J Med Syst (2011) 35:391–407
DOI 10.1007/s10916-009-9375-5

ORIGINAL PAPER

A Scalable Healthcare Information System Based on a Service-oriented Architecture

Tzu-Hsiang Yang · Yeali S. Sun · Feipei Lai

Received: 11 July 2009 / Accepted: 30 August 2009 / Published online: 24 September 2009

Abstract Many existing healthcare information systems are composed of a number of heterogeneous systems and face the important issue of system scalability. This paper first describes the comprehensive healthcare information systems used in National Taiwan University Hospital (NTUH) and then presents a service-oriented architecture (SOA)-based healthcare information system (HIS) based on the service standard HL7. The proposed architecture focuses on system scalability, in terms of both hardware and software. Moreover, we describe how scalability is implemented in rightsizing, service groups, databases, and hardware scalability. Although SOA-based systems sometimes display poor performance, through a performance evaluation of our HIS based on SOA, the average response time for outpatient, inpatient, and emergency HL7Central systems are 0.035, 0.04, and 0.036 s, respectively. The outpatient, inpatient, and emergency WebUI average response times are 0.79, 1.25, and 0.82 s. The scalability of the rightsizing project and our evaluation results show that the SOA HIS we propose provides evidence that SOA can provide system scalability and sustainability in a highly demanding healthcare information system.

Keywords Service-oriented architecture (SOA) · HL7 · Healthcare information system (HIS)

Electronic supplementary material The online version of this article (doi:10.1007/s10916-009-9375-5) contains supplementary material, which is available to authorized users.

T.-H. Yang (✉) · Y. S. Sun
Department of Information Management,
National Taiwan University,
Taipei, Taiwan
e-mail: robe@ntu.edu.tw

Y. S. Sun
e-mail: sunny@ntu.edu.tw

T.-H. Yang
Information Systems Office,
National Taiwan University Hospital,
Taipei, Taiwan

F. Lai
Graduate Institute of Biomedical Electronic and Bioinformatics,
National Taiwan University,
Taipei, Taiwan
e-mail: flai@ntu.edu.tw

Introduction

The healthcare information system

Medical information systems were introduced into hospitals three decades ago [1, 2]. Information systems in hospitals help physicians, nurses, and administrative staff in their daily operations. These systems also reduce errors, enhance the quality of patient care, and are patient-centered [3–5] that information is organized according to each patient, making it easier for physicians and nurses to provide appropriate care. Currently, healthcare information systems consist of an outpatient information system, an inpatient information system, an emergency information system, and a number of ancillary systems in the medical center. These systems cooperate with each other in the online processing of computer-based physician order entries (CPOE) [6–8] and treatments to present a single, integrated view of the patient-centered clinical record. The patient clinical record in the health information system represents the integration of multiple ancillary systems, and it must be correct, consistent, and complete to effectively support and improve the quality of physicians' and nurses' clinical decisions.

Healthcare information systems service patients in a variety of complex procedures. For example, the patient registration system handles admission/discharge/transfer (ADT) related processes. The pharmacy system handles pharmacy order entries ordered from outpatient, inpatient or emergency systems by physicians. The billing system must enable manipulation of information on billing accounts, charges, payments, and other related patient billing information. There are also a number of other supporting ancillary systems such as the radiology information system (RIS), the picture archiving and communication system (PACS), the laboratory information system (LIS), and the pathology information system (PIS). The combination of these systems and ancillary systems forms the healthcare information system.

Thus, the healthcare information system is a collection of all information systems in the medical center, with the result that it would be very difficult to use one single software application to support complex and varying patient care and treatment requirements. Medical centers can purchase new applications from software vendors and customize them to meet specific requirements, or they can develop them from scratch. Both approaches are acceptable in the integration of a healthcare information system. Therefore, the important issues and challenges facing the healthcare information system are scalability and interoperability, for both IT professionals and healthcare information employees.

Motivation and objective

The Hospital Information System (HIS) of National Taiwan University Hospital (NTUH), which contains outpatient and inpatient information systems running on an IBM mainframe, cannot fulfill the patient care and requirements of physicians and nurses. To provide higher quality patient care and have better scalability and adaptability in the future, such as the ability to integrate PACS/RIS into the NTUH HIS, it is necessary to rebuild the NTUH HIS. Since the scale of the new NTUH HIS software/hardware system and the performance requirements are unknown, this paper adopts the service oriented architecture concept to build a new NTUH HIS, as this concept allows for scalability, interoperability, reusability and is loosely coupled.

This paper describes the design and implementation of a successful renovation of a service oriented architecture healthcare information system (SOA HIS) for National Taiwan University Hospital (NTUH). The system is based on a service-oriented architecture [9–15] and web services that utilize Health Level Seven (HL7) standards [16] as the level of abstraction [17] of the complex medical environment in NTUH. Moreover, HL7 is used to define a standard interface between our own developing systems based on the SOA HIS database and heterogeneous third-party systems to present a consistent, integrated view of healthcare information to physicians, IT professionals and integrated third parties. Based on the proposed architecture, the entire renovation project started with the new implementation of the outpatient information system and its ancillary systems in July of 2004, and the outpatient system and related ancillary systems have been running since January 2, 2006. The inpatient system and related systems have been running since February, 2007. Finally, the emergency system has been running since February, 2009.

Previous articles [13, 14, 18, 19] presented the system, showing that the open, distributed implementation of this SOA-based HIS successfully supports highly demanding scalability and interoperability, as is the case in our medical center. This paper describes further details of the SOA HIS software, hardware architecture and system scalability and adaptability. This real-life SOA system application experience can be shared with other enterprises that can introduce a SOA to solve their heterogeneous system integration and data consistency problems and to make systems scalable in both their hardware and software aspects.

The rest of the paper is organized as follows. The "Background" section states the problems and issues of the old NTUH HIS system and describes the SOA design concepts. The "SOA HIS design and system architecture" section describes HL7Central implementation in detail. This section also describes the service groups and their sequence diagram, as well as the software architecture and hardware architecture of the SOA HIS. In "The scalability and adaptability of SOA HIS," we describe the scalability of the SOA HIS during the rightsizing project and discuss the scalability of service groups, databases, and hardware systems. The performance of the SOA HIS is presented in the Evaluation section, followed by conclusions.

Background

The problems of the NTUH medical information system before rightsizing

For the past 25 years, the Hospital Information System (HIS) of NTUH was based on the IBM Health Care Support/Patient Care System (PCS) [20]. The IBM PCS, marketed by IBM in the mid-1970s, was initially developed by a group of researchers at Duke University and was named the Duke Hospital Information System (DHIS). At that time, the DHIS provided only patient registration, order entry, pharmacy, accounting and billing services implemented on IBM mainframes such as the IBM 370/390 with terminals. It was widely deployed in the 1980s and remains one of the most widely implemented HISs in the world.

NTUH first introduced outpatient registration services, order entries, and accounting systems in 1983 based on the IBM virtual machine (VM) operating system using PCS. In 1989, NTUH converted all inpatient and outpatient services and ancillary systems to be based on the IBM VM systems. After using IBM Mainframe solutions for about a decade, NTUH built its emergency information system based on the Client/Server paradigm running with PowerBuilder applications and a Sybase database in 2000.

By early 2000, the inpatient, outpatient, and ancillary systems running on the IBM mainframes could no longer sustain the increasing demand for medical services. NTUH had about 2 million outpatients and 700 thousand inpatients in 2005. Worse, there were no mainframe-based solutions available for hospital information systems to cope with scalability and third-party interoperability on such a scale. Since that time, several new IT-driven systems were developed such as LIS, RIS and PACS. These systems must be integrated with the existing HIS systems to provide consistent and coherent services and support for the clinical decisions of physicians. Based on the issues of scalability, interoperability, clinical data consistency, integration and agility, we decided that a new system and software architecture was needed to efficiently manage the increasingly complex medical care environment and healthcare information systems, to be developed in a "rightsizing project" [21]. The mainframe rightsizing project started in December 2003. We decided not to replace the inpatient, outpatient, and emergency information systems at the same time, instead renovating each system one at a time and integrating the new system with PACS/RIS systems.

The biggest challenge in this process was determining how to have maximum system scalability during each step of the rightsizing project and to keep patient data such as patient demography and patient orders consistent between these systems during the rightsizing project. To achieve system hardware and software scalability and keep patient data consistent during the rightsizing steps, we adopted the service oriented architecture approach because of its scalability, reusability, and loose coupling.

The Service-Oriented Architecture (SOA) introduction

A service-oriented architecture is a design style that guides all aspects of creating and using business services throughout their lifecycles. An SOA is also a way to define an IT infrastructure to allow different applications to exchange data and participate in business processes, regardless of the operating systems or programming languages underlying those applications [22]. From the viewpoint of information technology, the service-oriented architecture is essentially a collection of services, where a service is a function that is well defined, self-contained, and does not depend on the context or state of other services.

The implementation of services in an SOA is not limited to a specific technology. Services may be implemented using a wide range of technologies, including SOAP, RPC, DCOM, CORBA, Java's RMI or Web Services [17, 23–26]. The services can be written in C# running on.NET platforms or in Java running Java EE platforms and the applications based on.NET or Java platforms can consume services running on either platform, facilitating reuse. The World Wide Web Consortium (W3C) refers to service-oriented architecture as "a set of components which can be invoked, and whose interface descriptions can be published and discovered" [27]. Also, according to Microsoft, the goal of SOA is to create a worldwide mesh of collaborating services that are published and available for invocation on a service bus [9].

The basic service-oriented architecture is illustrated in Fig. 1 using a service consumer that sends service requests to a service provider. The system should be able to operate without the service consumer knowing how the service provider actually performs its tasks, and without the service provider knowing when the service consumers will require the services. The service should be based on a formal definition such as WSDL, which is independent of the underlying platform and programming language.

The next section describes how to use SOA as a design guideline and then defines the services for the new scalable healthcare information system based on HL7 in the distributed environment.

The SOA HIS Design and System Architecture

The SOA HIS is composed of a number of services. This rightsizing project designed a new database, referred to as the SOA HIS database. All systems, such as outpatient registration, inpatient admission, and third party systems, have to access the SOA HIS database through the HL7Central service. The HL7Central is the web service-based HL7 abstraction layer which is newly introduced in this rightsizing project [28]. To realize the full SOA HIS architecture, we also adopt several service groups which all provide scalable and reliable services in the SOA HIS architecture. This section first describes HL7Central and its implementation, and then describes the service groups introduced in the SOA HIS architecture.

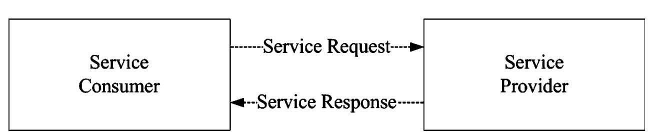

Fig. 1 The basic service-oriented architecture: service request and service response

HL7Central: The web services-based HL7 abstraction layer

As described in the background section, the old LIS system and new third party solutions such as PACS/RIS must be integrated with SOA HIS, so we chose web services implementing the HL7 application protocol as our interface for the SOA HIS database [28].

1) Health Level Seven: Health Level Seven (HL7) is one of the leading standards for exchange of clinical and administrative data among healthcare information systems. HL7 Version 2.5 (HL7 V2.5) specifications, which were approved as an ANSI standard, are a messaging standard that enables different healthcare applications to exchange clinical and administrative data. HL7 can be implemented using web services based on the request and reply service application model. However, the newer HL7 Version 3 based on the object-oriented development methodology and a Reference Information Model (RIM) to create messages [29] was still under development in 2004. Thus, this project used HL7 V2.5 as the messaging protocol in 2004.
2) HL7Central: HL7Central is based on Microsoft Internet Information Services version 6.0, which plays the role of "service provider." HL7Central is the data abstraction layer in the SOA HIS architecture that receives requests, queries SOA HIS databases, and replies to the WebUI using HL7 messages based on XML encoding over SOAP as shown in Fig. 2. The WebUI (service consumer) can only access the business data through the HL7Central (service provider). This process encapsulates the SOA HIS database operations.
3) HL7 messages based on XML encoding over SOAP: the HL7Central server is a web service server based on SOAP. Figure 3a shows HL7 messages based on the XML encoding query string sent using "HTTP POST" by WebUI or third-party applications as an HL7Input string parameter into the HL7Central web service. The HL7Central web service returns the HL7PortResult string parameter back to the WebUI in Fig. 3b.

The service groups introduced in the SOA HIS and their interaction

To achieve scalability for the SOA HIS, we divide servers into several service groups according to their functionalities. There are different communication mechanisms between service groups, and each service group has a primary service. This session describes the functionality of each service group and the communication protocols between service groups. For the high availability requirements of the healthcare environment, each service group has at least 2 real servers to provide service through a layer 4 switch, and each has its own suitable layer 4 service operation modes such as "load balance mode" or "active–standby mode". Figure 4 and Table 1 describe the 9 service groups in the system architecture. We will explain their configuration and describe the protocols for their communication with other service groups in the following sections.

Portal service group

The portal service that provides the Web-based SOA HIS login page has two real servers running in load balance mode using a persistent hash load balance matrix to provide consistent service. When a user logs in to the SOA HIS system, the real portal server issues the web services requests to the virtual IP address of the AA web service group. When the authentication processes finish, the user will be redirected into the portal function list page that shows the inpatient system, outpatient system, billing system, pharmacy system, and related ancillary systems. Authenticated users can choose their system from the menu and the portal server will do a "URL Redirect" to the user's browser, which will be redirected to the virtual IP address of the chosen WebUI URL, such as "http://oWebUI/WebApplication/Clinics/Default.aspx". The "oWebUI" hostname in the URL, which provides outpatient WebUI services running on several real servers, is the layer 4 virtual IP address of the outpatient WebUI service. Tables 3 and 4 show a summary of these portal service group characteristics.

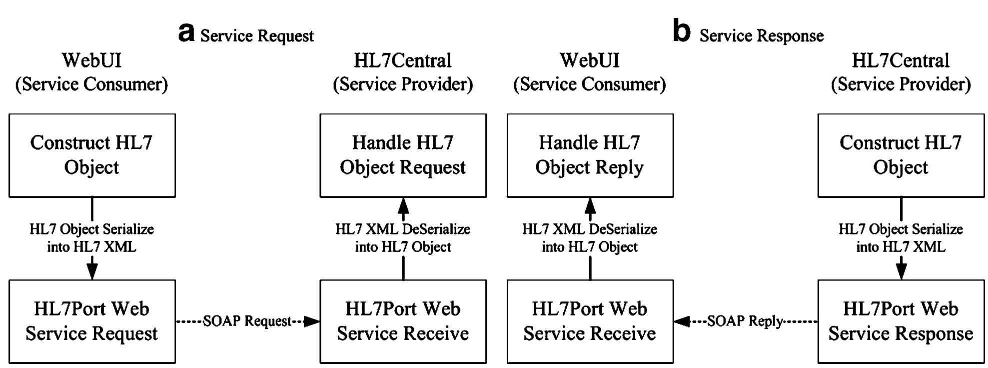

Fig. 2 HL7Central and WebUI are based on the service-oriented architecture

Fig. 3 **a** The SOAP request of HL7Central HL7Port web service. **b** The SOAP response of HL7Central HL7Port web service

a

```
POST /HL7Central_IIS/Central.asmx HTTP/1.1
Host: hiswsvc
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/HL7Central/HL7Port"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HL7Port xmlns="http://localhost/HL7Central/">
      <HL7Input>string</HL7Input>
    </HL7Port>
  </soap:Body>
</soap:Envelope>
```

b

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HL7PortResponse xmlns="http://localhost/HL7Central/">
      <HL7PortResult>string</HL7PortResult>
    </HL7PortResponse>
  </soap:Body>
</soap:Envelope>
```

Authentication and Authorization (AA) web service group

The authentication and authorization (AA) web service group provides authentication and authorization to the portal service group, inpatient WebUI, and outpatient WebUI service groups in the SOA HIS. The authentication and authorization services are based on the web service and running in layer 4 load balance mode using

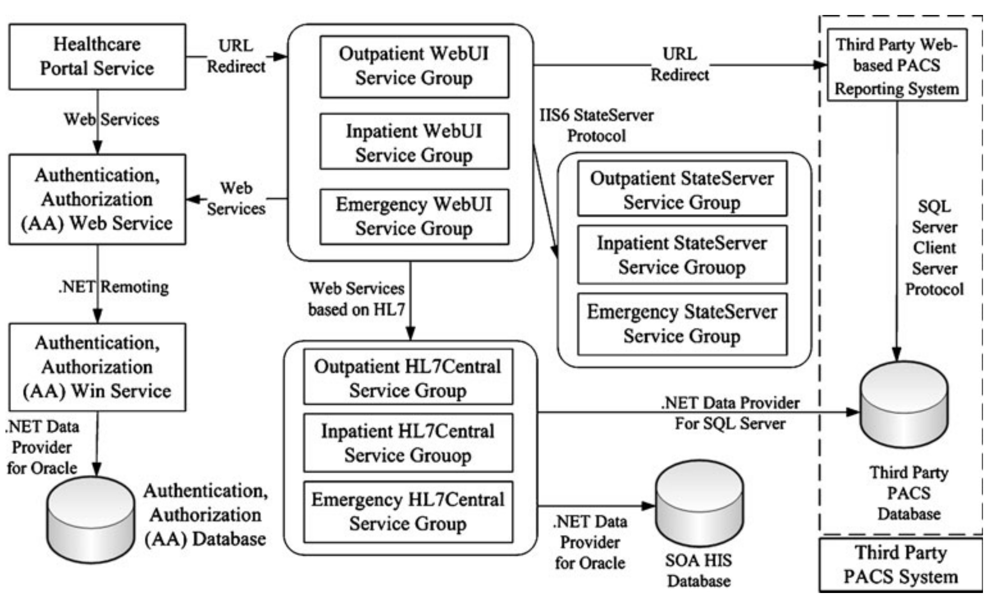

Fig. 4 SOA HIS service groups

Table 1 Service group name and its virtual server and real server settings

Service group name	FQDN of VIP	Virtual IP address	Layer 4 mode	IP of real servers
Portal Services	Portal	192.168.1.2	Load Balance/	192.168.2.1
			Persistent Hash	192.168.2.2
AA Web Service	WebAA	192.168.1.3	Load Balance/	192.168.3.1
			Least Connections	192.168.3.2
AA Win Service	WinAA	192.168.1.4	Active–standby/	192.168.4.1
			Persistent Hash	192.168.4.2
Outpatient WebUI	oWebUI	192.168.1.5	Load Balance/	192.168.5.1, 192.168.5.2
			Persistent Hash	192.168.5.3, 192.168.5.3
Outpatient HL7Central	oHL7Centra	192.168.1.6	Load Balance/	192.168.6.1,192.168.6.2
			Least Connections	192.168.6.3, 192.168.6.4
Outpatient StateServer	oStateServer	192.168.1.7	Active–standby/	192.168.7.1
			Persistent Hash	192.168.7.2
Inpatient WebUI	iWebUI	192.168.1.8	Load Balance/	192.168.8.1, 192.168.8.2
			Persistent Hash	192.168.8.3, 192.168.8.4
Inpatient HL7Central	iHL7Central	192.168.1.9	Load Balance/	192.168.9.1, 192.168.9.2
			Least Connections	192.168.9.3, 192.168.9.4
Inpatient StateServer	iStateServer	192.168.1.10	Active–standby/	192.168.10.1
			Persistent Hash	192.168.10.2

least connections. The AA web service group also provides Enterprise authentication to third-party applications, such as PACS authentication, which provides a single-sign-on mechanism in this SOA HIS. The AA web service group issues the.NET Remoting requests to the AA Win Service group, and the Win Service group does the real authentication and authorization validations. AA web service group only receives requests and redirects the web service based call into.NET Remoting calls to the AA Win Service group. Tables 3 and 4 show the AA web service group characteristics.

Authentication and Authorization (AA) win service group

The Authentication and Authorization (AA) Win services group receives the AA web service group real server requests and processes the authentication and authorization requests. The AA Win Service group has 2 real servers; each real server runs a windows service program, “SessionServices”. It handles the authentication and authorization functions, and keeps about 19 thousand authentication records and about 1 million authorization records. The authentication and authorization records are kept in memory to provide fast and real time authentication and authorization responses. The reason for the AA Win-Service group running in active–standby mode is that the active SessionServices program running on the real server keeps the database connection persistent, holds related tables, and periodically writes the authenticated users' information to the database tables. This is so that the standby SessionServices program running on the other real server cannot access related tables at the same time. Currently, we only can manually change 2 real servers into active or stand-by-mode.

Outpatient/Inpatient WebUI service group

The WebUI service group provides main CPOE interfaces to the physicians, nurses, and administrative users. In our SOA HIS, we have defined outpatient and inpatient WebUI service groups. Each WebUI service group also has its own layer 4 virtual IP address and fully qualified domain name (FQDN). We put the outpatient related systems in the outpatient WebUI service group real servers, and inpatient related systems are put in the inpatient WebUI service group. The outpatient WebUI service group is running in load balance mode with a persistent hash matrix. It has 10 real servers to service user requests. The WebUI service group real servers issue requests to the AA web service group and HL7Central service group. Each WebUI page has to check user authentication and authorization for the current page. This makes the real server issue web service requests to the AA web service group to check if the user has permission to access the page. If the user has permission to access the page, then the WebUI page will continue the process, otherwise the user will be redirected to the no-permission page. After the user session has obtained permission, the Outpatient WebUI real server has to fetch data from the HL7Central service group. It issues web service requests to the HL7Central service group based on HL7 XML message encoding, as described in the previous section. Tables 3 and 4 show the WebUI service group characteristics for the outpatient, Inpatient, and emergency systems. The WebUI service group also performs ASP.NET State Server requests to the ASP.NET State Server which provides the HTTP session state services between Web server and user browsers.

The Inpatient WebUI service group has the same environment as the Outpatient WebUI service group which provides outpatient and related services, and the same environment as the emergency WebUI service group. The WebUI service group and corresponding HL7Central service group provide good system scalability in this architecture. We will describe the scalability of the SOA HIS architecture in the section "The scalability and adaptability of SOA HIS".

Outpatient, inpatient, and emergency HL7Central service groups

The HL7Central service group receives requests from WebUI real servers and relays the queries to the SOA HIS database according to the requests. The protocol between WebUI and HL7Central is based on the web service which is encoded by an HL7 message in the XML format as described in the previous section. In our architecture, we have designed our system such that the outpatient HL7Central service group receives requests from outpatient WebUI real servers. The separation of outpatient and inpatient design is good for system maintenance and trouble shooting, such as system upgrades, fixing programming bugs, or system hardware maintenance. If the outpatient system has hardware or application problems, such as a long SQL query or programming bugs, we can deal with the outpatient WebUI or outpatient HL7Central real servers and the inpatient system can perform services without system downtime. For this reason, we have decided to separate the outpatient HL7Central service group and inpatient HL7Central service group. Tables 3 and 4 show a summary of the outpatient and inpatient HL7Central service groups' characteristics.

Outpatient, inpatient, and emergency IIS state server service group

In our architecture, we use ASP.NET running on the IIS 6.0 as our Web server. The ASP.NET uses a session state [30] to manage its server-side state. The ASP.NET session state has 5 modes described in Table 2: (1) In-Process (InProc) (2) State Server (3) SQLServer (4) Custom (5) Off.

The session state is used to store an HTTP session between the user's browser and Web servers on the Web server side. If there is only one Web server, all session state of user browsers are stored at the same process in the Web server. But we use a distributed server farm architecture to provide the outpatient WebUI service group with 10 real servers, so that there is not only one Web server to serve user browser requests. In this server farm environment, the InProc mode cannot service correctly if there is a real server crash. For this reason, we use State Server mode to store HTTP session state. The State Server service group also has its own Layer 4 virtual IP address and related Layer 4 service configuration. Tables 3 and 4 summarize the outpatient, inpatient, and emergency WebUI and HL7Central service group characteristics.

Data exchange central

In the next step of the rightsizing project, we introduced an HIS Data Exchange Central server into the new SOA HIS architecture. The Data Exchange Central handles all of the data exchange and data synchronization requirements from the SOA HIS system to the other systems. The Data Exchange Central server plays an important role in realizing the SOA HIS during each step of the rightsizing, as this server means that the outpatient information system, related ancillary systems and third party systems do not need to know about the details of data integrity and synchronization

Table 2 ASP.NET session state modes and descriptions

ASP.NET session state mode	Description
InProc	The HTTP session is stored in the Web server process, such as w3wp.exe for IIS version 6. This is the In-process mode session state.
ASP.NET State Service	The HTTP session is running in the out-of-process mode. The session state is stored on the ASP.NET state service which is a Windows service provide by.NET.
SQLServer	The HTTP session is running in the out-of-process mode, but the session state is stored on the SQLServer.
Custom	This mode stores session state data using a custom session state store provider, and system architecture and programmers have to implement the session state store provider.
Off	This mode disables session state.

Table 3 Server components and layer 4 configurations

SOA HIS server roles		Layer 4 service: Enabled	Mode	L.B matrix	Real servers: Prod. Env.	Test Env.	Dev. Env.
Portal		V	L.B	P-Hash	2	1	1
AA Web Service		V	L.B	L-Conn.	2	1	1
AA Win Service		V	A.S	P-Hash	2	1	0
Outpatient	Web UI	V	L.B	P-Hash	12	1	1[b]
	HL7Central	V	L.B	L-Conn.	12	1	0[b]
	StateServer	V	A.S	P-Hash	2	1[a]	0[b]
Inpatient	Web UI	V	L.B	P-Hash	6	1	0[b]
	HL7Central	V	L.B	L-Conn.	6	1	0[b]
	StateServer	V	A.S	P-Hash	2	0[a]	0[b]
Emergency	Web UI	V	L.B	P-Hash	3	1	0[b]
	HL7Central	V	L.B	L-Conn.	3	1	0[b]
	StateServer	V	A.S	P-Hash	2	0[a]	0[b]
Data Exchange Central		V	A.S	P-Hash	2	1	1
Oracle Database		X	Oracle L. B	Oracle RAC	Oracle RAC Database Solaris 9	Oracle RAC Database Solaris 9	Oracle Database Windows

L.B load balance, *A.S* active–standby, *P-Hash* persistent hash, *L-Conn.* least connections, *0[b]* use 1[b] as the same server, *0[a]* use 1[a] as the same server, *Prod. Env.* production environment, *Test Env.* test environment, *Dev. Env.* development environment

behind the HL7Central. Instead, they only have to consider how to send and receive HL7 messages over SOAP.

As illustrated in Fig. 5, all individual healthcare systems have four components: WebUI, HL7Central, Data Exchange Central, and the databases. The solid lines from ancillary or third party systems to the right hand database elements indicate that when an add patient (ADT_A05 with event A28) HL7 message is sent from the WebUI, the HL7Central web service receives the request and processes it with the ADT_A05 event A28 module. The ADT_A05 with event A28 is a module that requires data exchange, so the HL7Central will send the request to the Data Exchange Central. Finally, the Data Exchange Central adds the patient to the SOA HIS database system, laboratory information system (LIS), emergency information system, PACS/RIS systems, and mainframe inpatient system through the client server database communication protocol or mainframe protocol.

In another scenario depicted in Fig. 5, the physicians choose laboratory test orders from the outpatient COPE system, and the WebUI sends the OML_O21 with event O21 to the HL7Central. After the HL7Central processes the

Table 4 Communication mechanisms between service groups

	Portal	AA Web Service	AA Win Services	Web UI	HL7Central	StateServer	Data-Ex. Central	SOA HIS Database
Portal								
AA Web Service	Web Service							
AA Win Services	X	.NET Remoting						
Web UI	URL Redirect	Web Service	X					
HL7Central	X	Web Service	X	Web Service (HL7MSG)				
StateServer	X	X	X	ASP.NET Session-State	X			
Data-Ex. Central	X	X	X	X	TCP	X		
SOA HIS Database	X	X	Oracle Client	X	Oracle Client	X	Oracle Client	

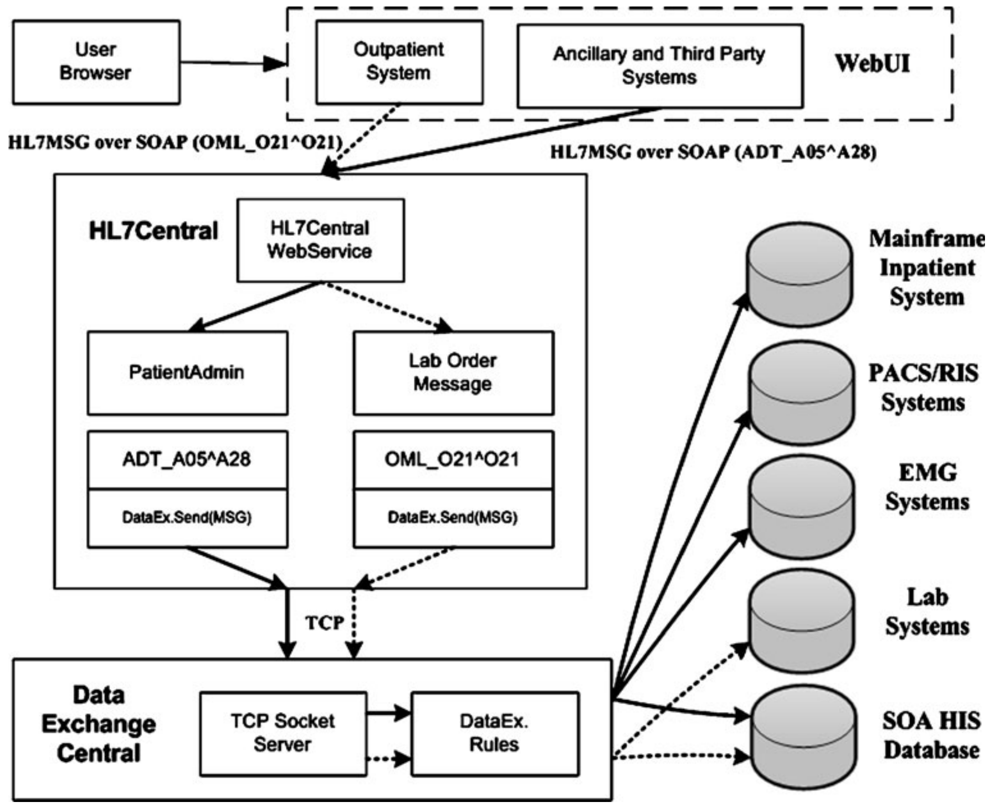

Fig. 5 Data exchange central and its procedure within the SOA HIS

order, the HL7Central sends the order to the Data Exchange Central, and the Data Exchange Central adds the order to the SOA HIS database and the laboratory information system database for future processing.

Figure 6 depicts the sequence diagram of the SOA HIS. The WebUI and HL7Central entries in this chart combine the outpatient, inpatient and emergency categories. Also, the AA Service entry in Fig. 6 contains both AA web services and AA Win Services.

Software architecture of SOA HIS

All individual healthcare information systems and ancillary systems use HL7Central to achieve data consistency and heterogeneous system integration. As a result of using SOA, the Web-based user interface in Fig. 7 becomes a 4-tier solution. Figure 7 also shows the integrated SOA HIS user interface running on the Microsoft Internet Explorer browser. The physicians can use integrated WebUI to enter orders, check laboratory reports, and view PACS images in an integrated environment.

To enable users' browsers to do local printing and reading/writing of the Taiwanese National Health Insurance (NHI) IC Card [31], user components such as ActiveX [32] or Smart Client [33] in Microsoft.NET must be embedded in the browser. The few embedded user components become a 3-tier solution, which we refer to as a "4-tier mixed with 3-tier web-based solution" as shown in Fig. 8.

Hardware architecture of SOA HIS

To provide a stable HIS environment, this SOA HIS is based on a distributed environment that uses firewalls and layer 4 switches to provide a server farm services network. The server farm has 2 firewalls and 2 layer 4 switches running in the active–standby mode. The firewall limits the user browser's access to servers in the server farm, and the layer 4 switch plays a role in service load balance and provides high availability. For the most important services, we configure virtual service port numbers and virtual IP addresses in the layer 4 switches. Usually, one virtual service must have at least 2 real servers providing the service in the server farm, and the virtual service can be configured in load balance mode or active–standby mode.

Figure 9 shows the user network and server farm architecture in the SOA HIS, and Table 3 shows the detailed layer 4 configurations of each server component. All of the network devices, networking cables, servers, and power supplies are configured in a redundant mode in case of a single point failure, and this paper does not discuss complex hardware configuration.

The scalability and adaptability of SOA HIS

The many heterogeneous systems in the NTUH healthcare environment presented a significant challenge during the

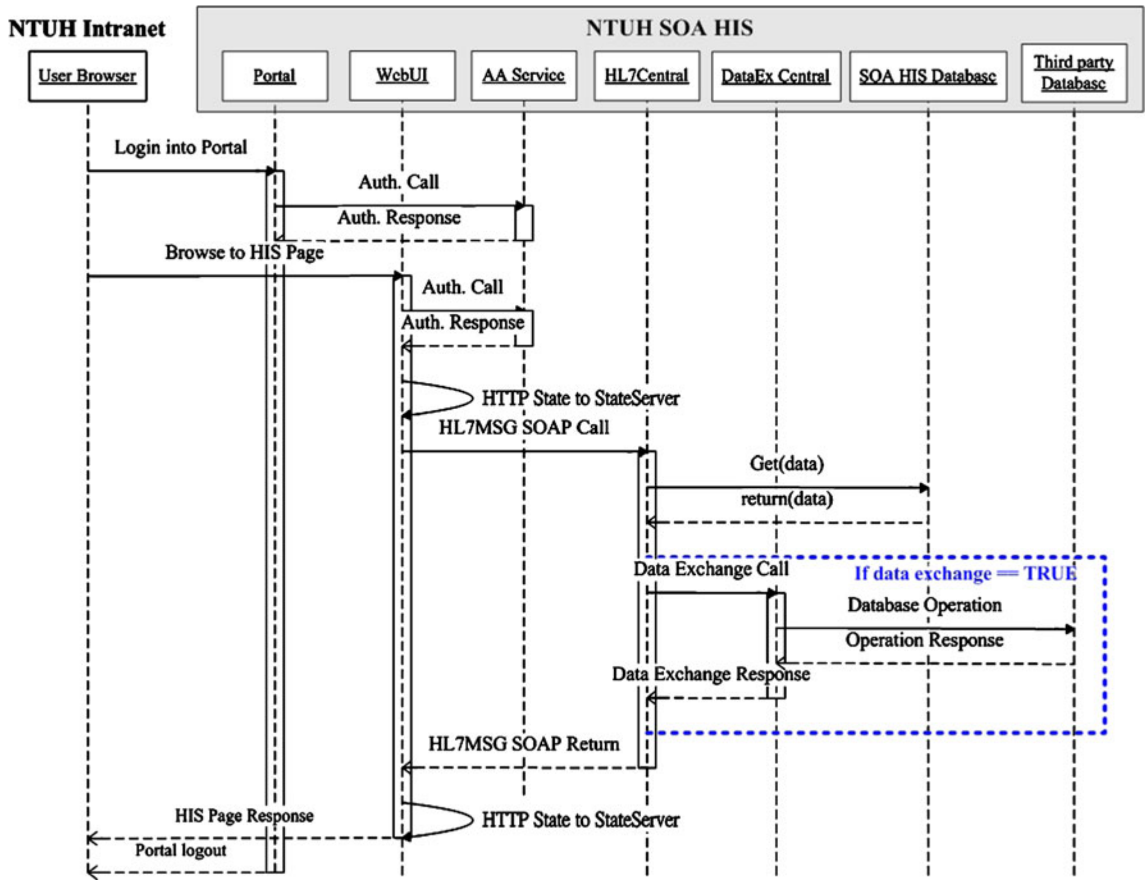

Fig. 6 The sequence diagram of the SOA HIS

rightsizing project. This section describes how this SOA HIS architecture is scalable both during and after the rightsizing project, and it also describes the scalability of the SOA HIS service groups, databases and hardware.

The scalability and adaptability of the SOA HIS during the rightsizing project

This rightsizing project consists of three main steps, outpatient, inpatient, and emergency system rightsizing. The outpatient and inpatient systems were previously running on the mainframe system with PCS and IMSDB, and they did not have the scalability and hardware performance to meet the new requirements. The emergency system was previously running on the HPUX with a Sybase database and a user interface written by PowerBuilder. The database hardware of the emergency system and its programming technology are out-of-date and cannot be easily integrated with RIS/PACS and other Web-based reporting systems.

In the first stage of outpatient rightsizing, the outpatient system runs on the new SOA HIS system, the inpatient system runs on the IBM mainframe PCS and IMSDB, and the emergency system runs on the HPUX with a Sybase database. We encountered complex data exchange environment and data integrity issues, as illustrated in the previous paragraph. In this stage, new patient registrations coming from the outpatient system will also be added to the inpatient and emergency systems through the HL7Central and the Data Exchange Central processing.

In the second stage of inpatient rightsizing, the inpatient information system is integrated with the outpatient information system using HL7Central to access the SOA HIS database and third party databases. In the second stage, there are no requirements imposed on the data exchange with the IBM mainframe system, and data exchange processing can be easily disabled in Data Exchange Central. This procedure has no impact on the WebUI and HL7Central developers, which do not need to recompile programs and deploy their systems.

In the final stage of emergency system rightsizing, the emergency information system was integrated into the SOA HIS database with the inpatient and outpatient systems. At this time, the code for data exchange was disabled in the ADT_A05 event A28 module in HL7Central. Data Exchange Central no longer receives

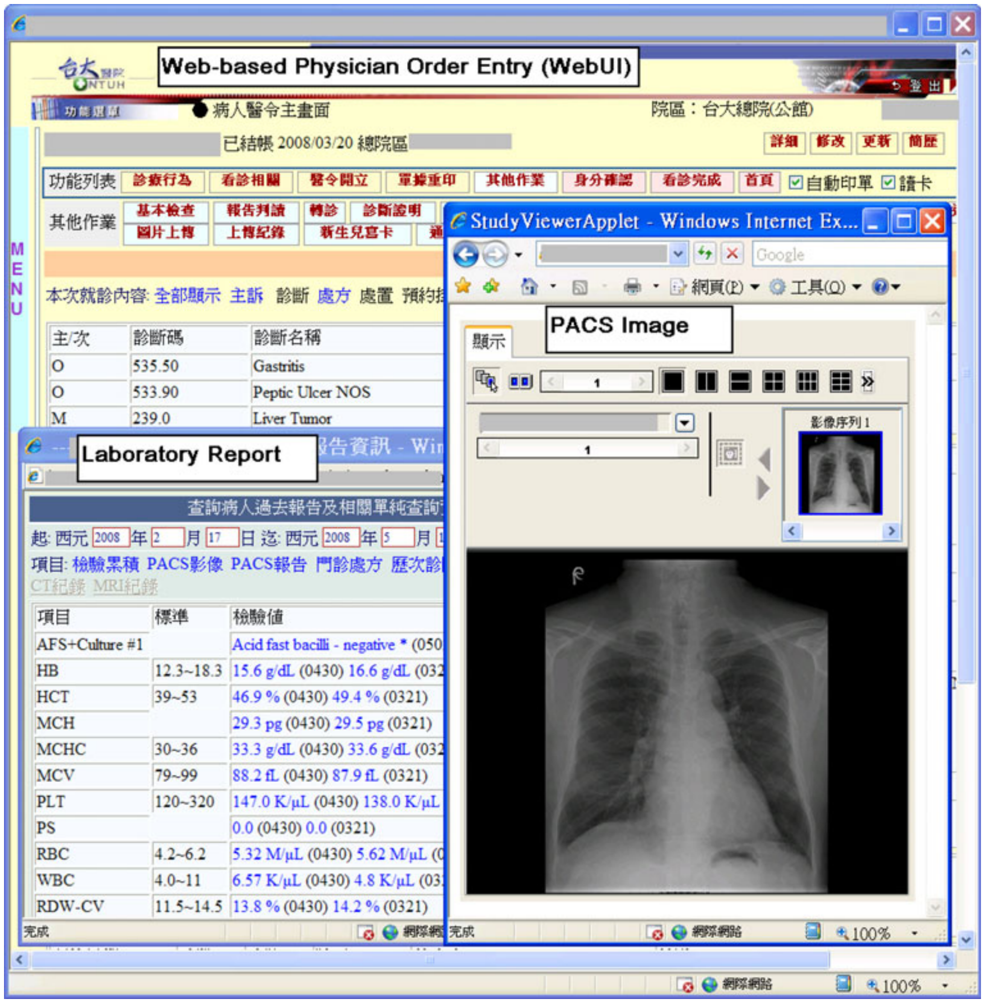

Fig. 7 The Web-based WebUI (CPOE) of SOA HIS

patient registration requests from HL7Central. The WebUI developers do not need to know about the complex data synchronization work behind the backend. This kind of service-oriented architecture reduces development time and gives a method of assigning work to programmers.

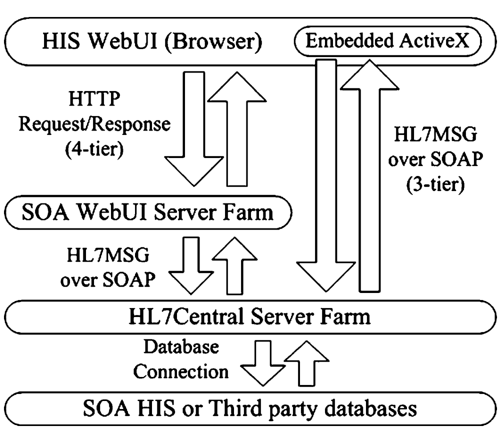

Fig. 8 4-tier mixed with 3-tier Web-based SOA HIS

The scalability and adaptability of the service groups in the SOA HIS

The outpatient project system began development in July 2004, and was online on January 2, 2006. At that time, we had 6 service groups in our SOA HIS: (1) portal service, (2) AA web service, (2) AA Win Service, (3) outpatient WebUI, (4) outpatient HL7Central, (5) outpatient State Server, and (6) Data Exchange Central. By February 24, 2007, the inpatient system was online and an inpatient WebUI service group with 6 servers had been added, as well as an inpatient HL7Central service group with 6 real servers.

On 18 February 2009, we brought the emergency system online. We followed the inpatient online procedure to easily construct emergency system related service groups in the SOA HIS, as described in the following steps:

(1) Prepare 8 servers and install server hardware, Windows 2003, Microsoft.NET runtime library, and Internet Information Services 6.0 (IIS version 6.0).

(2) Assign virtual IP and fully qualified domain name (FQDN) for each of the emergency WebUI, HL7Cen-

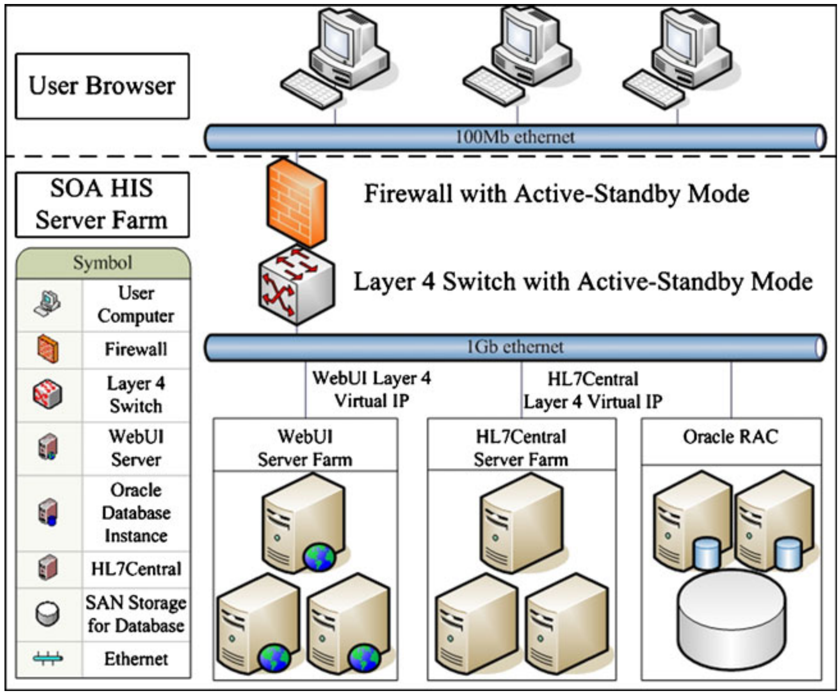

Fig. 9 SOA HIS hardware architecture

tral, State Server service groups and configure them in the Layer 4 switch and domain name servers.

(3) Assign 3 real servers to the emergency WebUI service group, 3 real servers to the emergency HL7Central service group, and 2 real servers to the emergency State Server service group.

(4) Configure Layer 4 switch settings such as service mode and a load balance matrix for each service group, as shown in Table 3.

(5) Deploy corresponding applications to each service group and set up the related IIS application nodes and settings.

The database scalability of the SOA HIS

According to the HL7Central service group as the middle layer of the SOA HIS, the SOA HIS has the ability to extend the database behind the HL7Central service group middle layer. For patient privacy, an HIS usage access log database was constructed to log "which employers access which patients at what time" in the SOA HIS after the second stage of inpatient system rightsizing. This was done using the MySQL database version 5.1.29 built on FreeBSD 7.1 running on the PC server, which has a dual Intel Xeon® 1.6 GHz Quad-Core with 4 G memory. Currently, there are about 60 thousand records in the outpatient, inpatient and emergency system access logs.

After the third stage of emergency system rightsizing, we started a pilot run to support the electronic medical record (EMR) project of the Department of Health (DOH) in Taiwan. This EMR project focuses on Public Key Infrastructure (PKI) based on a medical electronic certification mechanism to ensure the security of medical information. The DOH established the Health Certification Authority (HCA) and issued an HCA IC Card to physicians and nurses in Taiwan. This EMR project uses a new database schema that stores electronic medical records signed by nurses. Figure 10 shows the database scalability of the SOA HIS based on the HL7Central service group.

The hardware scalability and adaptability of the SOA HIS

This rightsizing project uses a design that was developed step by step through the cycle of development, testing, and production. As shown in Table 3, 54 servers serve the whole production environment, 11 servers serve the testing environment, and only 4 servers serve the developing environment. The developing environment database runs on

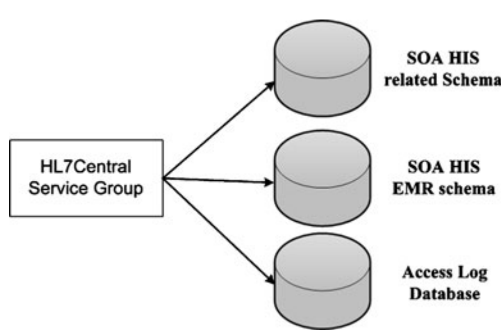

Fig. 10 SOA HIS database scalability based on the HL7Central service group

Windows 2003 with Oracle 9i. The testing environment and production environment databases run on Solaris 9i with Oracle 9i RAC. To ensure software quality, the program uses the same version source code and database schema through the developing, testing, and production processes.

Evaluation

NTUH has to serve about 9,000 outpatients and 300 emergency patients in a busy day. Also, there are about 2200 inpatient beds in the NTUH. About 4300 workstations have accessed the SOA HIS from 2009/05/25 to 2009/05/31.

We chose WebUI request processing and HL7Central request processing times as benchmarks to evaluate the performance of SOA HIS. To analyze the average performance, we aggregated all of the WebUI and HL7Central server logs and used the server processing times for each Web page and web service for our performance evaluation. The server processing time for each page in the IIS log file is the "time-taken" field. The time-taken field in the IIS log file means the length of time that page took to be processed in milliseconds. The "cs-uri-stem" field is the target of the WebUI page being accessed, for example, "/WebApplication/Clinics/OpenClinics.aspx." The URI prefix, "/WebApplication/", is our standard Web application, "/Clinics/" is the system name, and "OpenClinics.aspx" is the page being executed.

(1) The performance evaluation of the outpatient WebUI and HL7Central service groups

Figure 11 shows the outpatient system request counts and average response times during the week from 2009/05/25 to 2009/05/31. The average response time over the week for the outpatient WebUI system was about 0.79 s in Fig. 11. NTUH has no outpatient clinics Sunday and Saturday afternoons, so there were few request counts on these days. Also, 2009/05/28 was the Dragon Boat Festival, when NTUH did not offer outpatient clinic services. From Monday to Friday except for Thursday, outpatient clinics were open in both the mornings and afternoons. The outpatient SOA HIS system received a large number of requests as shown in Fig. 11 from 09:00 to 11:00 in the morning and from 14:00 to 16:00 in the afternoon. The request count peaked on Monday from 10:00 to 11:00, and the outpatient WebUI service group average response times are below 1.2 s from 08:00 am to 17:00 during this week. This shows that the outpatient WebUI service group of the SOA HIS is fast and robust enough to survive in a highly demanding healthcare environment. To clearly show the whole range of request counts, from a low below one thousand and a peak above 60 thousand, the Y axis is graphed on a log scale.

As shown in Fig. 12, the HL7Central servers receive more requests than the WebUI servers. The HL7Central servers still have a reasonable response time around 0.035 s for the heavy load requests. Similar to WebUI weekly statistics, high request counts were recorded during outpatient clinics service time. The HL7Central processing time contains the SOA HIS databases and few third-party database operations. This means that the SOA HIS runs

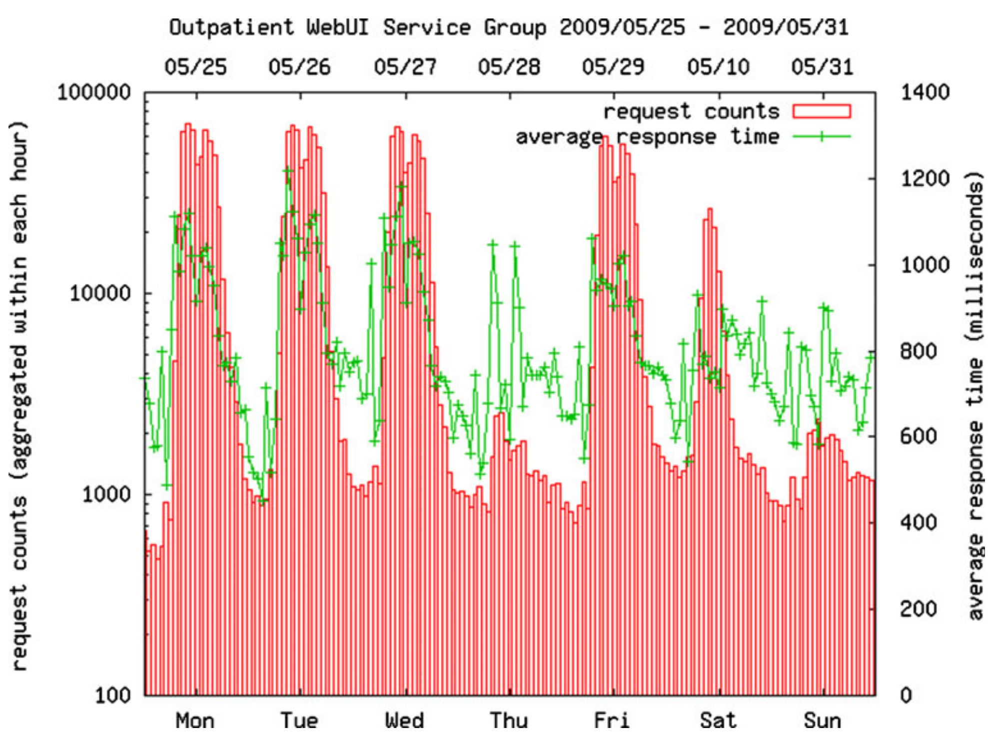

Fig. 11 Outpatient WebUI service group statistics

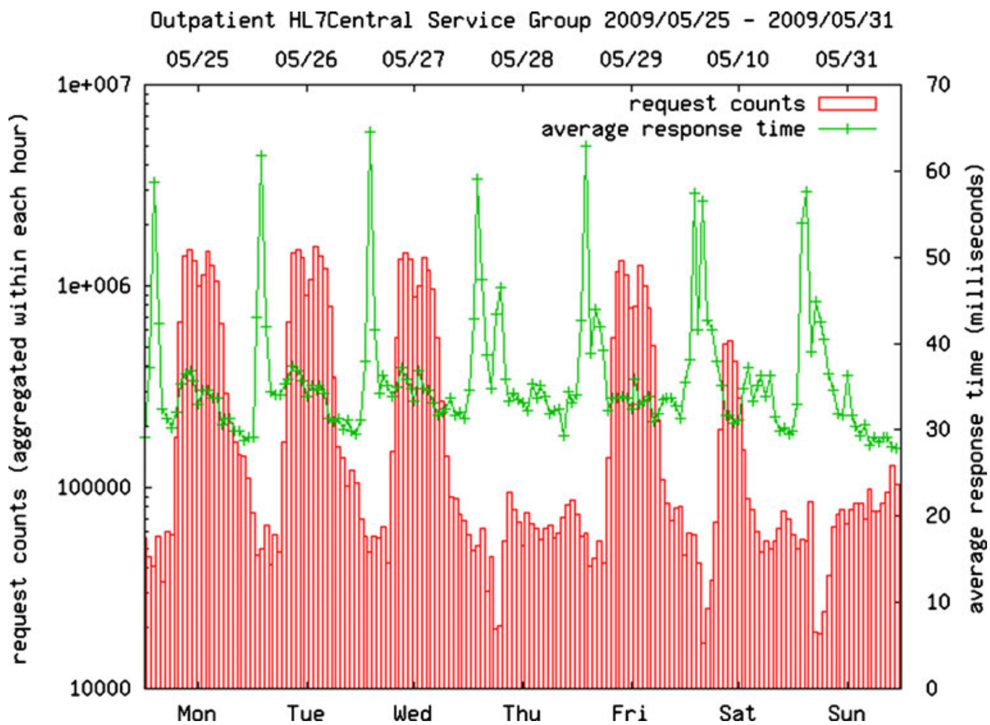

Fig. 12 Outpatient HL7Central service group statistics

with an excellent performance. The performance evaluation of the database is beyond the scope of this paper.

(2) The inpatient WebUI and HL7Central service groups performance evaluation

The same period from 2009/05/25 to 2009/05/31 was used to analyze the performance of the inpatient WebUI and HL7Central systems. The phenomenon of a heavy request load experienced by the inpatient WebUI and HL7Central systems from 08:00 to 18:00 every day is different from the outpatient clinic system, but the average response time aggregates within each hour are still within the acceptable range of 1 s.

The average response time over the week for the inpatient WebUI system is about 1.25 s in Fig. 13, and the peak average response time was 2.1 s from 09:00 to

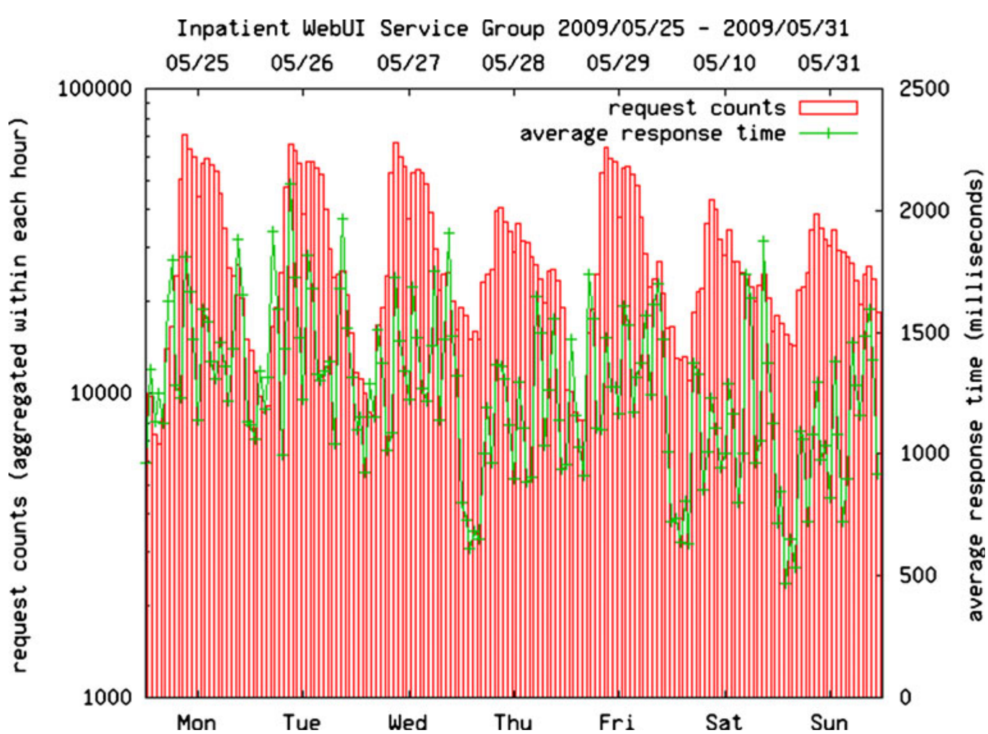

Fig. 13 Inpatient WebUI service group statistics

10:00 on Tuesday. The average response time over the week for the inpatient HL7Central service group is about 0.04 s, as shown in Fig. 14, and the peak average response time was 0.068 s from 00:00 to 01:00 on Monday. Also, the peak number of request counts for the week was about 70 thousand, from 09:00 to 10:00 on Monday.

As shown in Figs. 13 and 14, the inpatient system is used 24 h a day, and it is utilized differently than the outpatient system. This inpatient utilization is notable in several ways, including the fact that nurses and physicians check patient orders at night. This paper will not discuss the reasons for the differing patterns of utilization between the inpatient and outpatient systems in detail, but we do show that the response times of WebUI and HL7Central service groups are acceptable for the different systems in the same SOA HIS architecture.

(3) Performance of the emergency WebUI and HL7Central service groups

The same period from 2009/05/25 to 2009/05/31 was used to analyze the performance of the emergency WebUI and HL7Central systems. Emergency WebUI and HL7Central, which had a heavy request load from 07:00 to 23:59 on Tuesday and from 00:00 to 03:00 on Wednesday, is different from the outpatient and inpatient systems, but the average response time for each hour is still within the acceptable range of 1 s. Moreover, the request counts and the average response times of the emergency WebUI and HL7Central service groups are not affected by the Dragon Boat Festival.

The average response time over the week for the emergency WebUI service group is about 0.82 s, as shown in Fig. 15, and the peak average response time is 2.2 s from 10:00 to 11:00 on Friday. Also, the peak number of WebUI request counts during this week is about 3,400, from 15:00 to 16:00 on Tuesday. The peak average response time of 2.2 s could result from some program or batch jobs, but the log would need to be examined to find out which application caused the long operation time, and this will not be discussed further in this paper.

The average response time over a week for the emergency HL7Central service group is about 0.036 s, as shown in Fig. 16, and the peak average response time is 0.054 s, from 05:00 to 06:00 on Saturday. Also, the peak number of HL7Central request counts during this week is about 52 thousand, from 15:00 to 16:00 on Sunday.

According to the performance results of the outpatient, inpatient, and emergency HL7Central and WebUI service group logs, a healthcare information system based on the service-oriented architecture can provide a reasonable response time for healthcare information requirements.

As space is limited in this paper, we only present the statistics of the outpatient, inpatient, and emergency systems from 2009/05/25 to 2009/05/31. We do not discuss each individual ancillary system, such as the outpatient registration system, the billing system, the outpatient

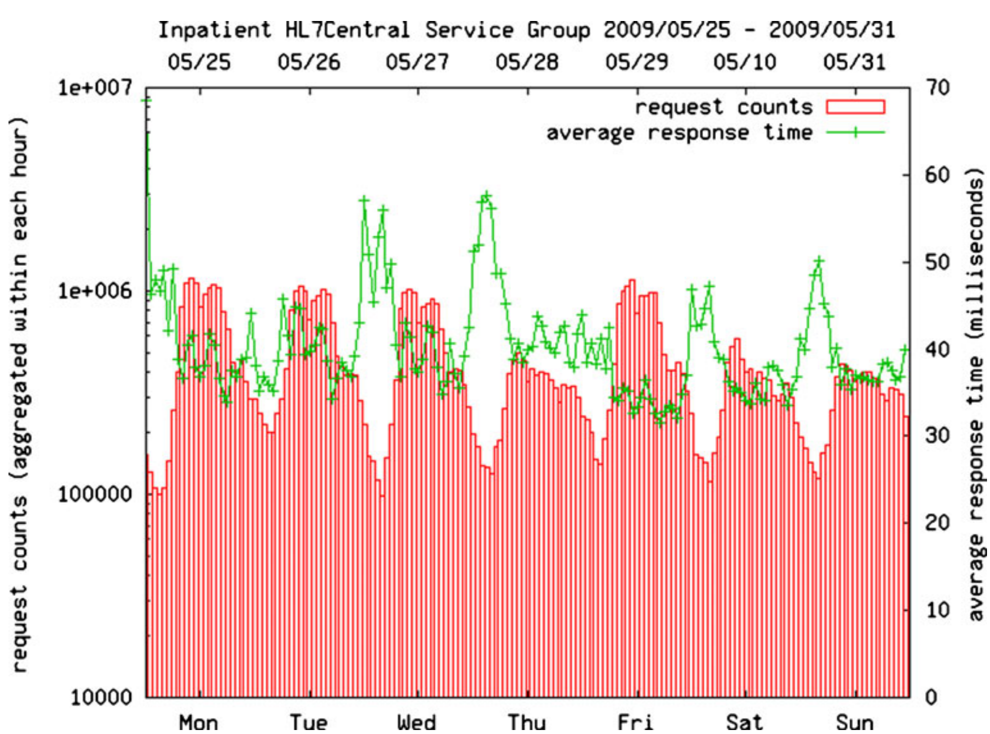

Fig. 14 Inpatient HL7Central service group statistics

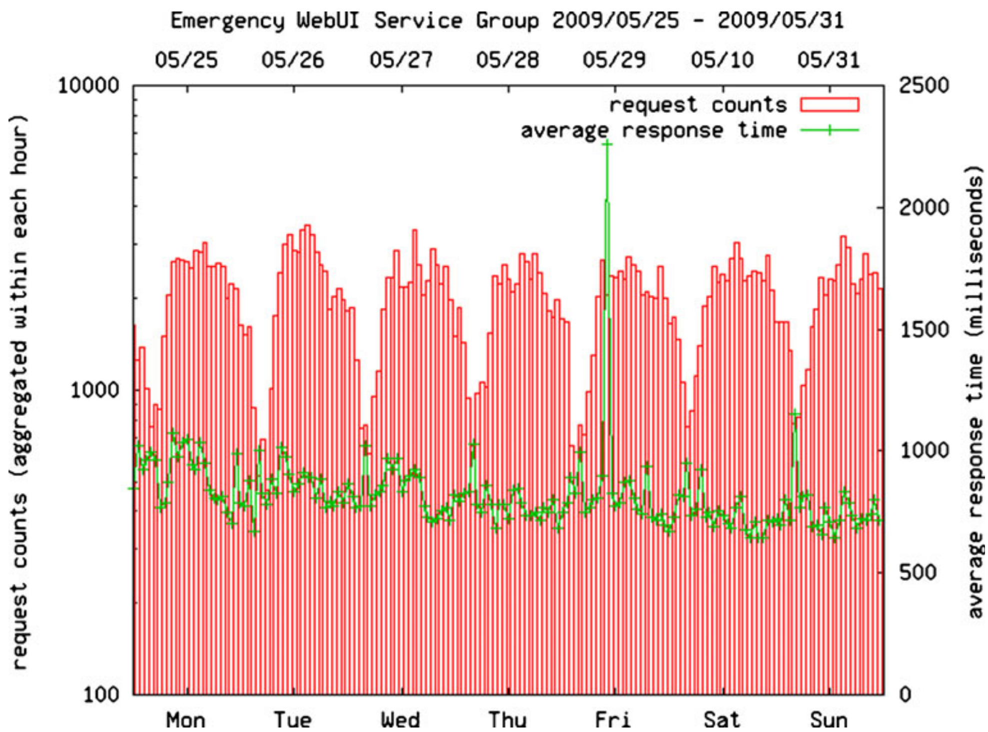

Fig. 15 Emergency WebUI service group statistics

physician order entry system, and the medical administration system. Figures 12 and 15 present the response time measurement during the week; the pikes of the average response times are due to batch jobs or program bugs; these special cases are not discussed in this paper.

Conclusion

This paper describes a successful rightsizing project from an IBM mainframe to a multi-tier architecture for the new healthcare information system based on SOA implemented

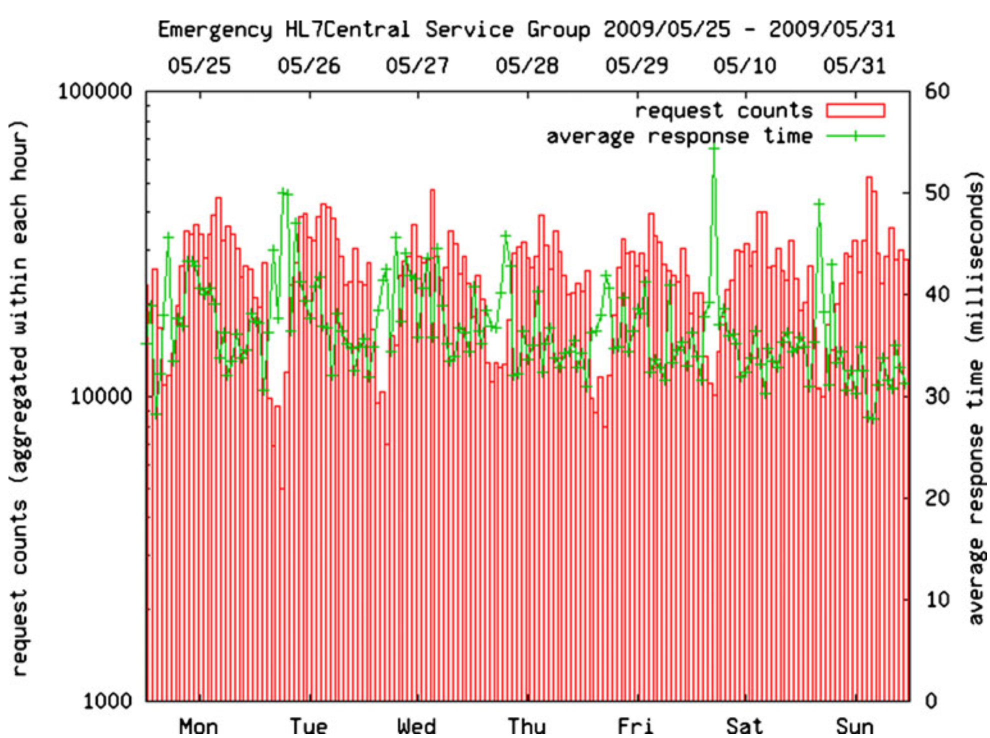

Fig. 16 Emergency HL7Central service group statistics

in National Taiwan University Hospital. The NTUH SOA HIS used HL7Central, based on the HL7 standard, and web service to implement the service-oriented architecture, which contains several service groups and provides a scalable system architecture. The NTUH SOA HIS demonstrated scalability during the rightsizing project, along with service groups, databases, and hardware. Moreover, the NTUH SOA HIS brings our mainframe HIS from a terminal solution to a state-of-the-art, web-based, multi-tier SOA solution. The performance evaluations reported here demonstrate that the SOA HIS solution also delivers reasonable response time under the different outpatient, inpatient, and emergency system usage conditions. This successful integrated healthcare information system provides evidence that an open and distributed environment based on SOA is sustainable in a highly demanding healthcare environment.

References

1. Tolchin, S. G., Stewart, R. L., Kahn, S. A., et al., A prototype generalized network technology for hospitals. *J. Med. Syst.* 6:359–375, 1982.
2. Watson, R., and Jenkin, M., A large-scale professionally oriented medical information system—Five years later. *J. Med. Syst.* 1:3–21, 1977.
3. Kohn, L., Corrigan, J., and Donaldson, M., *To err is human: building a safer health system*. *National Academy Press*, Washington, DC, USA, 2000.
4. Haux, R., Winter, A., and Brigl, B., *Strategic information management in hospitals: an introduction to hospital information systems*. Springer, New York, NY, USA, 2004.
5. Haux, R., Health information systems—past, present, future. *Int. J. Med. Inform.* 75:268–281, 2006.
6. Sittig, D. F., and Stead, W. W., Computer-based physician order entry: the state of the art. *J. Am. Med. Inform. Assoc.* 1:108–123, 1994.
7. Koppel, R., Metlay, J. P., Cohen, A., et al., Role of computerized physician order entry systems in facilitating medication errors. *JAMA*. 293:1197–1203, 2005.
8. Berger, R. G., and Kichak, J. P., Computerized physician order entry: Helpful or harmful? *J. Am. Med. Inform. Assoc.* 11:100–103, 2004.
9. Service Oriented Architecture. Available from: http://msdn2.microsoft.com/zh-tw/architecture/aa948857.aspx. 2008.
10. Michael, J. C., SOA what? *In Computer*. 41:92–94, 2008.
11. Nadkarni, P. M., and Miller, R. A., Service-oriented architecture in medical software: promises and perils. *J. Am. Med. Inform. Assoc.* 14:244–6, 2007.
12. Katehakis, D. G., Sfakianakis, S. G., Kavlentakis, G., et al., Delivering a lifelong integrated electronic health record based on a service oriented architecture. *Inf. Technol. Biomed., IEEE Trans.* 11:639–650, 2007.
13. Hsieh, S. H., Hsieh, S. L., Weng, Y. C., et al., Middleware based Inpatient Healthcare Information System. *In Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, pp. 1230–1234, 2007.
14. