changed.

3.3 Fact Finding Techniques

During the requirement analysis phase in system development, varieties of resources are necessary for successful design the system. Therefore we need to collect any related facts to fulfill the development purpose. Fact finding is an alternative technique for doing so.

3.3.1 Studying of existing system

Through studying the existing system, developers can have more understanding about the structure and design of the system. Besides, developers can experiment the system advantages and weaknesses.

3.3.2 Internet Searching

Internet is a large database that provides a lot of information. Developers can search for any related or wanted resources by using the available search engine like Yahoo, Google, MSN and others. With clicking on the button, a list of relevant web sites will be prompted for developers to select.

3.3.3 Library

Developers can find for physical resources from library or document room (small version of library). Developers allowed borrowing or copying books from the library. Developers also can refer to senior report as a source of information.

CHAPTER 4 SYSTEM ANALYSIS

This chapter will discuss about the requirements of the system, included functional and non-functional requirement, hardware requirement, and tools for building the system.

The modules of the system:

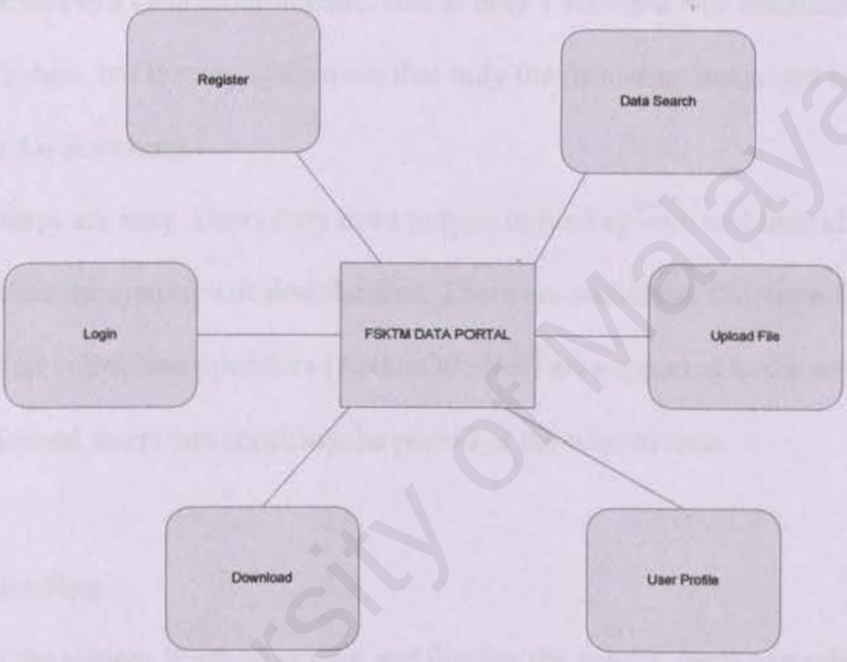


Figure 4.1: Overall Modules of FSKTM Data portal

4.1 Functional Requirements

4.1.1 Register & Login

Register and login are modules used to authorize users' access to the services in the system. New users are required to register themselves to the system by providing their personal details included user name and password. Each user will have a unique user name and password. Every time they want to use the available services, they have to login to the system.

4.1.2 Data Searching

It is not easy to find a file from billion of available resources on the internet. So the purpose of this module is enable users to find related data from several distributed databases located in a Grid environment. This is only a standard web interface like Googles or Yahoo, but is more systematic that only the databases integrated in the Grid are available for searching.

The steps are easy. Users only need to type in the keyword and then click on the button, and then the system will find the data. There are additional functions that users can apply. First is Boolean operators (AND, OR, Not) are supported in the searching keywords. Second, users can specified the period of the wanted data.

4.1.3 Downloading

After the system finish searching and display the results, users can select which files to be downloaded. This is a multisources downloading. The FSKTM Data Portal has an replication function that can dynamically copy the resources from one database to another when overhead occur. However the replicated file may divided into several pieces and replicated to different databases. So the pieces of files can be downloaded from different databases and combined become the original file.

4.1.4 User Profile

This module consists of 2 parts. The first part is history. History is a list that contains the search keywords that used by the users some time before. The purpose of this history is allows users for checking their searching before and provides faster returns from databases if users search with the same keyword.

The second part is user information. This section is for users to change their personal details included name, age, user name, password, address, and etc.

4.1.5 Uploading File

With this module, users can share their files by uploading the files to the databases in the Grid environment. Users only need to browse to the selected file and fill up some information about the file like file name, and then click on the upload button to successfully place the file into databases. After that, other users can search for that file using the FSKTM Data Portal.

4.2 Non-Functional Requirements

4.2.1 User Interface Friendliness

User interface is the only communication between the users and system. A simple and direct interface can make a users feel comfortable during using the system. The FSKTM Data Portal is like a normal HTML web page that consists of textboxes and buttons for searching data.

4.2.2 Security

The FSKTM Data Portal is using the SSL method to authorize users' login to the system. The system will check the user ID and password each time the users login into the system.

4.2.3 Flexibility

The FSKTM Data Portal is build using Grid technology that focused on distributed services. Hence other services can be easily integrated into the system.

4.2.4 Reliability

The back-end of the FSKTM Data Portal has a replication mechanism running all the time that ensures each connection to the databases will not broke down.

4.2.5 Interoperability

The system can run on different platforms: Windows and Linux.

4.2.6 Maintainability

The main core of this system is the back-end databases. As the system implemented using distributed databases architecture, all the data can be management through one interface without consider about whether the data is come from distributed databases.

4.3 Chosen Programming Language, Platform, Database and Middleware

4.3.1 Development Platform: Microsoft .Net

We are choosing Microsoft .Net because its powerful web services application construction is suit to the Grid concept, where all the applications are written into separately grid services. Microsoft .Net can provides flexible and secure implementation for the services run through the network in Grid environment.

Furthermore, Microsoft contains several functions that suitable for FSKTM Data Portal implementation. Because the communication between the portal system with its distributed databases is conducted using XML format query, SOAP[1] that included in Microsoft .Net is suitable for sending the XML messages via secure HTTP. Another reason is .Net support Secure Socket Layer (SSL) that ensures the secure login of users into the portal.

4.3.1 Programming Language: Microsoft C#.Net

For building the FSKTM Data Portal, we need to use third-party software called grid middleware as the system intermediate layer where all the services are written inside. The intermediate layer is as the bridge between the user interface and the back-end databases. Due to most the available middleware on the internet are developed in C#.Net or Java, we have chosen C#.Net.

Besides that, C#.Net is a modern programming language for web application that provides plenty of class libraries for writing the web services. C#.Net also can migrating

[1] SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is a XML-based protocol.

Java-based project into the Microsoft .Net development environment and its execution speed also much faster than Java.

4.3.2 System Platform: Windows XP Professional and Linux

As the FSKTM Data Portal is implemented on 2 different platforms, we have chosen Windows XP Professional and Linux.

Windows XP Professional is the newest released of Windows product from Microsoft. It includes the latest security version, Service Pack 2 that provides more secure protection on the operating system from the attacks of unsafe information come from internet. Windows XP Pro also has IIS resided on it that used to receive user requests. Besides, Windows XP pro is a stable and reliable system that most commonly used in the faculty research labs. So it is more convenient for us to develop our project and can achieve faster timing.

Linux is an open source system that can be found in the research lab. Linux is a ultra stable operating system that is suitable for develop an intensive data management application. This can ensure the connection between FSKTM Data Portal and the databases will not break down.

4.3.3 Databases: Microsoft SQL Server and MySQL

As the FSKTM Data Portal consists of 2 different types of databases, we have chosen Microsoft SQL Server and MySQL.

