私立東海大學

資訊工程研究所

碩士論文

指導教授：楊朝棟　博士

設計與實作一個以服務為導向架構之醫療網格平台

Design and Implementation of a SOA-based Medical Grid Platform

研 究 生：蕭珽之

中華民國九十八年六月

摘要

隨著資訊科技的日新月異、網路傳輸的無遠弗居，過去僅會出現於電影的神奇醫療情節，已漸漸地在我們日常生活中上演了。隨著醫院資訊化的快速成長，各類資訊系統的不斷開發和使用，然而在醫院資訊系統各自為政的開發方式之下，系統之間的資訊共用顯得日漸困難。

為了滿足醫院資訊系統對異質性、互用性、資料共享、及資訊整合的要求，一種新型的軟體整合系統架構應運而生--服務導向架構(Service Oriented Architecture, SOA)。因此本研究以 SOA 技術，實作了一個結合以基本網格為後端的醫院平台--醫療網格(MedicalGrid)，除了提供強大的運算服務和監控網格環境服務，且適用於跨多個網格計算環境上的資訊擷取服務，為資源經紀人(Resource Broker)的其他元件做資訊收集，還可提供使用者出院後的醫療及健康照護。

關鍵詞：資訊科技、服務導向架構、跨網格、資源經紀人

Abstract

As the evolution of information technology goes so quickly and the network technology spread so far, those healthcare scenarios only appeared in movies are coming toward us. With the E-hospital rapid growth, various types of information systems develop and use unceasingly. However, in the hospital information system of development methods under their own way, sharing of information between the systems is difficult day after day. In order to satisfy the hospital information system to the heterogeneity, interoperability, information sharing, and information integration, a kind of new software system architecture to came with the tide of fashion -- Service-Oriented Architecture (SOA) [1, 2, 7]. Our research take the SOA technology, combined with the grid based of the hospital for back-end platform—MedicalGrid [3, 8, 16, 18], provides a powerful computing capabilities, and fit with a Cross Grid Information Service (CGIS) [3] that enables Resource Brokers [11, 12, 13, 14] to get information from cross grid environments for other components. It can also provide medical care after discharge from hospital and health monitoring.

Keywords: Information Technology, Service Oriented Architecture, Cross grid, Resource Broker.

Acknowledgements

I would like to thank all of these people who have supported and helped me through the completion of this tiesis and all of my working for the business of our lab. In particular, I would like to thank my advisor, Dr. Chao-Tung Yang, who introduced me to SOA technology and give me the broad support and guidance. I would also like to thank Professor Yi-Min Wang, Professor Chao-Chin Wu, Professor Ching-Hsien Hsu and Professor Kuan-Chou Lai for their help for their valuable comments and advice while serving on my reading committee.

There are many people whom I would like to thank, especially my group-mate who help me construct experimental environment. And all the members of HPC lab, they have give me a lot of support of completing my thesis. I would like also thanks the people who support and encourage me for this thesis. Thanks for your help and considerations.

Finally, I would like to thank my family and all of my friends. Because of your unconditional support I could made this thesis complete.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The introduction of the National Health Insurance has gradually increased the competition in the medical market. Taiwan people for medical treatment financial may be the greatly progressive enhancement. But the distribution of medical resources there is a significant imbalance [6]. Many people in country sides need to have a longer time to take their medical treatments, thus delays the time for treatment.

Face in the quality of medical care services more and more better and cost considerations, complete patient care and enhance the quality of medical be taken seriously. In order to achieve personalized health management and maintenance the rights of the people to know. This thesis develops a grid-based medical resources sharing platform, which provide medical services to the populace. Aims at separately by the home viewpoint (patients) and the hospital viewpoint (doctors) provide the different service.

To the home point of view, provide a standard platform (middleware) to integrate the health care services when the patient being out of hospital. In order to achieves the goal of health management, which can help patients take the follow-up treatment. To the hospital point of view, this thesis use SOA technology to integrate Grid & Medical (MedicalGrid) systems, to provide physicians and patients interactive with each other. The health care services which can provide originally the hospital extended to individuals and families. In the process, we discuss with a number of attending physicians, nurses and related medical experts many times. The collection of clinical practice and views on professional advice, to sum up the views of staff to the general public demand for health services management as a major research direction will be produced after the system prototype design and development system.

1.2 Contribution

1.3In view of this, ours thesis was use the Service Oriented Architecture (SOA) and grid technologies come to develop a basic Healthcare Service Grid Infrastructure. It contains one of the main Grid Resource Management and Information Service, service platform and the entrance of health information user interface (Service Portal), HL7 [4, 9, 10, 17] Data Management System, Health Services API, and Data Replica and Parallel Download Management parts. We also propose a SOA solution to present a number of incompatible information systems, so that the services can be integrated and combined to each other. We implement a web portal to provide healthcare services and WSDL related documentations. We provide access right control mechanism to provide user more secure.

On the hospital's existing information systems architecture for service-oriented transformation, will present a number of incompatible information systems, so that they can link to each other. In order to achieve information technology and clinical operations of the synchronization between the integration to help streamline the hospital for medical treatment flow and information sharing to enhance the quality of medical services.

1.4 Thesis Organization

The remainder of this research is organized as follows. Background review and studies are presented in Chapter 2. Our new approach is outlined in Chapter 3. Design and implement of our scheme are presented in Chapter 4. Chapter 5 concludes this research article.

Chapter 2

Background review and related work

2.1 HL7

Since a variety of time and space factors, the general public may be at different hospitals for treatment, and various medical institutions to use the hospital information system (HIS) is not necessarily the same as a result of patient medical information can not be at each inter-hospital transfer, often creating the need for patients to check the duplicate, or duplicate medical acts, resulting in unnecessary waste of medical resources. By the HL7 Organization for medical information exchange standards, the role of medical information is structured, standardized so that they can through the network, at different medical institutions between the transmissions, to achieve the purpose of information sharing.

Health Level Seven (HL7) [4, 9, 10, 16] is one of several American National Standards Institute (ANSI) -accredited Standards Developing Organizations (SDOs) operating in the healthcare arena. Most SDOs produce standards (sometimes called specifications or protocols) for a particular healthcare domain such as pharmacy, medical devices, imaging or insurance (claims processing) transactions.

Health Level Seven's domain is clinical and administrative data. HL7 standard interface to pass through different borders and cultural messages to promote health messages also have permeability and dilated. HL7 Organization founded in 1987. Its main purpose is to develop various types of medical information systems, such as clinical, banking, insurance, management, administration and inspection, such as electronic data standards.

HL7 protocol brings together different vendors application software used to design the interface between a standard formats, which allows the various medical institutions of different application systems, carry out some important information to communicate. The protocol design while retaining a considerable degree of flexibility, making some information on the specific needs of a deal to maintain compatibility. The HL7 Organization reference to ISO (International Standards Organizations, ISO), the use of open systems architecture (Open System Interconnection, OSI) communications model as shown in **Figure 2.1.** The HL7 is satisfied for the highest level, that is, the application layer. It provides norms such as: the relevance of the classification, the emergence of an effective inspection, the structural mechanism for the exchange of information and consultation functions.

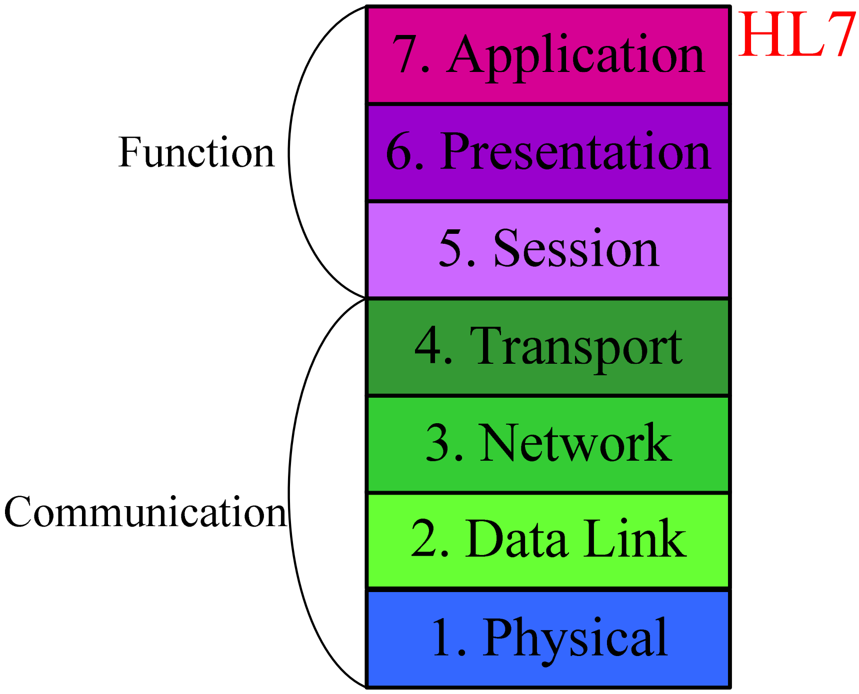


Figure 2.1: ISO-OSI Communication Architecture Model.

Taiwan health information exchange agreements seventh layer Association (HL7 Taiwan) was established in 2001, and at the Department of Health under the guidance of continuing education to organize various activities such as training and seminar, and actively promote the HL7 standards in the domestic [5].

2.2 Web Service

Web Services [9] is a software component, which accord the open standards (such as HTTP, XML and SOAP, etc.) and data formats for other applications to provide services. There are two key points: first, it is a component of services provided. Second, it is Web-based on open standards.

Provide services as components, it can be used to construct distributed systems architecture, distributed architecture achieve dynamic integration and balance the burden of the advantages of upgrading unit.

The use of Web-based on open standards, has been widely used in the Web onto the operation of the network architecture, the adoption of open standards so that Web Services have good interoperability on different platforms using different programming languages to build systems that can also be easily integrated overcome the current decentralized system, each using a different mechanism of integration caused by difficult circumstances.

The basis of Web Services include: XML, WSDL, SOAP, UDDI. Its underlying architecture model of the operation steps are as follows as the **Figure 2.2**: to XML format for the benchmark data into a Web Services data, the use of WSDL descriptions will serve to do a descriptive, so that the other side of could be a description of this special, to interpret the information gathered. SOAP communication to the ground floor, the motion transmitted to the UDDI search action or Register. We can see, WSDL, SOAP and UDDI are used to describe the XML method, followed by the detailed description of the four basic Web Services in general is to provide the Web Services to register to UDDI, much on the list of Web Services, we can see him one of the WSDL document.

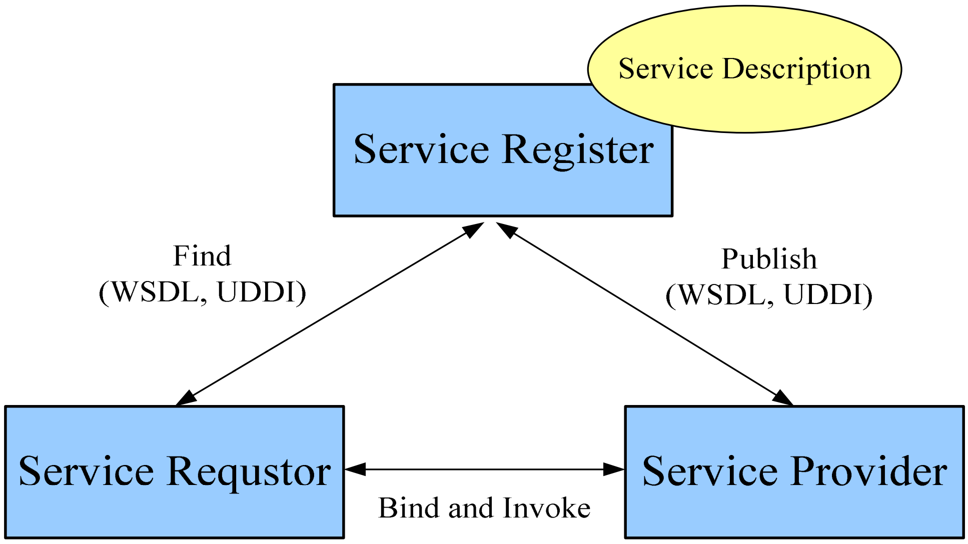


Figure 2.3: The key elements of the SOA paradigm.

- SOAP (Simple Object Access Protocol)

SOAP [24] is the Web Services used to transfer data encoding format. W3C has developed the SOAP 1.2 specification currently. The main purpose of SOAP is to provide the pur RPC (Remote Procedure Call) function. SOAP mainly followed XML standard to develop a message format. A complete packet of SOAP will be used the "Header + Body" to mark the part of the contents of each message. In addition to using HTTP, SMTP and other network protocol for data binding and exchange.

There is no relevant with SOAP standard and the underlying network protocol. Therefore, when we accorded SOAP specifications to produce SOAP documents, regardless of the use of any network protocol can be exchanged SOAP documents. Then both sides can be selected in accordance with the protocol to exchange information, shown in **Figure 2.3.**

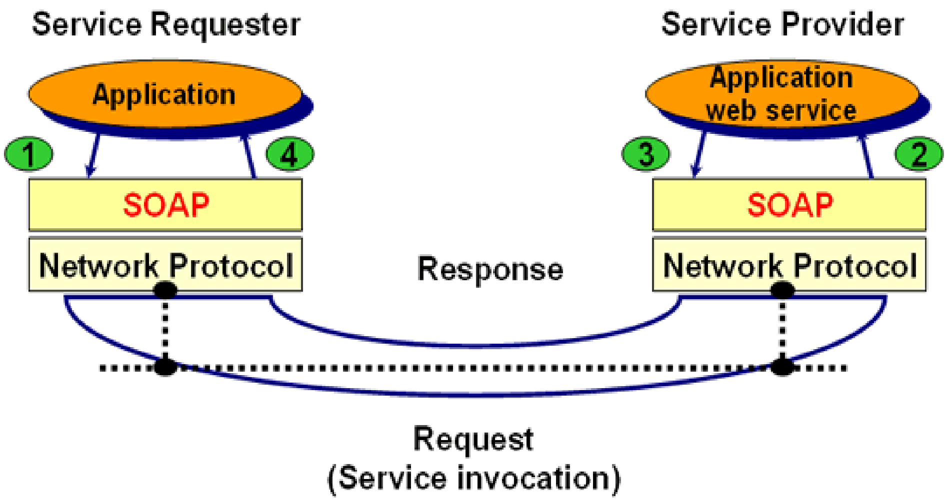


Figure 2.2: SOAP Operation mode.

- WSDL（Web Services Description Language）

  WSDL mainly used XML standard language to describe the details of Web Services, let Web Service applications can follow a standard method to describe what the capacity of its own. So that the interaction is more easily. If you want to know Web Services, the first step is to try to understand the WSDL definition.

- UDDI（Universal Description, Discovery and Integration）

  UDDI refers to a directory on the Web Services Registration Service, and its architecture is also based on XML document. Its main purpose is a provider of Web Services through UDDI provider to the other person has to provide Web Services, therefore the function of UDDI also similar to a telephone directory, or known as Yellow Pages, the purpose is to quickly inform service users can use him there is what the Web Services.

When we understand the use of SOAP, UDDI and WSDL components, **Figure 2.4** is shown that the relationship between each other and how to use. First of all, the user will find the services they needed from UDDI. UDDI will give the WSDL address which from service provider to the client. The client will use this WSDL file to find the actual location of the service. Then, the service provider sends the service details of the services back to the user. The above stages are at the design stage or dynamic execution stage. The client will be used to develop tools binding services that the WSDL document provided. Both of them will be bind and will begin to use SOAP to do data transmission. This stage is the actual process of using Web Services. We can see that all protocols (HTTP, SOAP, UDDI, WSDL, XML) were used the cross-platform, cross-language standards. Client and server even if the two sides use different platforms or development environment can't affect the use or provision of services.

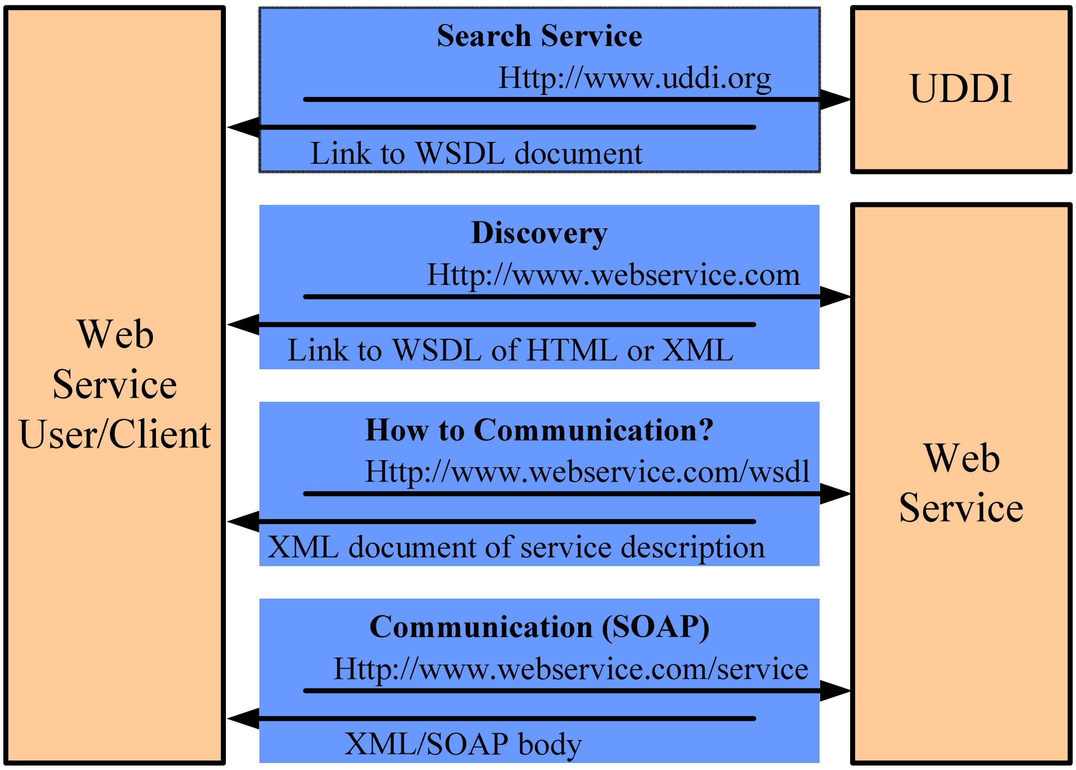


Figure 4.2: SOAP Operation model.

2.3 Service Oriented Architecture (SOA)

A Service-Oriented Architecture (SOA) [1, 2, 7, 19] is a group of services that communicate with each other. The process of communication involves either simple data-passing between a service provider and service consumers, or a more complicated system of two or more service providers. Intercommunication implies the need for some means of connecting two or more services to each other.

SOAs build applications out of software services. Services comprise intrinsically unassociated, loosely coupled units of functionality that have no calls to each other embedded in them. Each service implements one action, such as filling out an online application for an account, viewing an online bank-statement, or placing an online booking or airline ticket order. Instead of services embedding calls to each other in their source code, they use defined protocols that describe how one or more services can "talk" to each other.

A software developer, softwa re engineer, or business process expert associates individual SOA objects by using orchestration. In the process of orchestration, a software engineer or process engineer associates relatively large chunks of software functionality (services) in a non-hierarchical arrangement (in contrast to a class hierarchy) by using a special software tool that contains an exhaustive list of all of the services, their characteristics, and a means to record the designer's choices that the designer can manage and the software system can consume and use at run-time. The new generation of IT architecture should provide the platform to set up with the following characteristics of application services: (1) loosely coupled; (2) location transparency; (3) Agreement independent.

In response to such change in the trend of IT architecture, SOA are from Gartner's IT architecture was first put forward the concept of an enterprise IT infrastructure must be flexible enough to response to business needs with flexibility, such as the telecommunications industry may at any time for various services, adjust billing methods. As in recent years, an emerging model of software architecture, SOA concepts are the main business needs for a hospital or a combination of a group of software components. Combination of elements typically include: software components, services and flow. When the hospital or enterprises face an external request, the external flow is responsible for the definition of the requirements of treatment steps; Services of specific steps, including all program components and software components are responsible for the implementation of programs. SOA has become a modern software development technology, through the SOA allows easy integration of heterogeneous systems, the program also increased the degree of re-use. Do not have to own or have all the program components, developers can be required depending on the best combination of network services. Not limited to specific vendor product functionality or platforms, to achieve genuine openness. The main purpose of the SOA is to integrate the four items as follows: (1)The work process integrative: To integrate the different organizations process and work together. (2)The application integrative: The program module could be reused easily through the Internet. (3)The information integrative: To speed the up business workflow process and show the business domain concept. (4)The identification integrative: To control the business activity by identification. An effective services-based approach can enable both business and IT benefits that include:

1. Distributed: SOA components are from the much scattered in the network system from the combination may be local area network, or it may be from the WAN. For example, web services technology is the operation of HTTP links to each other SOA. This approach also allows web services technology quickly became the Internet all the support systems are able to use the platform technology.
2. Loosely Coupled: SOA can provide the service consumer a standard service interface which will not be affected by service provider change the service. It can increase the system scalability, the system component can be changed

and replace on demand. The service requester does not need to know where is the service provide and detail, they just need to invoke the service to communicate with each other. The enterprise can be better flexibility and extensibility at lower cost through re-usable, self-contained software modules.

3. Open standards: One of the core features in SOA is open standard; it can reduce integration and application development complexity through the use of standards. It different form the software such as CORBA, DCOM, RMI and J2EE which can not support composite applications and agile or on-demand computing drives inter-operability.
4. Process centric: At the beginning of constructing a system. First of all, it is necessary to understand the business process including data input and output. The programmer can choose and design the proper service to accomplish work.

Service Oriented architecture (SOA) software to change the traditional pattern of development, through the SOA and Web Service, makes the user is no longer confined to a single operating platform, be able to achieve more effective cross-platform operating environment, as well as efficient methods of service integration are future development goals.

Chapter 3

MedicalGrid

3.1 MedicalGrid

MedicareGrid, a grid based E-health system, to fulfill a practical future medical scenario for next twenty years. It uses grid and peer-to-peer technology to integrate all the computing and storage resources and develop medical data exchange mechanisms base on it. Those data form a data warehouse to support the data mining techniques that extract valuable information for medical decision support and future research.

MedicalGrid is to establish from individuals, families, hospital physicians to complete medical service mechanism, in order to improve the way we have to be treated in a hospital nowadays. It can not only take care of the old, handicapped persons or anyone who need long-term medical care but also decrease the opportunity of accidents and prevent from cardiovascular diseases taking place. Moreover, it would reduce the expense of medical insurance system and create a better life for all human beings.

This system construct of one medical data exchange and sharing platform. Allow any person in any place for medical treatment, keep to have a complete patient record. To help physicians predominate the patient's medical history in order to provide complete medical services. It uses data mining technology and methods to provide the provision of medical decision in the grid platform. We depend personal medical information to carry out follow-up medical (such as medication, medical diagnosis, care, etc.) analysis, to reduce medical expenditures. To medical quality assurance, RFID technology will be integrated into the emergency care, surgery safety, patient safety and medication management. According to community as a unit, the development of wearable physiological signal flow measurement system.

Nowadays the application of grid computing instrument, and did not have a set of comprehensive use of a mechanism for users to more easy way, just in their own computer to install the public on the widespread use of the software, you can network through the Internet on the specific scope of Grid resources to carry out access, efficient computing and control platform. First of all, we want to develop the Resource Broker will make use of Globus Toolkit [15] as the future of the Grid software agent. The main objective of this program is to develop a Grid Monitor, used to monitor the computing platforms on all available computing resources of the current status, such as machine load on CPU, Memory size available, the node between the current instantaneous frequency wide, and the network flows in each node shown in **Figure 3.1** and job monitor shown in **Figure 3.3**. This Grid Monitor is NWS (Network Weather Service) [25, 26] as the main core, by the NWS is responsible to retrieve the information (bandwidth, flow rate, CPU load, etc.), and then use RRD Tool with MRTG to carry out mapping to produce page, supply users browser and Jane transaction. **Figure 3.2** show the Ganglia snapshot from grid nodes. Then a set of questions future prediction model at the grid on the network flow and execution time for job applications.

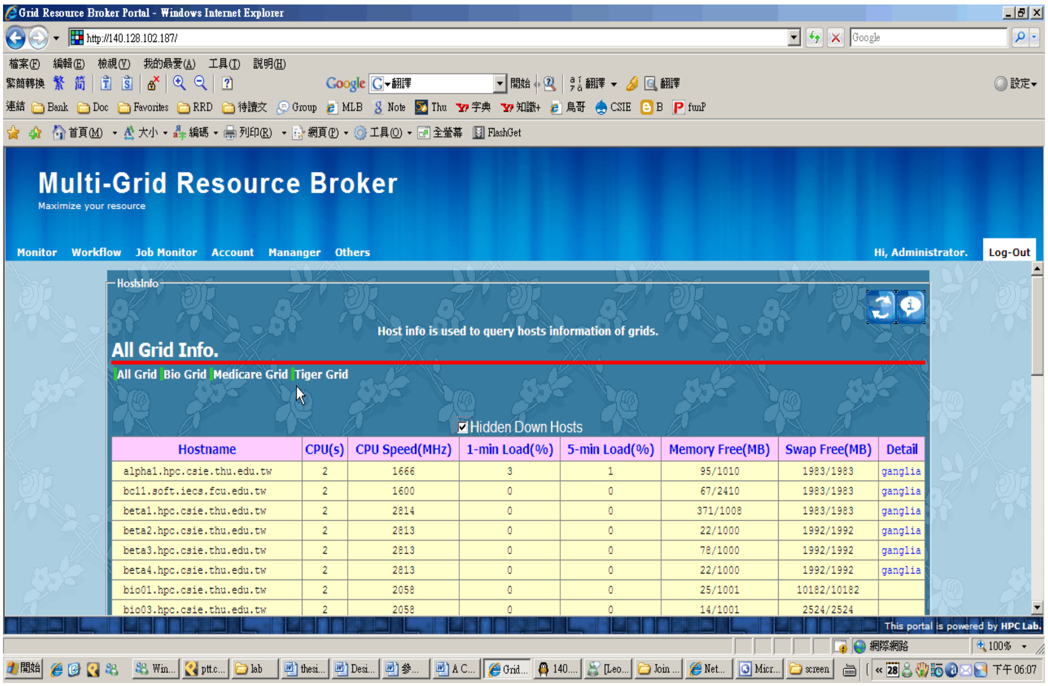


Figure 3.1: show the status of grid nodes.

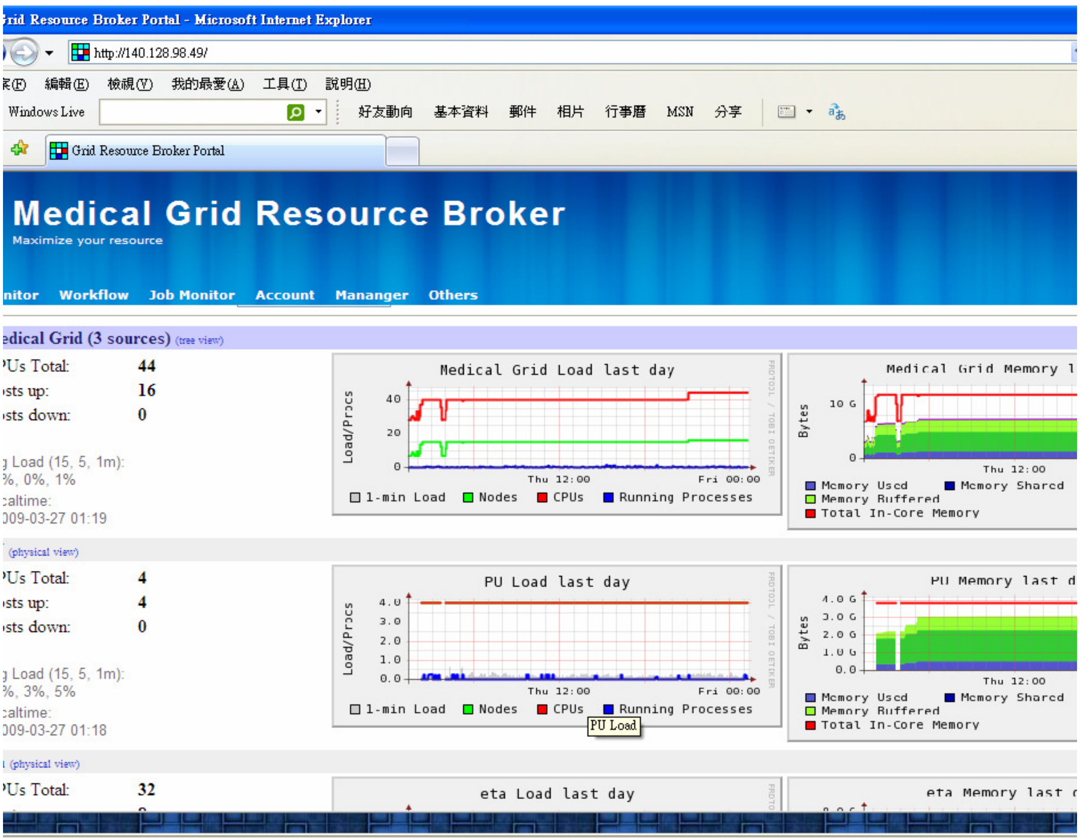
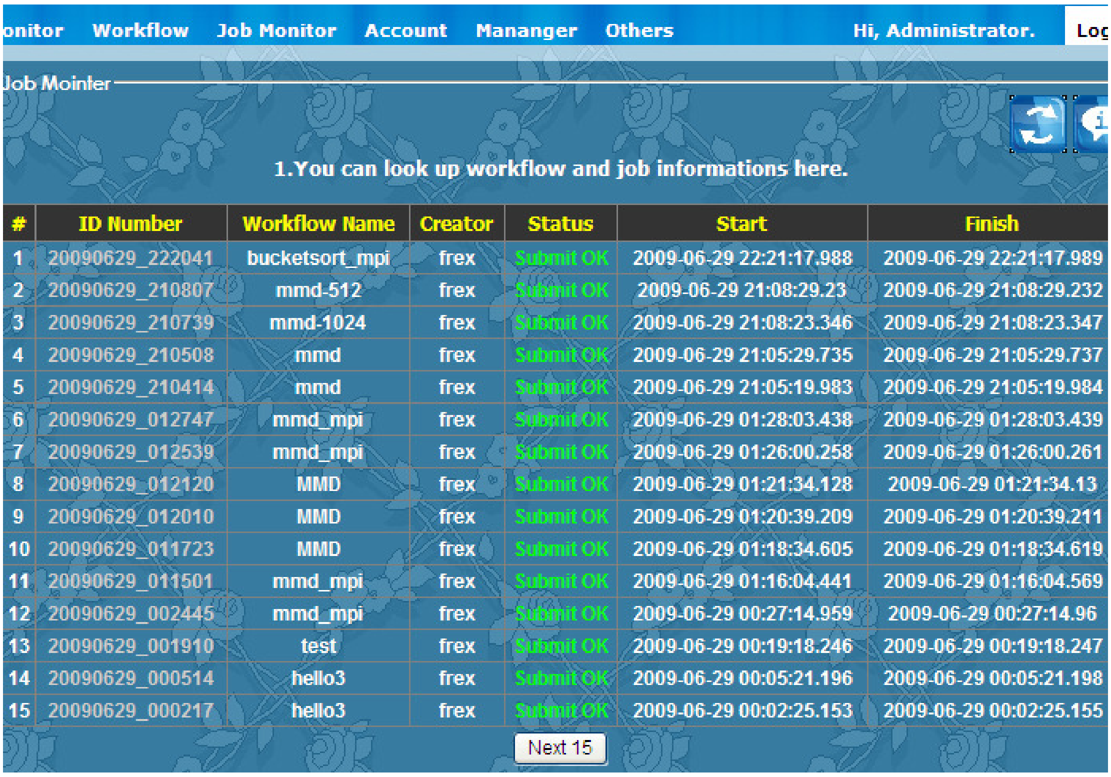


Figure 3.2: Show the Ganglia snapshot from grid nodes.



#	ID Number	Workflow Name	Creator	Status	Start	Finish
1	20090629_222041	bucketsort_mpi	frex	Submit OK	2009-06-29 22:21:17.988	2009-06-29 22:21:17.989
2	20090629_210807	mmd-512	frex	Submit OK	2009-06-29 21:08:29.23	2009-06-29 21:08:29.232
3	20090629_210739	mmd-1024	frex	Submit OK	2009-06-29 21:08:23.346	2009-06-29 21:08:23.347
4	20090629_210508	mmd	frex	Submit OK	2009-06-29 21:05:29.735	2009-06-29 21:05:29.737
5	20090629_210414	mmd	frex	Submit OK	2009-06-29 21:05:19.983	2009-06-29 21:05:19.984
6	20090629_012747	mmd_mpi	frex	Submit OK	2009-06-29 01:28:03.438	2009-06-29 01:28:03.439
7	20090629_012539	mmd_mpi	frex	Submit OK	2009-06-29 01:26:00.258	2009-06-29 01:26:00.261
8	20090629_012120	MMD	frex	Submit OK	2009-06-29 01:21:34.128	2009-06-29 01:21:34.13
9	20090629_012010	MMD	frex	Submit OK	2009-06-29 01:20:39.209	2009-06-29 01:20:39.211
10	20090629_011723	MMD	frex	Submit OK	2009-06-29 01:18:34.605	2009-06-29 01:18:34.619
11	20090629_011501	mmd_mpi	frex	Submit OK	2009-06-29 01:16:04.441	2009-06-29 01:16:04.569
12	20090629_002445	mmd_mpi	frex	Submit OK	2009-06-29 00:27:14.959	2009-06-29 00:27:14.96
13	20090629_001910	test	frex	Submit OK	2009-06-29 00:19:18.246	2009-06-29 00:19:18.247
14	20090629_000514	hello3	frex	Submit OK	2009-06-29 00:05:21.196	2009-06-29 00:05:21.198
15	20090629_000217	hello3	frex	Submit OK	2009-06-29 00:02:25.153	2009-06-29 00:02:25.155

Figure 3.3: Job monitor – All Workflows

Chapter 4

System Design and Implement

4.1 Architecture

The system software stack includes three layers constructed using a bottom-up methodology as shown in **Figure 4.1**. The layers are described below:

- Bottom Layer: principally consists of Nodes. The layer contains two main blocks: the Information Provider, which uses Ganglia to gather machine information on Nodes, such as number of processors/cores, processor loading, total/free memory, and disk usage, and Network Weather Service (NWS), which gathers essential network information such as bandwidth and latency. The second block contains Grid Middleware, used to join Grid Nodes together, and the MPICH-G2 required for running parallel applications on the Grid.
- Middle Layer: we use SOA technology to build a middleware, combined with Resource Broker, MedicalGrid and P2P network. User can invoke services and resources that MedicalGrid and P2P network provided.
- Top Layer: when user login to SOA web portal, it will accord your privilege what services you can apply.

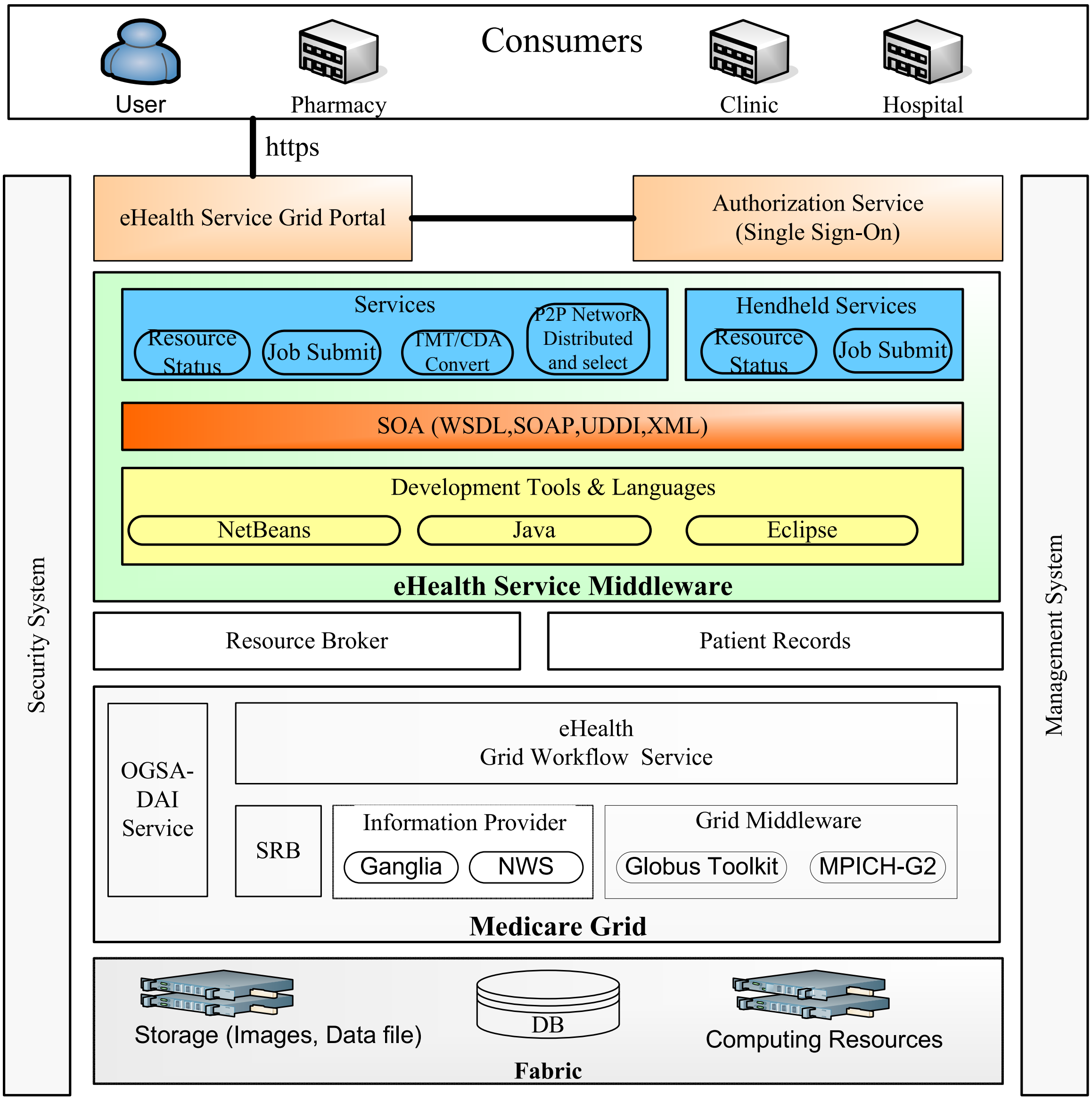


Figure 4.1: System Architecture.

4.2 Resource Broker

The main layers of our system architecture include SOA web portal, Services, Grid portal, Resource broker, Grid nodes.

We use a Resource Broker for Computational Grids. It discovers and evaluates Grid resources, and makes job submission decisions by comparing job requirements with Grid resources. The Resource Broker system architecture and the relationships among components are shown in **Figure 4.2**. Each rectangle represents a discrete system component.

The Resource Broker's primary task is to compare user requests and resource information provided by the Information Service. After an appropriate assignment scheme is chosen, Grid resources are assigned and the Scheduler submits the job for execution. The results are then collected and returned to the Resource Broker, which records them in the Information Center database via the Information Service Agent. Users can view the results through the Grid portal.

The system architecture includes five layers: SOA web portal, Service, Grid portal, Resource broker, Grid nodes. The following sections will be described in detail.

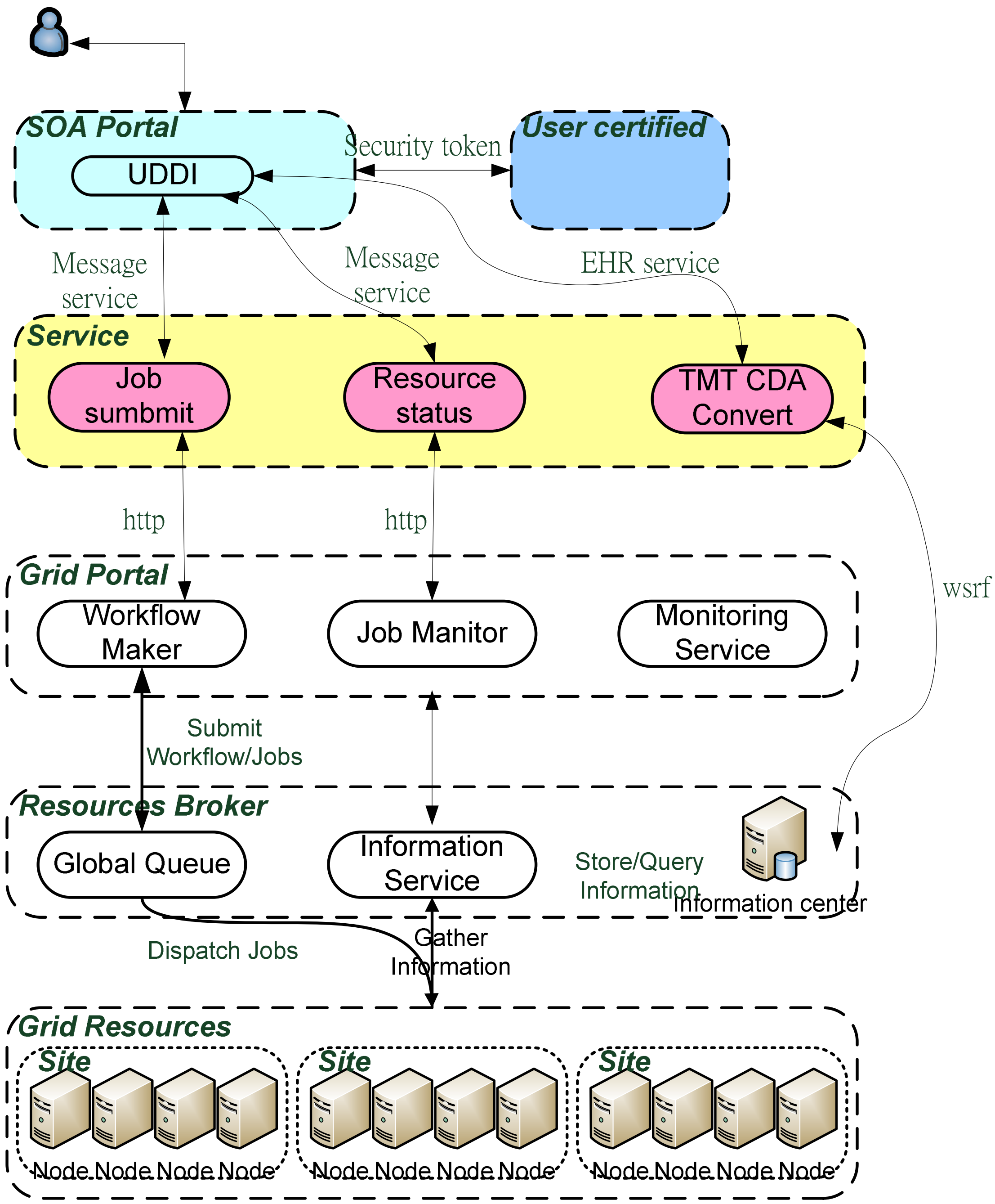


Figure 2.3: Resource Broker system architecture

4.3 SOA Web Portal

We propose access right control mechanism into our portal to provide user more secure way to utilize our healthcare system. Unauthorized user is only capable of seeing the service information and health and medical information without service invoke, as shown in **Figure 4.3.** Our web portal lists the WSDL documents shown in **Figure 4.5**.

While user login to web portal, the user account, password are verified by our authorization server (access from Identity Server). Then user can get a security token from authorization server. The following steps describe Security Token System (STS) token issuance and request/response process shown in **Figure 4.4**:

1. The client initializes and sends authentication request to the STS**.**
2. The STS validates the client's credentials**.**
3. The STS issues a security token to the client**.**
4. The client initializes and sends a request message to the service**.**
5. The service validates the security token and processes the request**.**
6. (Optional) The service initializes and sends a response message to the client**.**

Each account can have more than one security tokens which provide different service to register. The account privilege is defined by their virtual community (friends and family). Unauthorized user is only capable of browsing and seeing the information. Authorized user also can browse private dates (ex: patient records), using the services (ex: telehomedcare of diagnosis and inspection, submit job by making workflow, monitor grid node information and modify own account information). Administrator user can use all functions of authorized user (but can not browse private dates of patients). The additional function of administrator is cleaning of job records and image files.

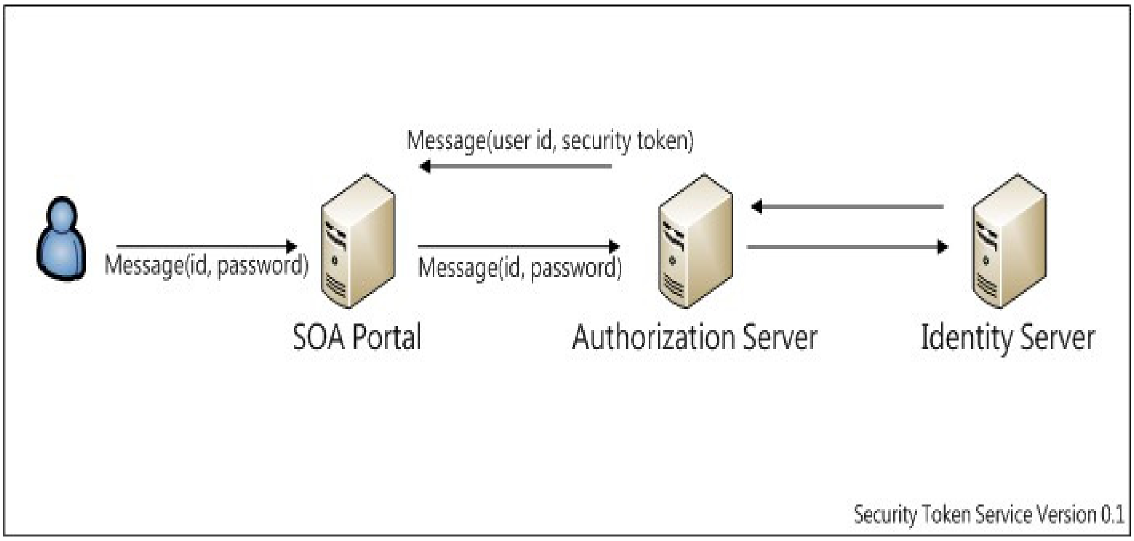


Figure 4.4: User login and Single Sign-On Model.

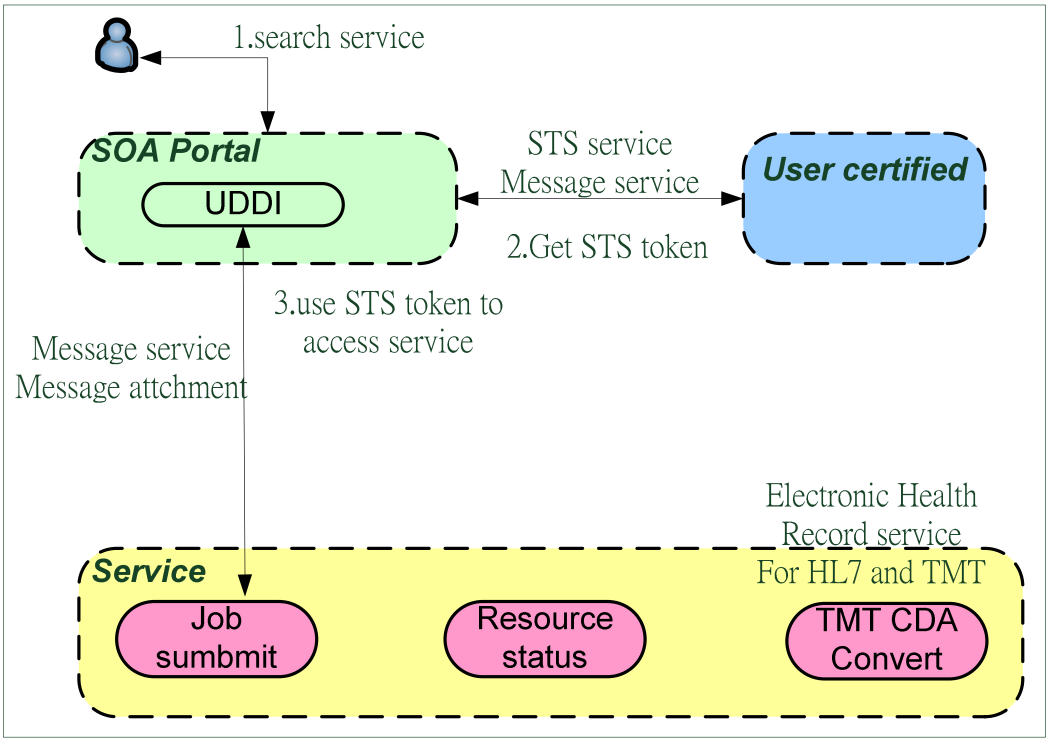


Figure 4.4: Security token service operation model.

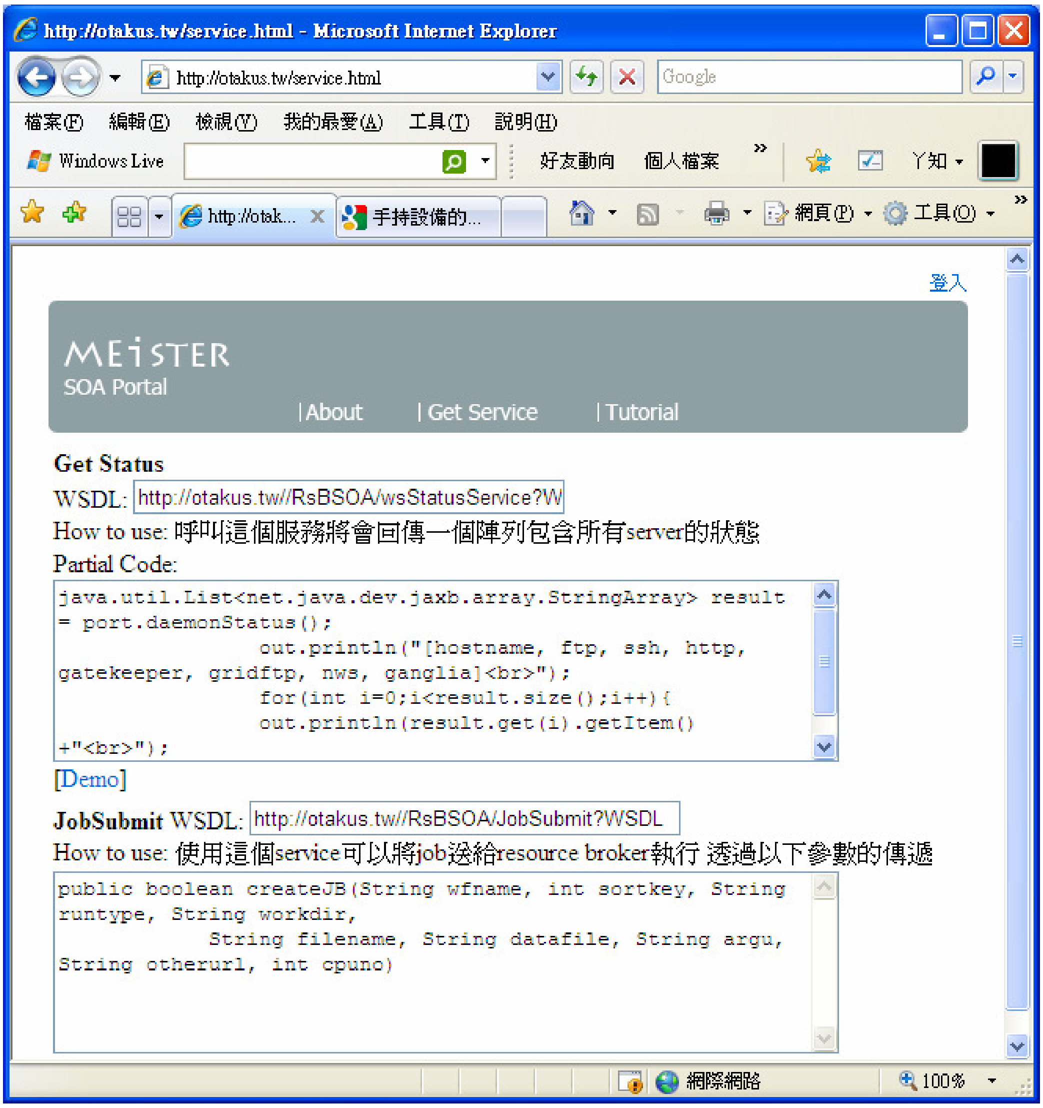


Figure 4.5: The services on the web portal.

4.4 SOA Technology Combination Resource Broker

SOA combination of elements typically include: software components, services and flow. When the hospital or enterprises face an external request, the external flow is responsible for the definition of the requirements of treatment steps; Services of specific steps, including all program components and software components are responsible for the implementation of programs. We use the characteristics of SOA, and resource brokers (Resource Broker) for the combination of resource brokers go through the medical access grid (MedicareGrid) system. The High-level interactions between various entities, components, resources, and human participants and the channels of control as shown in **Figure 4.6** is described below:

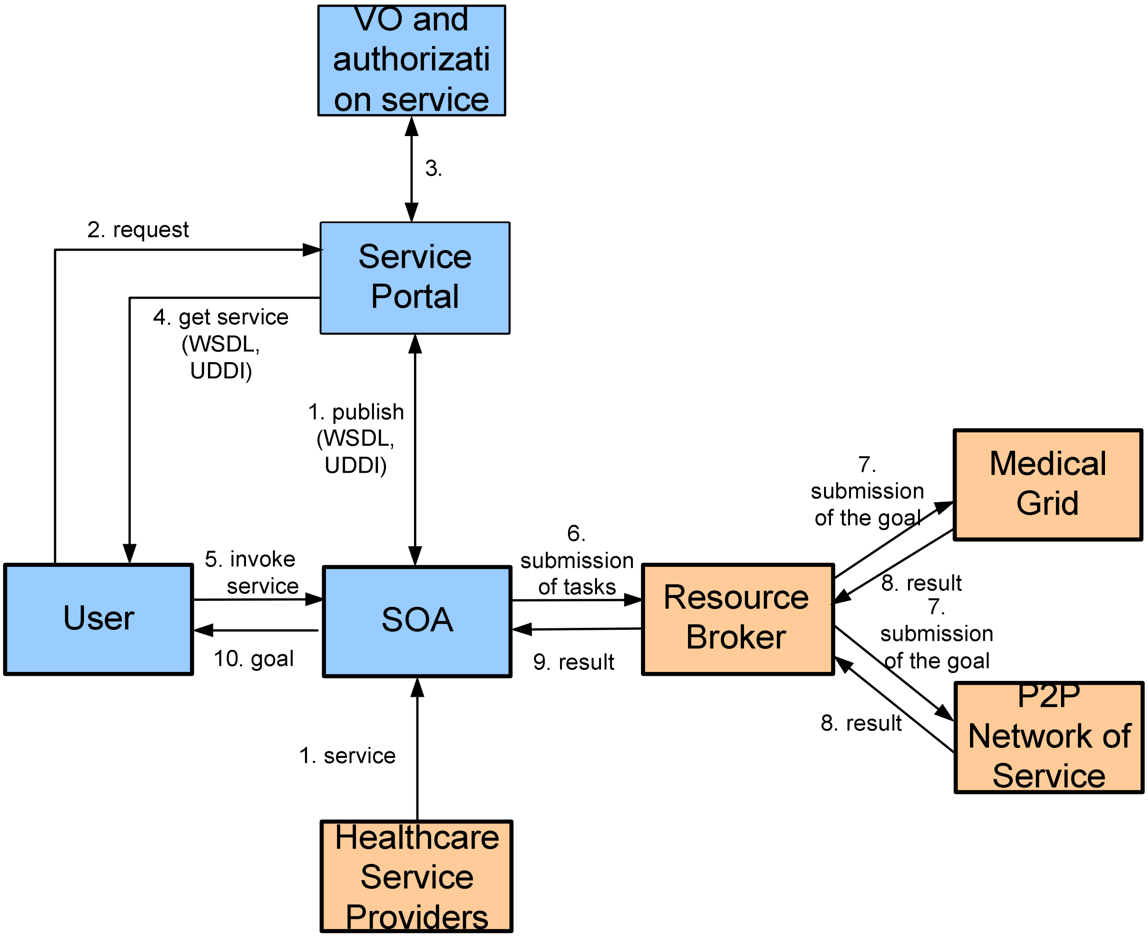


Figure 4.6: The flow chart of SOA operation

1. SOA components will be imported from time to time updates on web services, the service's Web Services Description Language (WSDL) to release to be published on the SOA web portal for users to search.
2. User initiates session with portal by selecting modeling resources required and 'planning' experiment with the resources. .On the SOA web portal, the user can search for suitable services and resources.
3. User obtains authorization through VO and authorization service.
4. User to get the service of WSDL document.
5. WSDL of the service allows users to be connected to the Internet through the SOA components, and will invoke service.

6. SOA portal through the SOA technology contacts Resource Broker to request resources (remote/local services or database). MedicalGrid and P2P network. SOA component will take submission of tasks to resource broker.
7. Resource broker submitted the tasks to MedicalGrid or P2P network in order to achieve the goal.

8-10. Results are visualized on the users' desktop through the Web browser.

4.4.1 Services

We identify the following modules in our thesis:

1. Resource Status: The main function was provided users to view the machines status when user create job, then they can choose the correct machines. The user chooses one or more nodes and sends a request to the Information Service, which finds the information and returns it in a sorted list, as shown in **Figure 4.7**. Information type include hostname and machine status.

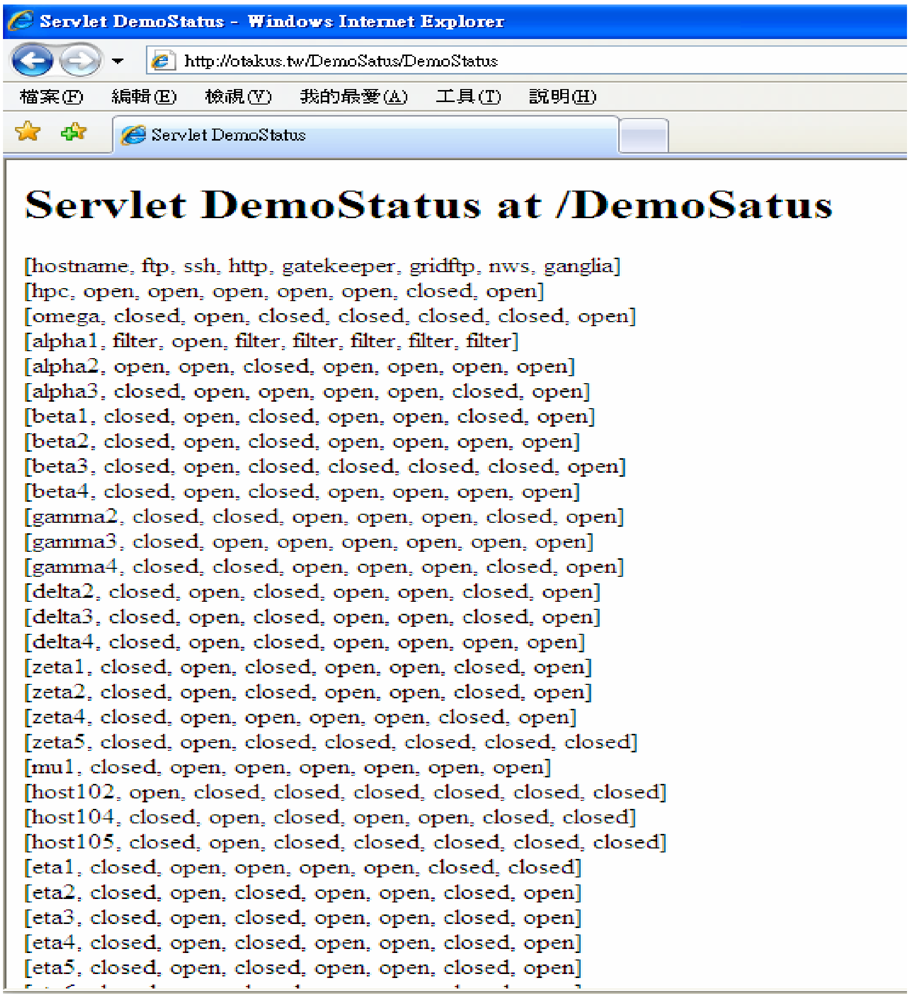


Figure 4.7: The return of machine status.

2. Job Submit: The users can submit job into the Workflow Engine by the SOA middleware, and then will be allocated to the grid environment for computing machines by the resource broker (RB). **Figure 4.8** show the user use SOA portal to set parameters for a job. Then the broker return “true” denote job submissions success as shown in **Figure 4.9.** The user can see workflow ID, name, status, and running time in MedicalGrid portal shown in **Figure 4.10.**

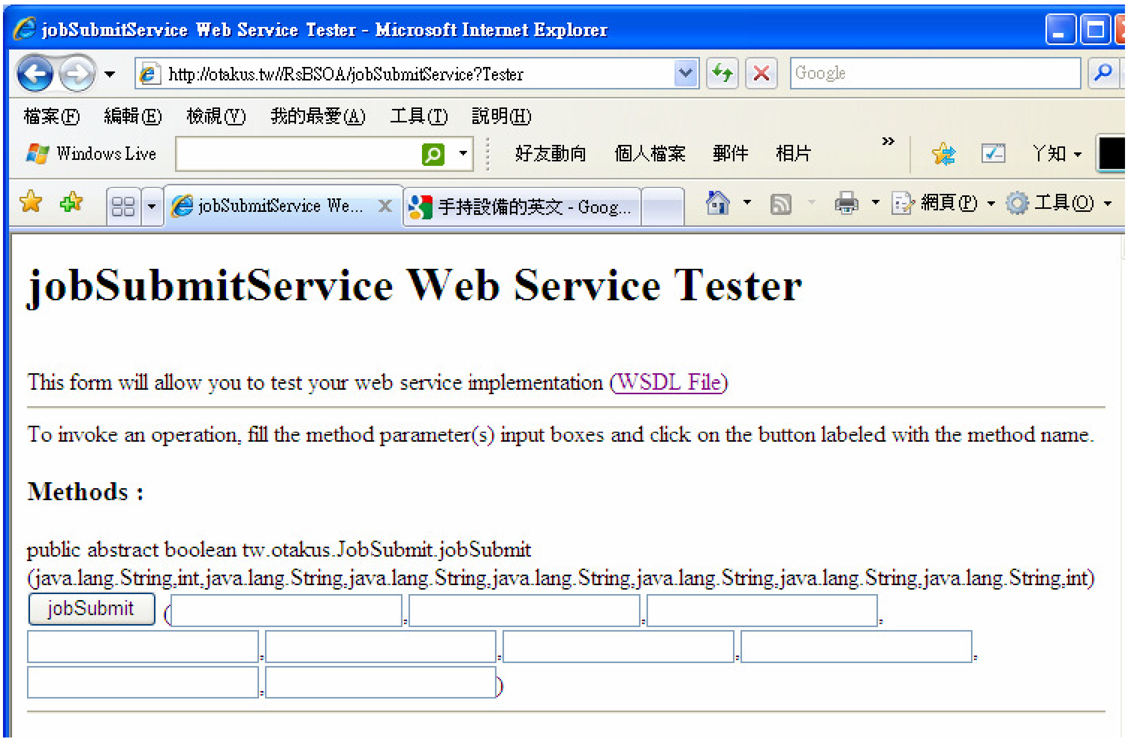


Figure 4.8: The job submit service.

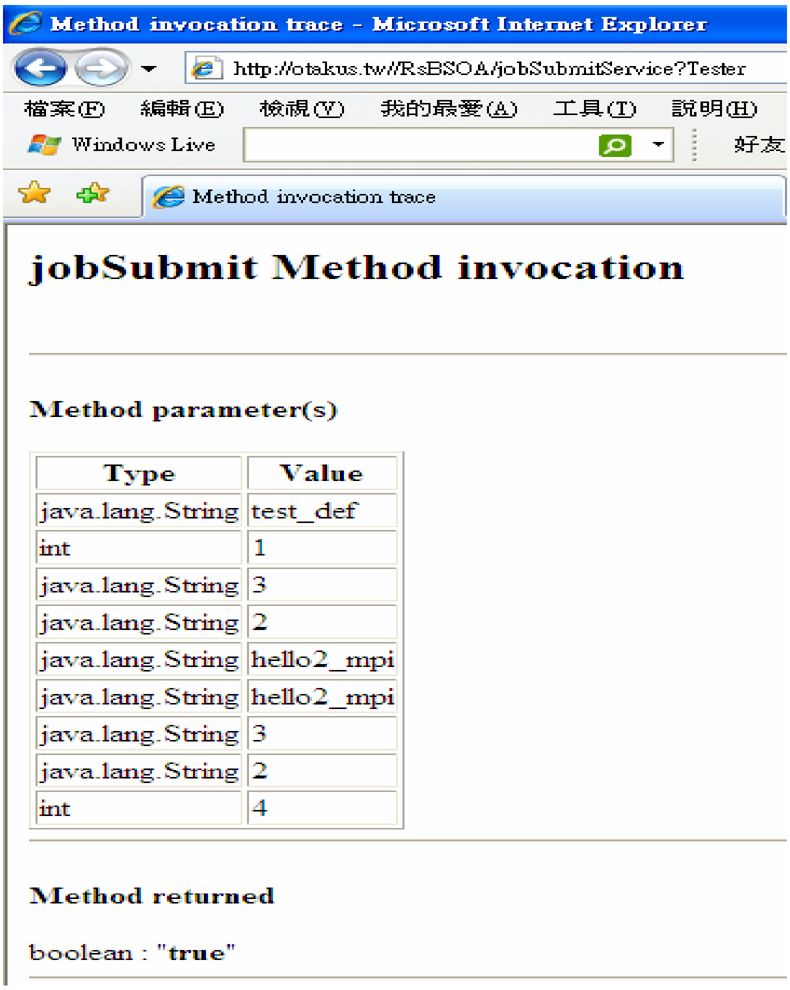


Figure 4.9: The return of submit OK.

Figure 4.10: Job monitor – Submit OK.

3. P2P Network Distributed and Selected: each peer in the P2P network provided services each other. They cooperative composition in the space of services for discovering and composing some services that solve the problem. Each node in the network contributes to discover the peers that can originate useful compositions. According to the P2P model, peers become a crucial part of the architecture, since with this model the network lacks of structural components for discovery and composition. Each peer is responsible of receiving requests from other nodes (goals), and fulfilling them (i) by relying on service operations or lower level features available on each peer or (ii) by forwarding the request to other known peers. All services can register on our SOA web portal. SOA web portal will list all WSDL of services. User connected to our web portal to find services they wanted. The P2P nodes communicate each other to complete the

job.

4. TMT/CDA Convert: the general public of the personal health record information, because of differences in medical situations may be scattered in a variety of medical institutions; to make the information on to the network for the exchange and integration, we need the same information in a standard format will be edited. We use international standards as the Health Level 7 as the exchange of information integration.

XML-based Grid Information Management System (XGIMS) [21, 27] presents Index-based Query Engine (IQE) that supports powerful, flexible and efficient query by speeding up document parsing and query process. IQE contains two components: Index-based Parser (IBP) and Index-based XPath Engine (IXE).

The experiments have been carried out on a PC with dual 2.4GHz Xeon processors, 1GB RAM, Redhat Linux9, and Jdk1.4.2 with 256M JVM maximum memory. First, we create some XML documents contained different amount of elements. The amount of XML documents may seem to be very little (only 7), but it is selected from more than thirty documents we have tested and enough to show the change of performance while size increasing. Then, we make quantitative analysis of the costs of parsing time, using MSXML4.0 and IBP respectively.

The XML documents are parsed for 1,000 times without grammar caching. The average parsing time can approximately reflect the performance of the parser shown in **Figure 4.11**.

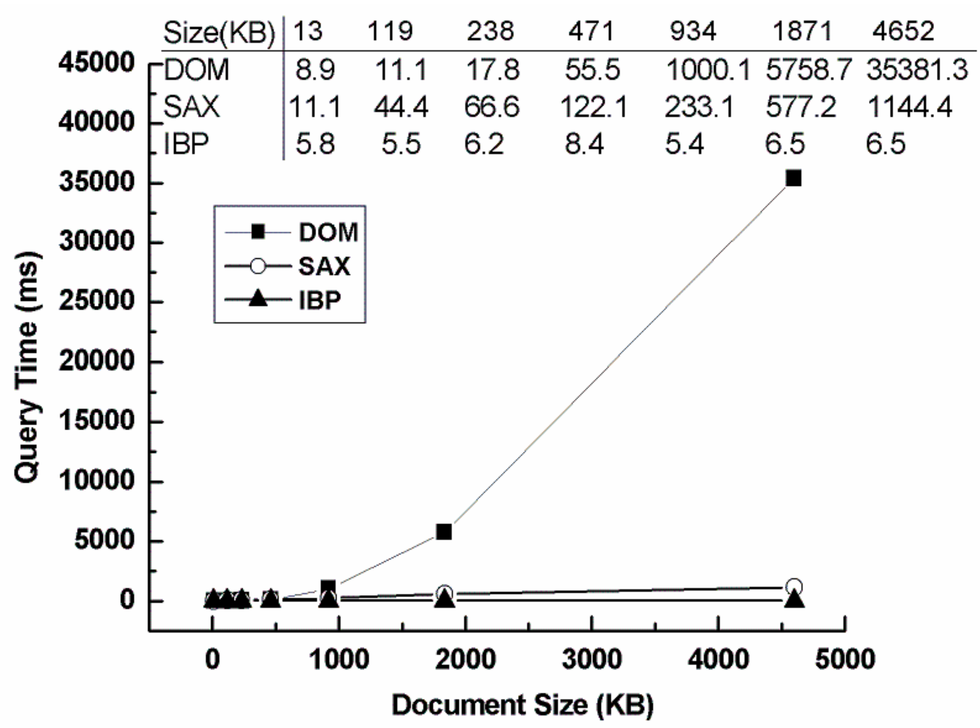


Figure 4.11: Average parsing performance (1000times).

DOM parser performs pretty well on small XML documents, but cost for large documents exceeds the limit of our tolerance. SAX parser performs better than DOM parser, but it is still not very efficient. During the initialization process, the IBP parser creates sub-tree index table and key tag index table, so its parsing process actually includes two operations: first, lookup the tables; second, match element in special sub-tree. Using IBP method, index tables are optimized with hash functions, which makes the lookup time almost constant.

5 Handset service: We use Android simulation software to implement handset device web services. We use SOAP protocol as intermediate. kSOAP library [23] is a SOAP web service that can connect with handset devices. **Figure 4.12** is shown that handset device can submit job to resource broker via SOA technology. We also tested on the HTC magic [31] mobile phone. The MedicalGrid web portal showed "testAndroid" was finished. The handset device

screen will Return “true” message to inform user that service (job submit) is OK.

kSOAP is a SOAP web service client library for constrained Java environments such as Applets or J2ME applications.

Please note that SOAP introduces some significant overhead for web services that may be problematic for mobile devices. If you have full control over the client and the server, a REST based architecture may be more adequate.

kSOAP 2 is a complete redesign, taking the lessons learned from kSOAP 1.x into account. Some important changes are:

- Structure cleaned up
- kSOAP2 has improved support for literal encoding
- SOAP Serialization support is now optional and contained in a separate package
- Several separate classes have been integrated into the class SoapSerializationEnvelope, providing SOAP serialization support. SoapSerializationEnvelope extends the base class SoapEnvelope.
- A dotNet flag can be used to switch the SoapSerializationEnvelope from standard behaviour to the namespace handling that seems to be default in .NET

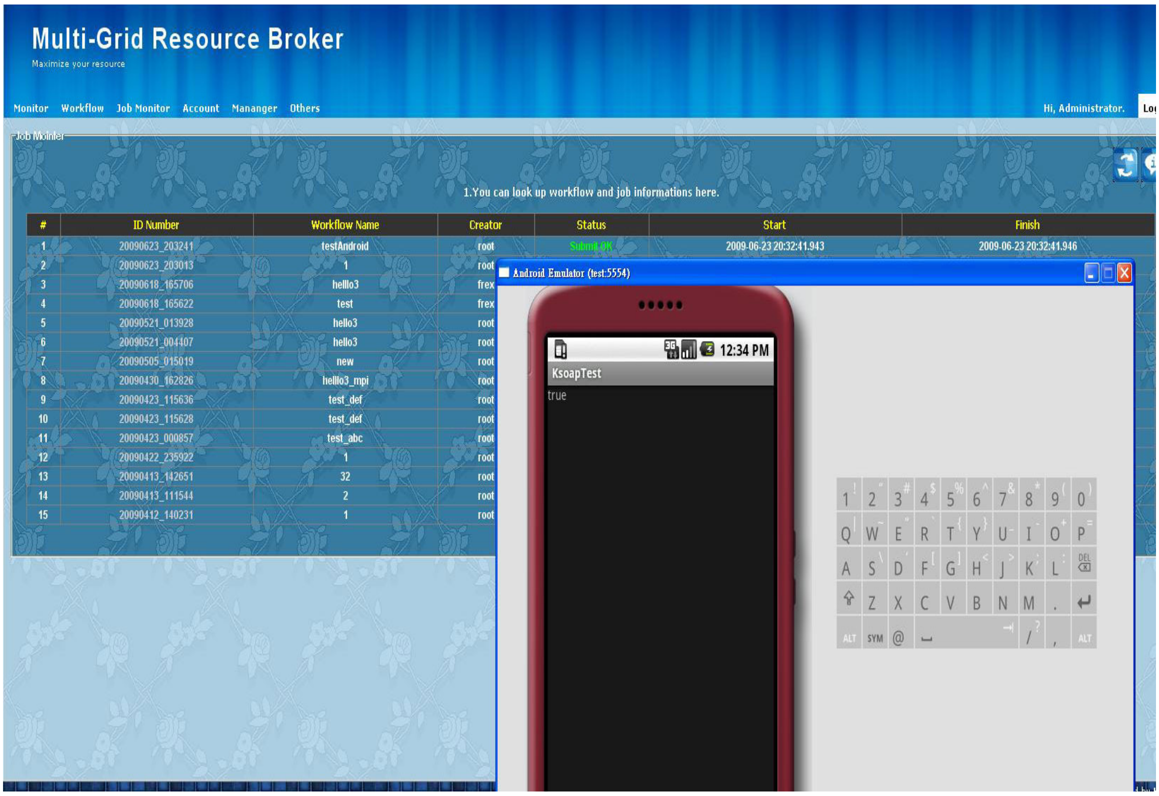


Figure 4.12: Job monitor –handset device submitted job.

4.5 SOA Solution

There are many good services have been written in the Internet. We propose a template to solve the service of source code with different programming languages to write. It can through the SOA technology let heterogeneous systems easy to integrate, and the program also increased the degree of reuse. We do not have to have all the program components or develop the program by ourselves. The developers can get the best combination of services which they need. Boundless to a particular vendor of platform or product features. It can reach real openness. The steps of template describe as the follow:

- We dispatch the Apache Axis [28, 29] server to service providers.

  Servlet container: tomcat 6.0. [30]

- Axis support three methods of web service dispatch and develop:

  1. Dynamic Invocation Interface (DII)

2. Stubs

3. Dynamic Proxy

- Dynamic Proxy:

  If we get a source code: HelloClient

  We can add the source code to AXIS_HOME, rename to HelloClient.jws

  Then we can login to http://localhost:8080/axis/HelloClient.jws?wsdl, the page will show the WSDL which Axis auto-generated.

- First of all, we prepare a proxy interface: HelloClientInterface.java

  The name and parameters definition of the proxy interface and the web service (HelloClient.jws) must be the same. We can use function call to do this.

- Produce and implement client programs. The name and parameters will obtain by the WSDL document.

- Return the result.

Chapter 5

Experiment Results

5.1 System Development Tools

5.1.1 NetBeans

Our development tools using open source: NetBeans IDE 6.5 shown in **Figure 5.1**. NetBeans is set up the open-source software development tools by Sun Microsystems, the framework is an open, extensible development platform, can be used in Java, C language / C++ such as the development itself is a development platform, you can extend the plug-ins to extend functionality.

At NetBeans platform, application software is a series of software modules (modular software components) constructed. These modules are a jar file (Java archive file) that contains a set of categories of Java programs and their implementation in accordance with NetBeans-wide basis for the definition of a public interface as well as a series used to distinguish between different modules described in the definition file (manifest file). Depends on the benefits of modular, with modules to construct an application may need only add a new module can be further expanded. Because modules can be independently developed by the NetBeans platform, so applications developed will be able to use third-party software, very easy and efficient to carry out the expansion.

One of the main features of NetBeans platform is:

1. User interface management
2. User settings management
3. Storage management

4. Window management

5. Wizard framework

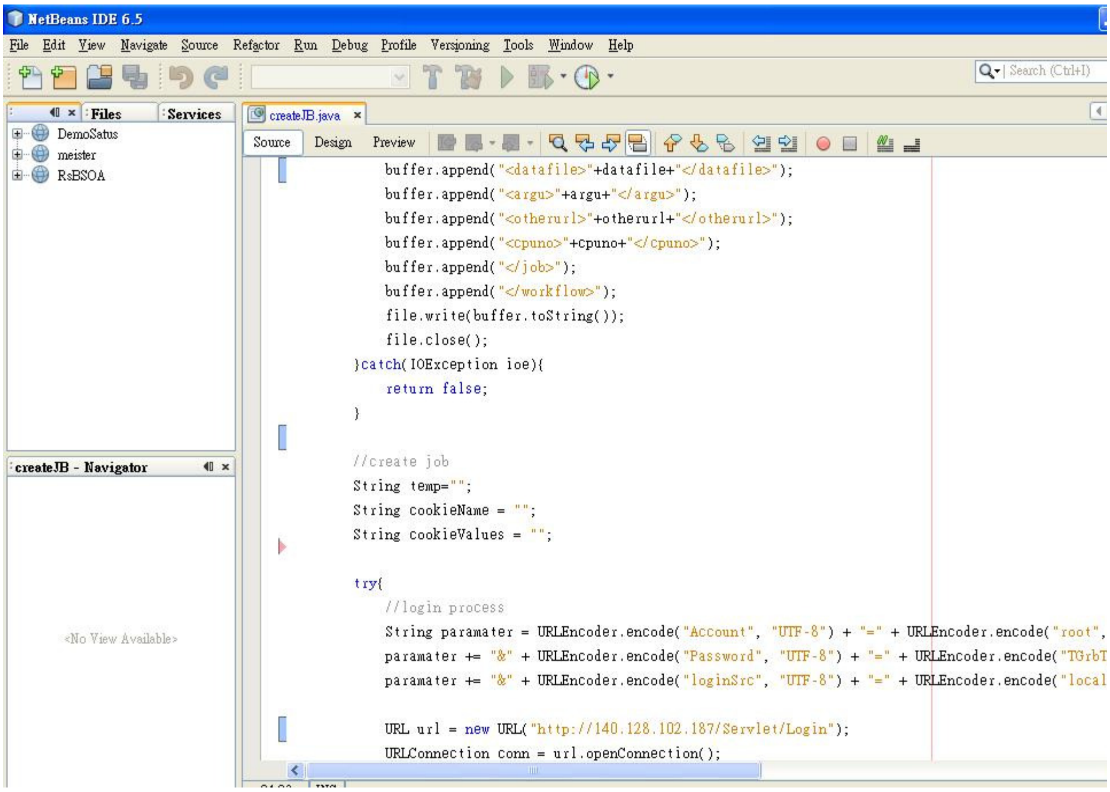


Figure 5.1: Development tool.

5.1.2 Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android platform is the product of the Open Handset Alliance, a group of organizations collaborating to build a better mobile phone. The group, led by Google, includes mobile operators, device handset manufacturers, component manufacturers, software solution and platform providers, and marketing companies. From a software development standpoint, Android sits smack in the middle of the open source world.

The first Android-capable handset on the market was the G1 device manufactured by HTC and provisioned on T-Mobile. The device became available after almost a year of speculation, where the only software development tools available were some incrementally improving SDK releases. As the G1 release date neared, the Android team released SDK V1.0 and applications began surfacing for the new platform.

To spur innovation, Google sponsored two rounds of "Android Developer Challenges," where millions of dollars were given to top contest submissions. A few months after the G1, the Android Market was released, allowing users to browse and download applications directly to their phones. Over about 18 months, a new mobile platform entered the public arena.

With Android's breadth of capabilities, it would be easy to confuse it with a desktop operating system. Android is a layered environment built upon a foundation of the Linux kernel, and it includes rich functions. The UI subsystem includes:

- Windows
- Views
- Widgets for displaying common elements such as edit boxes, lists, and drop-down lists

Android includes an embeddable browser built upon WebKit, the same open source browser engine powering the iPhone's Mobile Safari browser.

Android boasts a healthy array of connectivity options, including WiFi, Bluetooth, and wireless data over a cellular connection (for example, GPRS, EDGE, and 3G). A popular technique in Android applications is to link to Google Maps to display an address directly within an application. Support for location-based services (such as GPS) and accelerometers is also available in the Android software stack, though not all Android devices are equipped with the required hardware. There is also camera support.

5.2 Experimental results and analyses

We calculate the latency time of job submission in personal computer (PC) and handset device. The latency time is denote by a job submit to the resource broker until it return “true” message to user. We can see the result in **Figure 5.2**. The latency time of handset device is higher than PC about 3 times. Because the handset devices use wireless network and android system use SOAP protocol. It spent more time to use XML parser to explain the SOAP message. The SOAP encoding rules include some type of additional information. It will take up some storage space and increase system encoding and decoding of the additional processing time. But the highest latency time of handset device is close to 1.5 seconds. The time can be ignored for user.

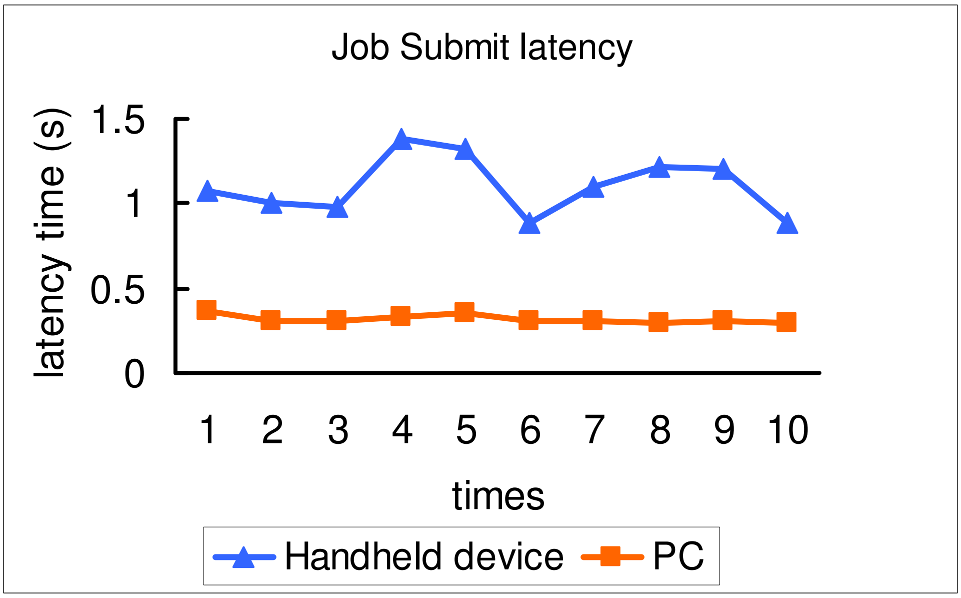


Figure 5.2: Job submit latency comparison.

Chapter 6

Conclusions

6.1 In this thesis, we use SOA technology to develop a healthcare service grid infrastructure. We design and implement a web portal for SOA-based systems and propose the security token to provide more secure. We use SOA and web service technology to integrate with the MedicalGrid. Let MedicalGrid of the computing resources or other services through the standard interface provided the user to use. We also propose a template of SOA solution to integrate heterogeneous services of platform or product feature. The standardized interface of the services that system can easy expansion or deletion. Using SOA architecture is beneficial for our system since it allows easily adding new features without modifying the existing ones. Because these are based on existing services, the code reuse is maximal, and then development and testing time is minimal.

The paper provided an analysis of use of the SOA paradigm in the context of an e-health Web-based system. We design and implementation a SOA web portal. The web portal provides the services that combine resource broker. The resource broker enables users to submit job, monitoring grid nodes and viewing patient records, etc. via a SOA web portal. So, user will be able to take care of their own will be more convenient.

Our feature work will focus on handset device API and the service from hospital. We hope the patient records can follow user all the time. Whenever the user got sick, they can be anywhere for medical treatment.

Bibliography

[1] Xingchen Chu, Andrew Lonie, Peter Harris, S. Randall Thomas and Rajkumar Buyya (2008), "A service-oriented Grid environment for integration of distributed kidney models and resources," *Middleware for Grid Computing: Future Trends,* vol. 20, pp. 1095–1111, 2008.

[2] A. Andrzejak, C. Mastroianni, P. Fragopoulou, D. Kondo, P. Malecot, A. Reinefeld, F.Schintke, T. Schütt, G.-C. Silaghi, L.M. Silva, P. Trunfio, D. Zeinalipour-Yazti, E. Zimeo, "*Grid Architectural Issues: State-of-the-art and Future Trends,*" Project no. FP6-00426, 2008.

[3] Chao-Tung Yang, Tsui-Ting Chen, Keng-Yi Chou, William C. Chu, "Design and Implementation of an Information Service for Cross-Grid Computing Environments," *12th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pp.99-105, 2008.

[4] HL7, "Health Level 7 Version 2.3.1 Final Standard", 1999.

[5] 洪睿璇、黃衍文、張啟明、邱瑞科、彭振興,「標準 HL7/XML 訊息格式轉換系統實作」,第六屆亞太 HL7 健康資訊交換標準研討會。

[6] Tsung-Mei Cheng, "Taiwan's New National Health Insurance Program: Genesis And Experience So Far", *Health Affairs*, no. 3, pp. 61-76, 2003.

[7] Laura E. Grit , "*Broker Architectures for Service-oriented Systems*,", 2005.

[8] Dagmar Krefting, Julian Bart, Kamen Beronov, Olga Dzhimova, Jürgen Falkner, Michael Hartung, Andreas Hoheisel, Tobias A. Knochf, Thomas Lingner, Yassene Mohammed, Kathrin Peter, Erhard Rahm, Ulrich Sax, Dietmar Sommerfeld, Thomas Steinke, Thomas Tolxdorff, Michal Vossberg, Fred Viezens, Anette Weisbecker, "MediGRID: Towards a user friendly secured grid infrastructure," *Future Generation Computer Systems* , vol. 25, pp. 326-336, 2009.

[9] 林蓉懋、吳季勳、甘惠瑩、周怡廷、林苡秀、洪睿璇、黃惠丹、黃衍文，「發展 HL7 訊息編碼與解碼之網路服務自由軟體，協助醫療資訊交換之研究」，國立台北護理學院 資訊管理系。

[10] Bojan Blazona, Miroslav Koncar, "HL7 and DICOM based integration of radiology departments with healthcare enterprise information systems," *International Journal of Medical Informatics*, vol. 76, pp. 425-432, 2007.

[11] G. Aloisio and M. Cafaro, "Web-based access to the Grid using the Grid Resource Broker portal," *Concurrency Computation: Practice and Experience*, vol. 14, pp.1145- 1160, 2002.

[12] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," *Software Practice and Experience*, vol. 32, pp. 135-164, 2002.

[13] C.T. Yang, C.F. Lin, and S.Y. Chen, "A Workflow-based Computational Resource Broker with Information Monitoring in Grids," *in Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, pp. 199-206, 2006.

[14] C.T. Yang, S.Y. Chen, and T.T. Chen, "A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids," *Advances in Grid and Pervasive Computing - Second International Conference, GPC 2007, Lecture Notes in Computer Science*, Editors: Christophe Cerin and Kuan-Ching Li, vol. 4459, pp. 1-12, Springer, Paris, France, May 2-4, 2007. (EI)

[15] Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," I*nternational Journal of Supercomputer Applications*, vol. 11, no. 2, pp. 115-128, 1997.

[16] Health Level Seven, http://www.HL7.org

[17] MultiGrid, http://140.128.102.187/

[18] MedicalGrid, http://eta1.hpc.csie.thu.edu.tw/ganglia/

[19] Wiki, http://en.wikipedia.org/wiki/Service-oriented_architecture.

[20] Microsoft, http://www.microsoft.com.

[21] Haihui Zhang, Xingshe Zhou, Zhiyi Yang, Xiaojun Wu, Zhiwen Yu, "The Research of an XML-based Grid Information Modelling and Retrieval System" *Third International Conference on Semantics*, Knowledge and Grid, vol. 29-31, pp.346-349, 2007.

[22] http://www.microsoft.com/taiwan.

[23] Ksoap, http://ksoap2.sourceforge.net/.

[24] IBM webservice, http://www.ibm.com/developerworks/webservices/.

[25] R. Wolski, N.T. Spring, and J. Hayes, "The network weather service: a distributed resource performance forecasting service for metacomputing", *Future Generation Computer Systems*, vol. 15, pp. 757-768, 1999.

[26] Network Weather Service, http://nws.cs.ucsb.edu/ewiki/.

[27] XML parser, http://www.sosnoski.com/opensrc/xmlbench/results.html.

[28] Apache Axis, http://ws.apache.org/axis/.

[29] Apache Axis, http://www.stylusstudio.com/apache/axis.html#.

[30] Tomcat, http://tomcat.apache.org/.

[31] HTC magic, http://www.htc.com/www/product/magic/overview.html