Microsoft SQL Server is a database that fully operated with the .Net framework and C#.Net. Through the using of SQLClient, developers can easily connect to the database from the .Net environment. Besides, the Data Transformation Services provided enable the system to extract or load data from heterogeneous sources. On the other hand, MySQL is a free database service that can running on the Linux system. We can easily find for any additional components or updates or related documents for the MySQL database due to it is an open source application available on the internet.

In common, Microsoft SQL Server and MySQL both provide replication and full text searching capability. These 2 features are important functions available within the FSKTM Data portal system.

4.3.4 Middleware: OGSI.Net

We are chosen OGSI.Net because it provides sufficient and comprehensive functions. Compare to MS.Net, OGSI.Net is a more completed version of grid middleware. The OGSI.Net has a dispatcher used to communicate the users' request with the corresponding services. The light-weight wrapper enables the system dynamically generated services instance for corresponding request or destroys the services instance. Besides that, OGSI.Net also supports the SOAP messages transfer between the application and databases.

CHAPTER 5 SYSTEM DESIGN

System design is a phase that follows system analysis. All the conceptual requirements defined in the system analysis phase will be converted into technical specifications.

In system design phase, there are several tasks to be implemented. First is architecture design, the whole system is represented in terms of their data, processes, interfaces, and components. Based on the results from analysis phase, for example, the system processes and data stores can be documented using a Data Flow Diagram. Second is the database design, this will state out the FSKTM Data Portal databases structure including the data relationship and data dictionary. The last one is the interfaces design. Interfaces enable developers to work with the actual input and output of the system.

5.1 Overview of FSKTM Data Portal Architecture

As you can see, the FSKTM Data Portal consists of one user interface that is the only interaction between the system and users. Every user login is protected by the Secure Socket Layer technology. After that, all the available services are defined in the intermediate layer and all the resources are come from the distributed databases that run on different platforms.

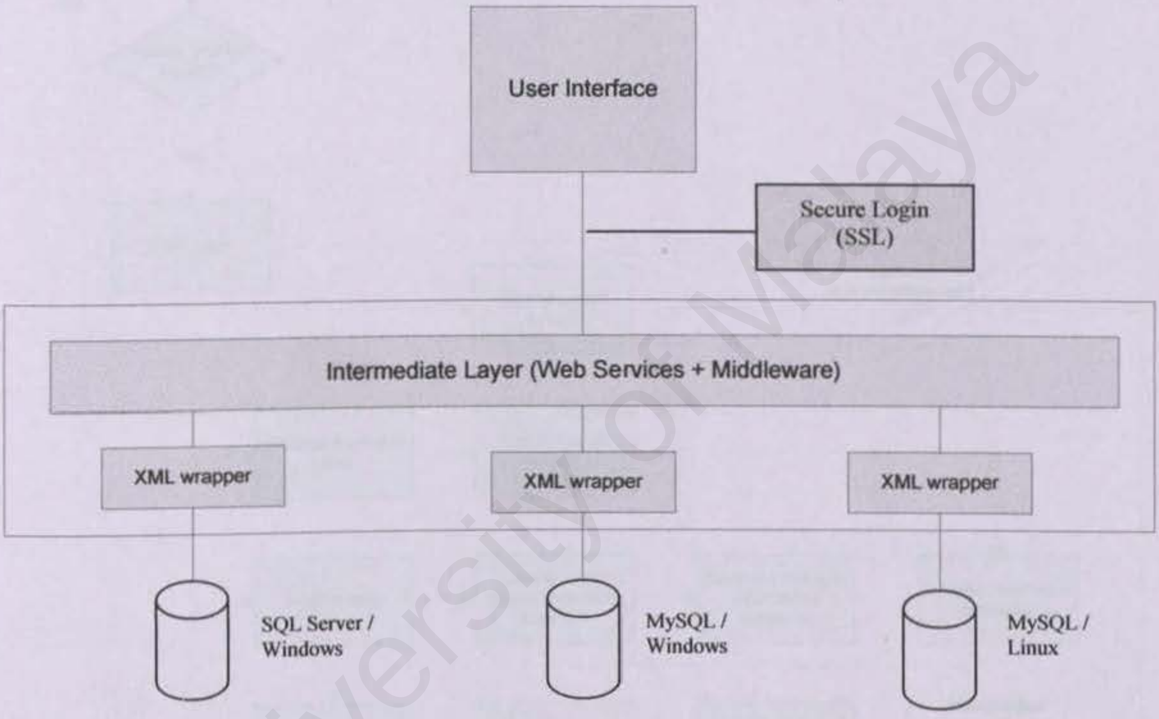


Figure 5.1: Overview Architecture of FSKTM Data Portal

5.2 System Work Flow

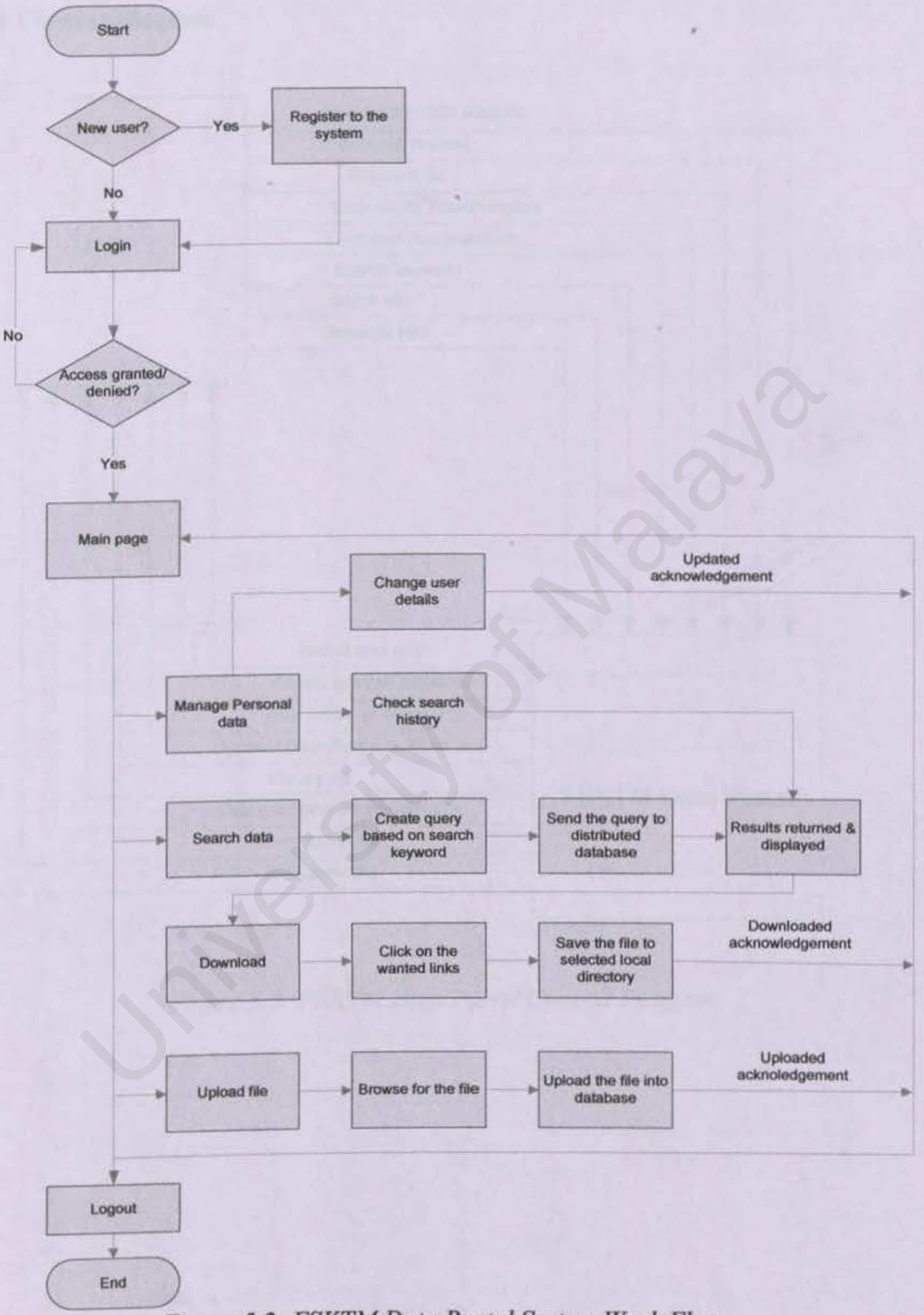


Figure 5.2: FSKTM Data Portal System Work Flow

5.3 System Functionality Design

5.3.1 Context Diagram

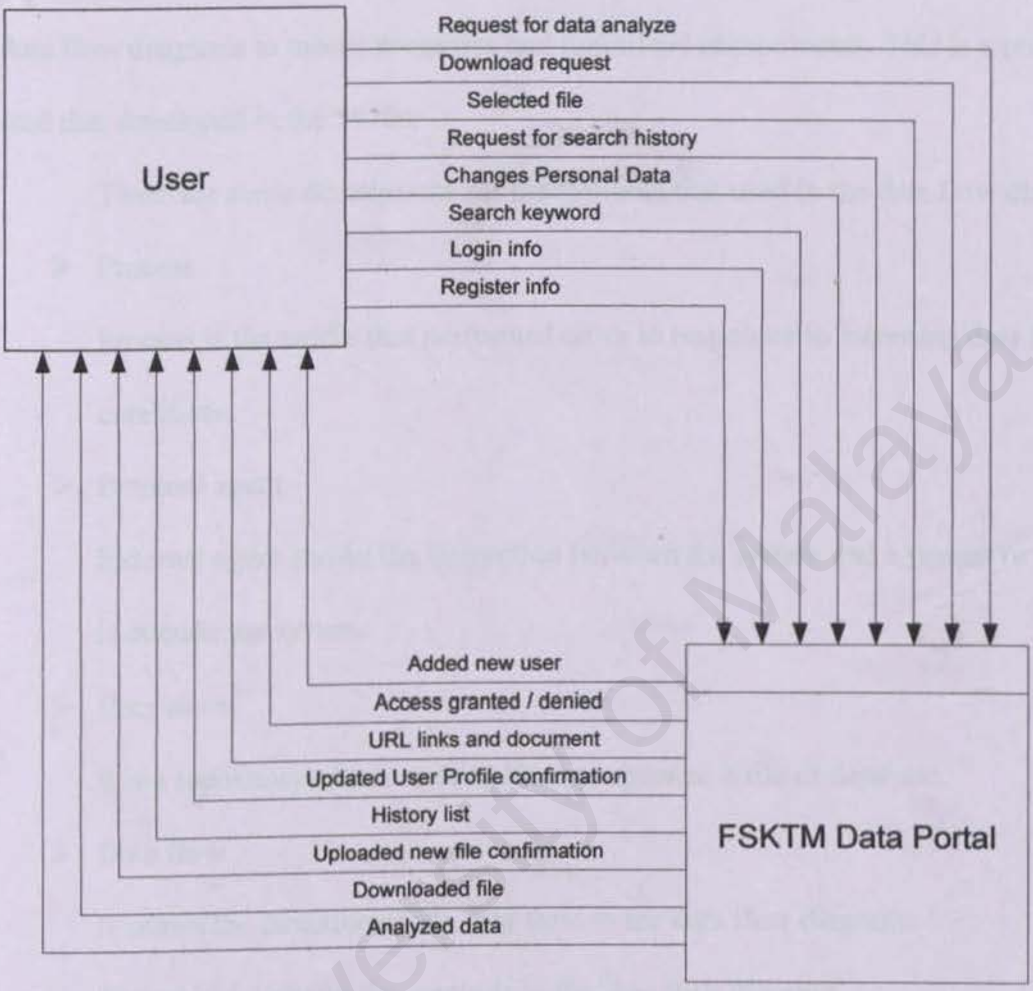


Figure 5.3: FSKTM Data Portal Context Diagram

5.3.2 Data Flow Diagram

The flow of data or information can be shown in the data flow diagram. Data flow diagrams can be grouped together or decomposed into multiple processes. Analysts use data flow diagrams to model processes and functional requirements. This is a primary tool that developed in the 1970s.

There are some descriptions for the symbols that used in the data flow diagrams.

- Process

  Process is the works that performed on or in responses to incoming data flow or conditions.

- External agent

  External agent shows the interaction between the system and a person or group. It is outside the system.

- Data store

  It is a repository of information. It can represent a file or database.

- Data flow

  It shows the direction of the data flow in the data flow diagram.

The table 5.1 shows the symbols in the data flow diagram.

Symbols	Attribute
□	Entity

⟶	Flow of Data
▢	Process
⊏	Data Store

Table 5.1: Symbols of Data Flow Diagram

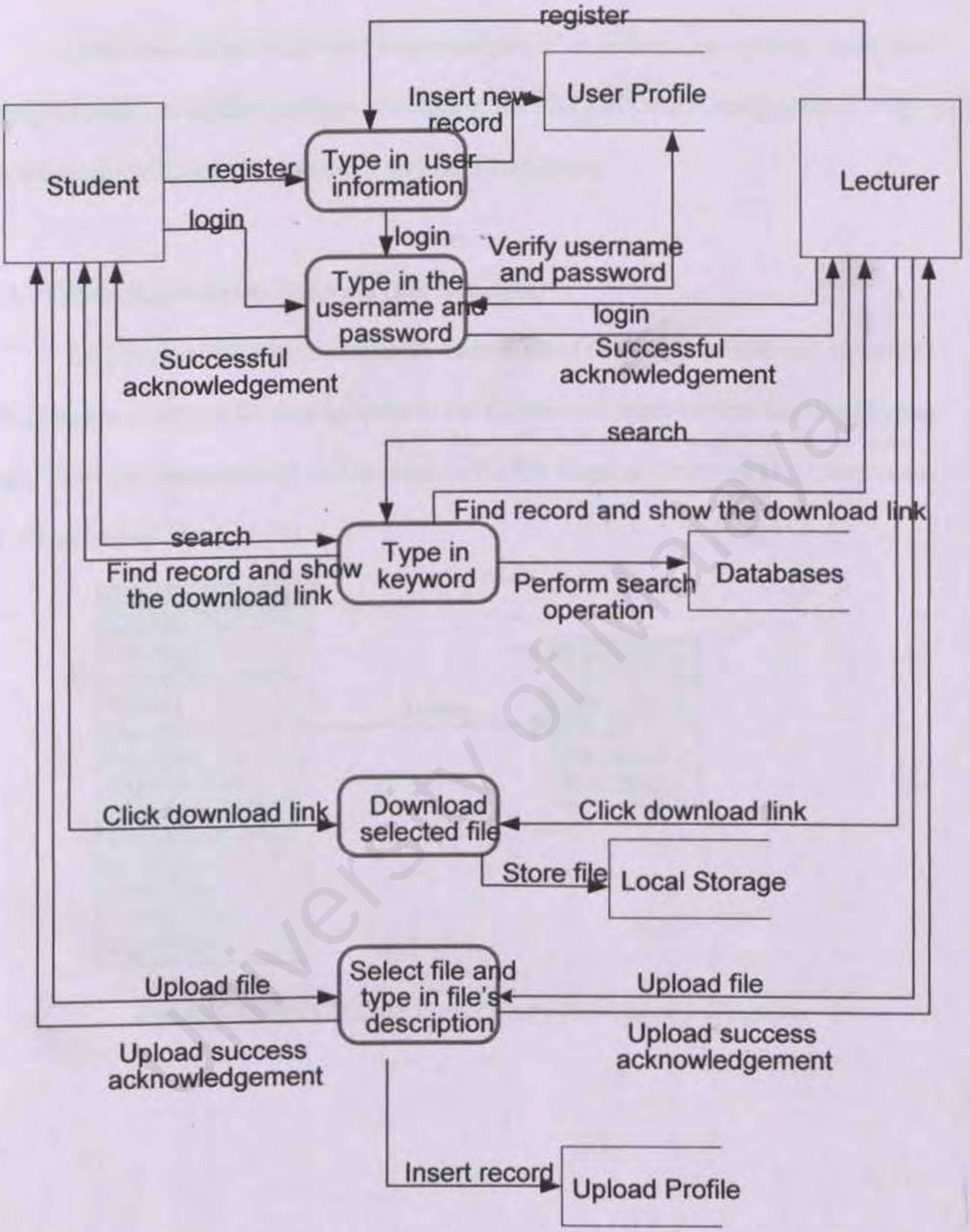

register
Insert new record
User Profile
Student
register
Type in user information
Lecturer
login
login
Verify username and password
Type in the username and password
login
Successful acknowledgement
Successful acknowledgement
search
Find record and show the download link
search
Type in keyword
Find record and show the download link
Perform search operation
Databases
Click download link
Download selected file
Click download link
Store file
Local Storage
Select file and type in file's description
Upload file
Upload file
Upload success acknowledgement
Upload success acknowledgement
Insert record
Upload Profile

Figure 5.4: FSKTM Data Portal Data Flow Diagram

5.4 Databases Design

Databases design is the most important part of an information system. There are 2 method used to show the databases structure of the FSKTM Data Portal system, Entity-Relationship Diagram (ER Diagram) and Data Dictionary.

5.4.1 Entity-Relationship Diagram (ER Diagram)

ER Diagram is the most frequently used method to show the databases structure. ER Diagram is depicts the data in terms of the entities and relationships described by the data. There are three types of relationships in the ER diagram: One: one (1:1), one: many (1 :*) and many: many (* : *).

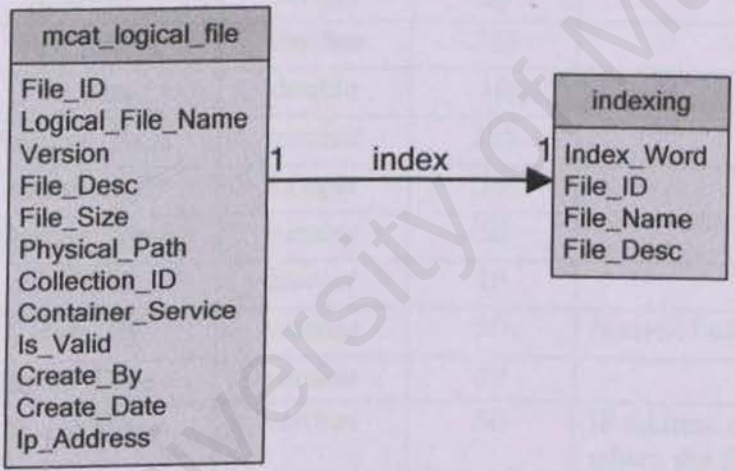


Figure 5.5: FSKTM Data Portal Entity-Relationship Diagram

5.4.2 Data Dictionary

Data dictionary is a special method used to show the databases in a form like dictionary as references to the developers. All the entities of each table in the databases including the entities name, type, description, and length will be documented for future modification.

Database name: dataportal_mcat

Table name: mcat_logical_file

Description: Stores the files information.

Field Name	Data Type	length	Note
File_ID	integer	10	
Logical_File_Name	varchar	255	
Version	integer	20	
File_Desc	varchar	255	
File_Size	double	10	
Physical_Path	varchar	255	
Collection_ID	integer	10	
Container_Service	varchar	50	
Is_Valid	integer	10	
Create_By	varchar	50	Name of owner
Create_Date	varchar	50	
Ip_Address	varchar	50	IP address of machine which the file located

Table 5.2: Table of mcat_logical_file

Table name: indexing

Description: Stores the indexed file information.

Field Name	Data Type	length	Note
Index_Word	varchar	50	
File_ID	varchar	255	
File_Name	varchar	255	
File_Desc	varchar	255	

Table 5.3: Table of indexing

CHAPTER 6 IMPLEMENTATION

6.1 Introduction

In the system implementation phase, the major work includes coding and debugging which are needed to solved the problems and errors found during the coding stage. Programming language characteristics and coding style can profoundly affect the system quality maintainability.

Coding step translates a detail design representation of software into a programming language realization. In a software project, the requirements analysis, system design and implementation phases do not have a clear boundary. Each phase tend to overlap one another. This phase at time involves some modifications to the previous design.

Certain attributes of the system may have to be implemented differently from the original plan due to hardware or software limitations. As long as the system fulfills critical requirements such as the core functions of the system and do not violate the pre-determined business rules, modifications to the system design is allowable.

In this chapter, a thorough description will be given on the entire implementation process and the approaches used to implement the system.

6.2 Development Environment

The development environment depicted here are the settings, conditions and surroundings where the actual system development takes place. It is important to take note of the environment in which the development was carried out because it may have effect on the speed and effectiveness of the system's implementation.

6.2.1 Software Requirement

Table 6.1 shows the software specifications that were needed during the system's implementation. They are essential towards the development of the system, e.g. IIS 6.0 must be installed as the web server because the system utilized a web application as the interface for the user to retrieve and view information from the database. The Microsoft Internet Information Services 5.1 is one of the Microsoft products. Hence, it is compatible in the .NET environment. Furthermore, since the development is carried out in a .NET environment, thus the Microsoft .NET Framework must be installed before system implementation can be carried out.

In the implementation phase, the core development tool is the Microsoft Visual Studio 2003. All the code-behind coding is implemented by using this tool. Besides that, Dreamweaver MX is used to develop the HTML coding and design the GUI interface.

Category	Description
Operating System	Windows XP
Web Server	Microsoft Internet Information Services 5.1
Special Requirements	Microsoft .NET Framework 1.1
Development Tools	Microsoft Visual Studio.NET 2003
HTML Coding	Dreamweaver MX

Table 6.1: Software Specification for the Implementation Process

6.3 Approaches to the Development Of The system

There are some methods or methodologies to implement the system design into the actual coding. A review of the coding methodologies, convention and the best practices for the developing the system is showed in the following sections.

6.3.1 Coding Methodologies

A study to the different approaches which can be used to develop the system is needed before starting the development of the system's implementations. There are two types of methodologies which are used to develop the systems. The two different methodologies are top down approach and bottom up approach.

6.3.1.1 Top-Down Coding Methodology

Top-down programming refers to a style of programming where an application is constructed starting with a high-level description of what it is supposed to do, and breaking the specification down into simpler modules, until a level has been reached that corresponds to the primitives of the programming language to be used.

Top-down programming tends to generate modules that are based on functionality. Typically, the high-level specification of the system states functionality. This high-level description is then refined to be a sequence or a loop of simpler functions or procedures, that are then themselves refined, etc.

Among the advantages of top-down implementations are:

i. Avoiding chaos of attempting to design the system all at once. Planning and implementing can be incredibly complex. Attempting to get all subsystems in place and running at once is basically planning to fail.

ii. Enables separate development works to be done in parallel on different but necessary subsystems that can save a great deal of time.

iii. Prevents the developer from getting mired in details that might cause him/her to lose sight of why the system is developed in the first place.

Although top-down implementations have a lot fo advantages, this methodology still have its weaknesses or disadvantages. There are:

i. All decisions made from the start of the project depend directly or indirectly on the high-level specification of the application. However, specification tends to change over time. When that happens, there is a great risk that large parts of the application need to be rewritten.

ii. There is a chance that the system will be wrongly divided into unsuitable subsystems. Attention must be paid to overlapping needs and sharing of resources so that the partitioning of subsystems makes sense for the system.

iii. Modules generated are very specific to the application that is being written, thus not very reusable

6.3.1.2 Bottom-Up Coding Methodology

An application is constructed starting with existing primitives of the programming language and constructing gradually more and more complicated features, until all of the application has been constructed. This types of coding methodology is bottom-up coding methodology.

In a language such as C or Java, bottom-up programming takes the form of constructing abstract data types from primitives of the language or from existing abstract data types. In Common Lisp, in addition to constructing abstract data types, it is common to build *functions* bottom-up from simpler functions, and to use macros to construct new *special forms* from simpler ones.

The advantages of bottom-up implementations are:

i. Testing is simplified since no stubs are needed. While it might be necessary to write test functions, these are simpler to write than stubs, and sometimes not necessary at all.

ii. Pieces of programs written bottom-up tend to be more general, and thus more reusable than pieces of programs written top-down.

iii. Critical functions can be coded first to test their efficiency.

At the other end, bottom-up implementation also has its weaknesses, as described below:

i. Development maybe somewhat directionless because overall program organization and higher level design is postponed.

ii. Developers may get too deep into the specific details of the different modules of the system and end up missing the deadline.

iii. The sub-modules of the system will still be eventually integrated.

6.3.1.3 The Chosen Methodology

After reviewing the two approaches, the top-down coding methodology is used to develop and implement the system.

This decision is made partly due to the characteristics of the chosen programming language for development: C#.NET and ASP.NET which both advocates the usage of object oriented programming practice. Development in an object oriented environment practically envisions an abstract object at the top level before going down to the specifics.

Bottom up coding convention is more suitable for function driven programming languages like C and C++.

6.4 Coding Implementation

To implement the system design into actual coding, it is almost get started with the coding process. Before that, it is important to come up with a set of rules which should be abided by during the coding process, as described in the following sections.

6.4.1 Coding Convention

The important attribute of the source codes is the coding convention. It determines the intelligibility of a program. Internal (source code level) documentation, methods of data declarations and approaches to statement constructions are the element of coding conventions.

Several coding conventions have been employed in the coding process to ensure system consistency, maintainability and readability. These coding conventions are as follow:

i. Use meaningful variable names, constant names, procedures names and parameter variable names to self document a program without excessive use of comments.

ii. Plan the layout of the program source code to improve its readability. Each sentence is started on a new line; statements following control structures are indented; white space is used to set off related blocks of codes, etc. These may seem insignificant but will help reduce the time needed to understand the program flow in the case where other programmers refer the code.

iii. All variables are declared at the beginning of procedures and declarations are separated from executables statements with a blank line to improve program readability.

iv. Insert comments to document the program and improve program understandability.

v. Group related types of codes together

6.4.2 Coding Documentation

To begin the coding documentation, it should start with the selection of identifier names and then continue with the composition of the connecting all separate components. At the end, it should follow by the re-organization of the program.

Elements taken to consideration in coding for an easy to maintain and enhanced system are internal documentation, standard naming conventions and standard graphical user interfaces.

To provide a clear guide to programmer for future enhancements, comments are used in coding. The source code should embed with the statements of purpose indicating the functions of modules and descriptive comments. This is useful to describe the processing functions

Standard naming convention and also a standard usage of graphical user interface components is employed in developing the system. Standard naming conventions provides the programmers with easy identification of variables. A standard usage of graphical user interface components provides users an environment that will not generate much surprise to them.

6.4.3 Classification of Program Code

In this system, program coding is basically can be divided into a few sections. All these sections are divided based on the functions that performed. They are:

i. Database operations

This includes populating ASP.NET server controls with values, displaying data from database and basic data operations such as creating, updating and deleting records.

ii. Web Services

Web Services provide a group of functions that let the web applications to invoke.

iii. Windows Service

Windows Services run as agent in the client computer to let the web services to invoke for certain functions in this system.

6.4.3.1 Database operations

(i) Importing the required Class

Operations related to database for this system utilized the ADO.NET data access technology. Therefore, all database related operations must have the following statement:

```
using System.Data.SqlClient;
using System.Data.Odbc;
using System.Data.OleDb;.
```

(ii) Connecting to the Database

The database connection string is stored in the Web.Config file.

```
<add key="MySQLConnection" value="driver={MySQL ODBC 3.51
Driver};server=127.0.0.1;database=dataportal_mcat;uid=mca
t;pwd=mcat" />
<add key="SqlConnection"
value="server=(local);database=DataPortal;uid=portal;pwd=
portal" />
```

Therefore, any reference to the connection string is done like this:

```
String strConnection =
ConfigurationSettings.AppSettings["SqlConnection"];

SqlConnection objConnection = new
SqlConnection(strConnection);
```

Putting the connection string in the Web.Config file ensures that any future changes to the database (location, name, etc) will only need to be done to this file and not to the respective connection strings in all the different modules.

(iii) Database Operations

A standard database operation can utilize any of the ADO.NET components, including the Data Set, Data Adapter, Data Table, Data View, Data Column and Data Row. Below is a sample code that let user to change their password.

```
string strQuery = "update user_profile set
password="+"'"+password+"'" +"where
userName="+"'"+username+"'";

SqlCommand command = new SqlCommand(strQuery,connection);
command.ExecuteNonQuery();
connection.Close();
```

6.4.3.2 Web Services

The web services contain the core functions of this system. The web applications need to invoke the functions in the web services to execute functions in this system. Below is a sample for user to download a file.

```
byte[] buffer = new byte[1024];
IPAddress = ipAddress;
FileStorePath = path;
DownloadPath = "C:\\Inetpub\\wwwroot\\FSKTM Data Portal\\temp\\";
Size = size;
string message = "";
int total = 0;


TcpClient client = new TcpClient();
client.Connect(IPAddress,5356);

byte[] image = new byte[1024];
NetworkStream netStream = client.GetStream();
StreamWriter sw = new StreamWriter(netStream);
sw.WriteLine(FileStorePath);
sw.Flush();
sw.WriteLine(Size);
sw.Flush();

string filename = Path.GetFileName(FileStorePath);
DownloadPath = DownloadPath + filename;
FileStream fs = new
FileStream(DownloadPath,FileMode.OpenOrCreate,FileAccess.ReadWrite
);
while (total < Size && netStream.CanRead)
{
      int i = netStream.Read(image,0,image.Length);
      fs.Write(image,0,i);
      total = total + i;
}
netStream.Close();
fs.Close();
sw.Close();
```

6.4.3.3 Windows Services

Below is the sample code for user download a file by invoke the windows services.

```
TcpListener listener = new TcpListener(address,5356);
listener.Start();

Socket socket = listener.AcceptSocket();
if(socket.Connected)
{
      byte[] buffer =new byte[1024];
      NetworkStream stream = new NetworkStream(socket);
      StreamReader sr = new StreamReader(stream);
      string path = sr.ReadLine();
      FileStream file = new FileStream(path,FileMode.Open,
FileAccess.Read);
      int i = file.Read(buffer,0,buffer.Length);
      stream.Write(buffer,1, i);
      stream.Close();
}
```

CHAPTER 7 TESTING

7.1 Introduction

System testing is the testing of a complete system prior to delivery. The purpose of system testing is to identify defects that only surface when a complete system is assembled. Testing of performance, security, configuration sensitivity, startup and recovery from failure modes are included in the system testing. Besides that, system testing consists of four basic concepts:

i. Error detection

   Identify error through some approaches such as inspection, walkthrough or others.

ii. Error removal

   Debugging and removing the identified error

iii. Error tracking

   Finding and correct the cause of the errors

iv. Regression testing

   Testing is executing after the changed or modify on the code.

Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

This can help in pin-pointing the goals that the testing process needs to achieve. The objectives of the testing process have been determined as below:

i. Software Reliability

Making sure that the system performs critical tasks or core functions as pre-determined in the system design phase. Furthermore, specific user requests and business rules must be fulfilled.

ii. *Software Quality*

Software quality is characterized by the correctness of program logic and implementation. It begins with testing the software during development. Each module must be tested to make sure that it functions correctly at the time it is written or modified. Test values and boundary conditions must both be verified. Next, the module should undergo interface testing to check for functional errors.

iii. *System Assurance*

The main purpose of system assurance is to deliver a quality product. Conformance to requirements increases the organization's confidence in the system.

iv. *Optimum Performance and Capacity Utilization*

Another purpose of testing is to ensure optimum performance and capacity utilization of e-commerce system components. The purpose of stress or capacity testing/planning is to make sure that the system is able to perform acceptably at peak usage.

v. *Price of non-conformance*

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

7.2 Types of Testing

There are several testing to let developers top detect and correct the error in the system. The types of testing are including unit testing, integration testing and system testing.

7.2.1 Unit Testing

The process of test the procedure, function or the object class of the individual component is known as unit testing. Developers use this approaches to find out the errors in the components. The purpose of the unit testing is to ensure all the components within a system can operate correctly. There are several steps being carried out for this approach:

i. Develop test cases to show input is properly converted to desired output

ii. Boundary conditions are tested to make sure the functions run at boundaries established for limiting or restricting process

iii. Debugging the error to find out the cause of error and fix it

7.2.2 Integration Testing

The integration testing is performed after all the object, components and individual sub modules have passed to local unit tests. To ensure the system that develop meet with the system design specification, a verification process is carried out to test the system. Integration testing ensures the valid linking and dynamic relationships which establishes between modules in the whole system.

7.2.3 System Testing

System testing is a series of different tests which primary purpose is to fully exercise the computer-based system. It is designed to reveal bugs not possible to individual components or to interaction between components and modules. System has been tested thoroughly to ensure that the simulation run smoothly. There were involved functional testing and performance testing for the system testing of the system.

CHAPTER 8 SYSTEM EVALUATION

8.1 Introduction

At the end of the System Development Life Cycle, after the product has been delivered and deployed, the system developer is left with one final task: evaluate the system, the processes involved to develop the system and most importantly lessons learnt from the entire lifetime of the system development.

This is important as it can help us identify best practices which have been applied and remember to use it again in the future. Besides that, we must also take note of the less than perfect events that happened and remember NOT to repeat the same mistake again.

Every system should go through the system evaluation phase, for the benefit of the system and also the system developer. It is only through experience that system developers can learn and enhance their skills. Therefore, a thorough evaluation should always be carried out at the very end of every system's life cycle.

8.2 Problems Encounter and Solution to overcome them

The development of the system was definitely not problem-free. With a time constraint impended on the system, it is important to overcome problems as fast as possible and with the best available solutions. The project schedule should be followed stringently amid problems to ensure the system will be delivered on time.

8.2.1 Difficulty in Choosing Suitable Development Tools

With the introduction of .NET technology, it was a blessing in some ways but a huge limitation in others. .NET provides unimaginable possibilities with its cross-language feature, managed code concept and powerful language capabilities. However, once you decide to plan the system on top of a .NET platform, there is no turning back.

There is no choice of development tools as currently there are only one .NET compliant development tools, and that is the Visual Studio .NET 2003. Whether you like it or not, you will be stuck with it for the entire lifetime of the system. There is no freedom in choosing the development tools you like, e.g. there are a myriad of development tools for the Java language: JCreator, Sun Suite or even as simple as the notepad.

As the problem with .NET is inevitable, I tried to work around the dilemma by doing extensive research on the Visual Studio.NET suite to make sure I fully understand the features embedded inside and make full use of it. If I can't fight the problem, I might as well try to work with it.

8.2.2 Difficulty in Choosing Suitable Programming Language

After deciding on the .NET technology, I was faced with the problem of choosing the programming language for my system. In order to satisfy everybody and cater for a large audience of programmers, Visual Studio .NET 2003 encompasses a variety of programming languages, among them: Visual Basic.NET, C#.NET, Visual C++, ASP.NET and J#.

With limited knowledge on the respective programming languages, it was understandable that I was worried I will end up choosing the wrong language for the system.

It was impossible to try out all the programming language and decide which is more suitable as there was not enough time. Therefore, I decided to seek advice from professional programmers who were proficient in all the languages. I did this by joining online forums on programming languages and asking the opinions from different people to avoid biased judgments.

As a result, I got more than enough help from the online forums. The users there have been extremely helpful and provided many valuable insights on the characteristics of each programming language.

8.2.3 Lack Knowledge in the Chosen Programming Language

I did not have any experience before in the chosen programming language: C# and ASP.NET. Again, there was no sufficient time to learn everything entirely from scratch. Thus, I had to pick up pieces from time to time as I moved on through the development stages of the system. Due to lack of full understanding on the concept of these two languages (plus the fact that they were very new languages themselves), I encountered some unavoidable problems during the coding of the system.

Again, I turned to the Internet for the possible solutions to my problems. I limited myself to not spending more than three days on the same problem. If the same problem persists, I will look for alternative solutions. It is just not feasible to hang on to the same problem, even if I might eventually manage to solve it three weeks later. There was a deadline to consider.

8.2.4 Lack of Knowledge on the Chosen Technology

.NET Technology is still in its infancy. We have never been exposed to it formally in lectures and there is no special courses dedicated specially to it yet. Therefore, a lot of research has to be done on our own.

.NET is very famous for its own "managed library" where a huge collection of classes and methods have been created specially for the .NET technology. This means that all the .NET programming languages can utilize it and eventually any cross usage between the programming languages is supported also.

From the beginning, I had identified a few classes from the .NET managed library which are essential for the development of my system. However, as time passes by, I realized that the classes that have been chosen were unable fulfill all the needs of my system. Due to lack of understanding of the .NET framework, I was faced with the problem of having an incomplete system on my hands.

Fortunately, there was still the traditional API library to turn to. It was found to be compatible with the .NET framework and the .NET languages. In the end, the classes from the API library were used instead of the .NET managed library.

8.2.5 Difficulty to Choose a Middleware

After deciding chosen .NET, a suitable middleware become another problem in this system. The current Grid middleware is developing in two main programming languages which are Java and .NET. Hence, the research on the middleware starts to find out the suitable middleware to develop the system in the .NET environment. I find more middleware and samples which operate in the .NET environment through search engine such as Google and Yahoo.

With limited knowledge on the middleware, it is difficult to find out the most suitable middleware. Besides that, time constraints become another problem to choose suitable middleware.

Fortunately, many researchers put the relevant resources in the internet. These resources are helping me make more suitable decision in the choosing suitable middleware.

8.3 System Strengths

The system has a few characteristics which denote the strength of the system. There are as described below:

i. Immediate response to User Query

   By utilizing the "Auto Post Back" function of the ASP.NET controls, any selection done on the controls is posted back to the server for immediate action without having to press a submit button. This ensures minimized response time to the user request and higher satisfaction for the user.

ii. Reduced interactions with the Database

   By using the newly introduced features of the ADO.NET data access technology, interactions with the database is reduced with the Data Set object where records are only retrieved once and loaded into the Data Set object for subsequent data operations. After the data operations are no longer needed, the records in the Data Set object can be updated back to the database. Therefore, records are only retrieved and copied back once.

iii. User Friendly Graphical Interface

   The graphical user interface design is tailored to the average user capabilities and promises easiness of usage. Even a novice user will have no problem navigating through the web application.

iv. Reusability

   Source codes are organized into reusable classes in an object oriented environment. Interface codes are separated from the logic processing clearly. This advocates for easy reuse of the different components of the system or while adding future enhancements to the source codes.

v. Automatic startup at System Boot Time

The agent which installed in the client computer will start up automatically after system boot up without any manual intervention by the administrator.

8.4 System Limitations

The finished system is not without its limitations. There are described as below:

i. Invisibility of the Indexing Agent

The agent which is responsible for indexing the data in the database on the schedule time still visible from sight although can hide to the system tray. Users can stop the agent easily.

ii. Limitation to the Windows platform

Although Grid computing do not have any limitation to the platforms, this system just only execute in the Windows platform due to the chosen programming language. The system cannot operate in the Linux environment.

iii. No file replication

The entire files in the databases are standalone and no replication. Users can only retrieve file from single databases. Therefore, users fail to get the files if the files are corrupted or deleted by the owner.

8.5 Future Enhancements

There are a few suggestions for future enhancements to overcome the limitations of system. The suggestions are showed below:

i. Agent invisible from sight

The indexing agent should be visible from sight to ensure users do not stop agent easily.

ii. The system should operate in different platforms

To satisfy the Grid computing requirement, the system should operate in different platforms. The chosen middleware play important role to overcome this problem.

iii. File replication

The system should do the replication to improve the performance of the system

BIBLIOGRAPGY

Document

[1] Ian Foster, Carl Kesselman, and Steven Tuecke (2004). The Anatomy of the Grid Enabling Scalable Virtual Organizations. Available at http://www.globus.org/research/papers/anatomy.pdf

[2] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. (2004). The Data Grid:Towards an Architecture for the Distributed Management and Analysis of Large Scientic Datasets. Available at http://www.globus.org/documentation/incoming/JNPCApaper.pdf

[3] Vijayshankar Raman, Chris Crone, Laura Haas, Susan Malaika, and Chaitan Baru. (2002). Data Access and Management Services on Grid. Available at www.cs.man.ac.uk/grid-db/papers/dams.pdf

Sample Application

[4] Biology WorkBench. Available at http://workbench.sdsc.edu/

[5] DSG Grid Portal. Available at https://l59.dsg.port.ac.uk/dsgPortal/

[6] CCLRC e-Science DataPortal. Avaliable at http://www.e-science.clrc.ac.uk/web/projects/dataportal, http://dataportal.dl.ac.uk:8080/, and http://dataportal.dl.ac.uk:8080/dp/

[7] Alliance Science Portal. Available at http://www.extreme.indiana.edu/alliance

[8] International Lattice DataGrid. Available at http://www.lqcd.org/ildg/

Grid Middleware

[9] GridPort middleware. Available at https://gridport.npaci.edu/

[10] MS.Net Grid. Availbale at http://www.epcc.ed.ac.uk/~ogsanet/

[11] OGSI.Net Grid. Available at http://www.cs.virginia.edu/~humphrey/GCG/ogsi.net.html

[12] Globus Toolkits. Available at http://www.globus.org

[13] MyGrid. Available at http://www.mygrid.org.uk

[14] MSRF.Net. Available http://www.cs.virginia.edu/~gsw2c/wsrf.net.html

Web Sites

[15] http://www.mysql.com

[16] http://ip158.fsktm.um.edu.my/ilmiah2004/

[17] ftp://202.185.107.158/research/thesis

[18] http://www.winsupersite.com/reviews/

USER MANUAL

This user manual is for the FSKTM Data Portal. This user manual divides into two parts which are prerequisite requirements and description of the each module's functions.

Prerequisite Requirements

1. Install WSRF.Net and Web Services Enhancement Service Pack 1 in the server machine.

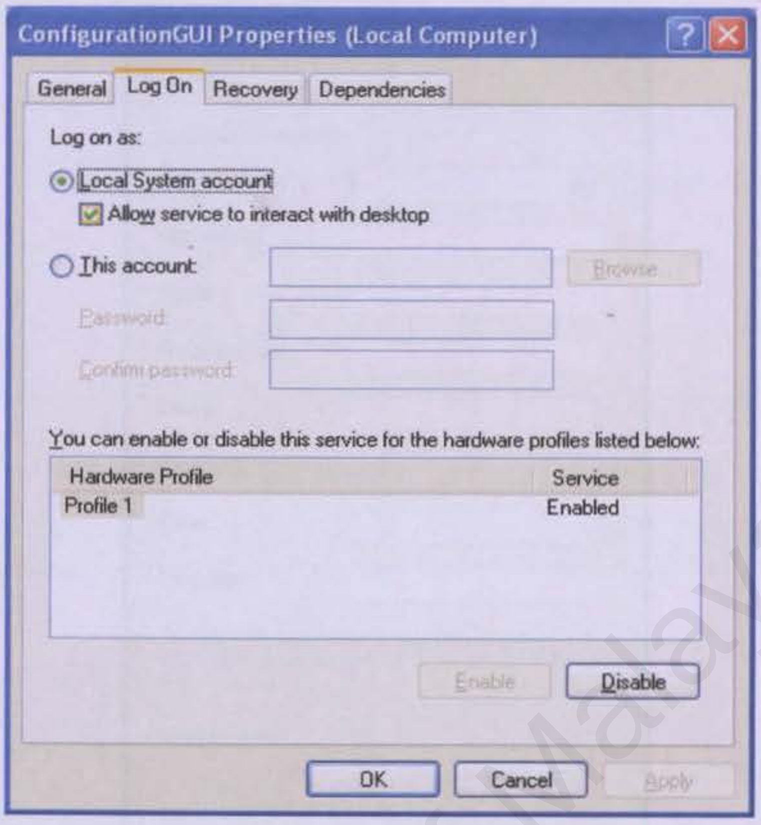
   PS: Within the CD submitted, you can finds out the setup file for the WSRF.Net, WSE sp1, and the installation instructions file.

2. Install the GridServer into the server machine. Then go to Control Panel -> Administrative Tools-> Services, start the ServerSocket.

   PS: to install the GridServer, just run the Setup.exe in the Setup1 folder.

3. Install the GridAgent into the client machines. Then go to Control Panel -> Administrative Tools-> Services, select the properties of ConfigurationGUI then select the allow service to interact with desktop. After that, start the ConfigurationGUI and download.

   PS: to install the GridAgentr, just run the Setup.exe in the Setup folder.



Log on as:

● Local System account

☑ Allow service to interact with desktop

○ This account:

Password:

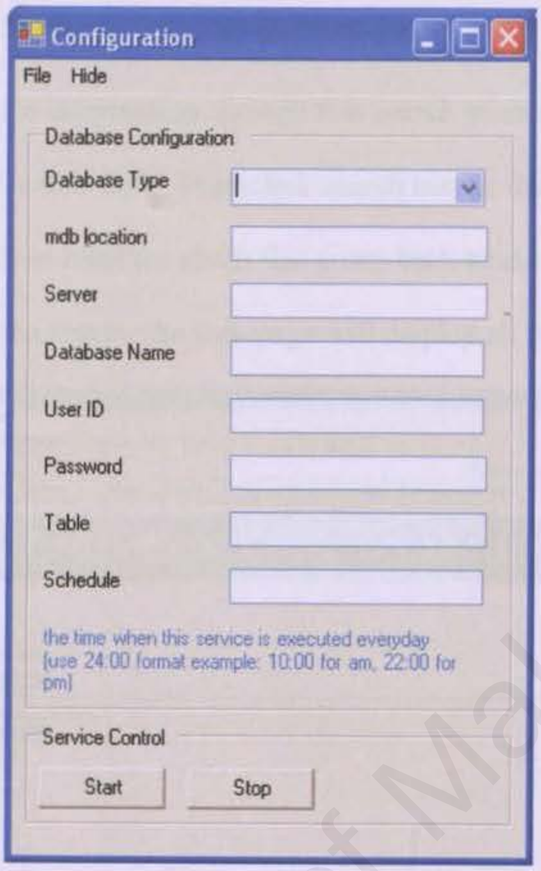
Confirm password:

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure i: The properties dialog box of ConfigurationGUI

Another GUI is prompt out. User need to fill in the all the information that requested and click the start button.

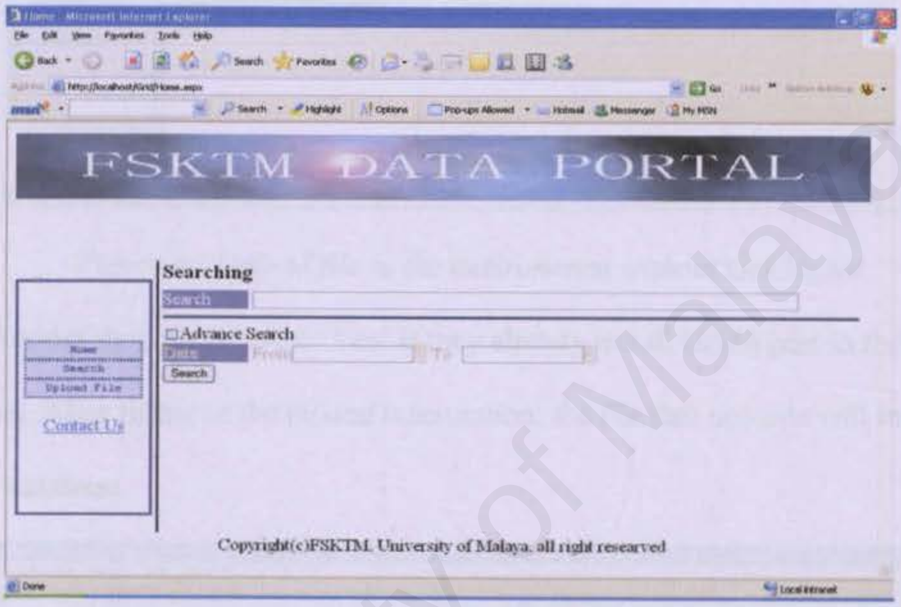


the time when this service is executed everyday (use 24:00 format example: 10:00 for am, 22:00 for pm)

Figure ii: GUI of ConfigurationGUI to let users fill in information

Functionality of the System

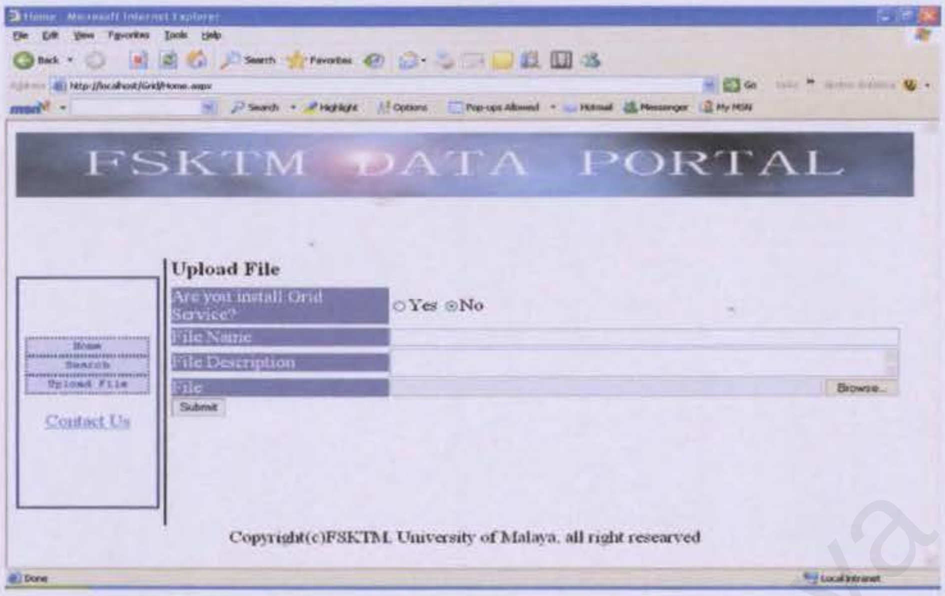
1. Users can search the information through this search module. Users need to key in keyword in the search field. Then click search button. Besides that, users can use advance search to limit the result that come back based on the date. After the result returned to the server, the web page will display all result to users.



Copyright(c)FSKTM. University of Malaya, all right researved

Figure iii: Search the information

2. Besides that, users can upload file to the databases in the Grid environment. If the users do not install the GridAgent in the client machines, they need to choose the 'No' in the first question. Then the users require filling in the related data and browsing for the files which need to upload. Then click submit button. The files will copy to the default database in the server.


Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/GridHome.aspx
Go
Search Highlight Options Pop-ups Allowed Hotmail Messenger My MSN
Home
Search
Upload File
Contact Us
Browse...
Submit
Done
Local intranet

FSKTM DATA PORTAL

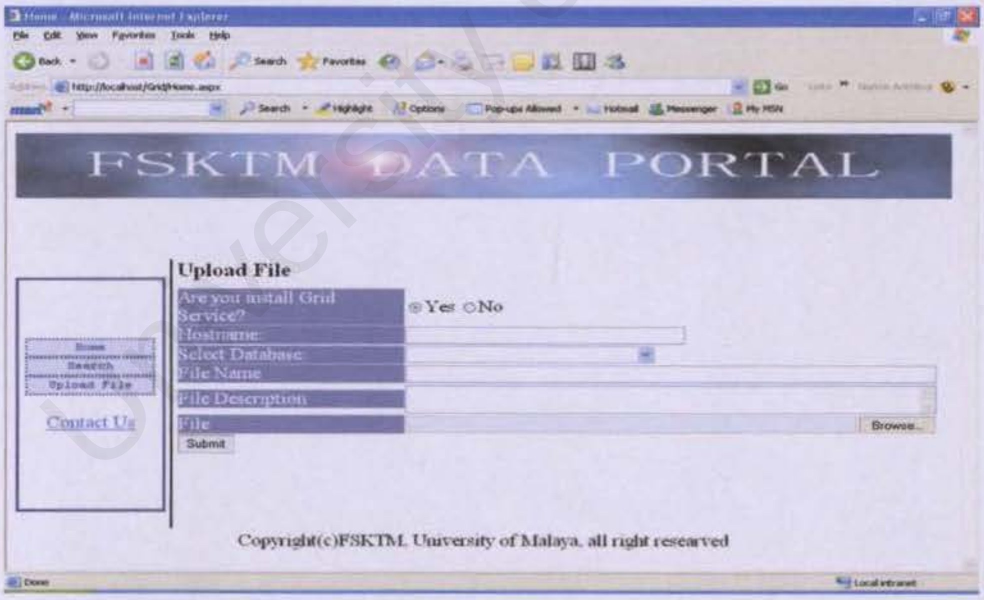
Upload File

Are you install Grid Service?	○ Yes ● No
File Name	
File Description	
File	

Copyright(c)FSKTM, University of Malaya, all right researved

Figure iv: Upload file in the environment without GridAgent

Besides that, users select 'Yes' if they already install GridAgent in the client computers. After filling in the related information, the file that uploads will insert into the local database.


Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/GridHome.aspx
Go
Search Highlight Options Pop-ups Allowed Hotmail Messenger My MSN
Home
Search
Upload File
Contact Us
Browse...
Submit
Done
Local intranet

FSKTM DATA PORTAL

Upload File

Are you install Grid Service?	● Yes ○ No
Hostname	
Select Database	
File Name	
File Description	
File	

Copyright(c)FSKTM, University of Malaya, all right researved

Figure v: Upload file in the GridAgent environment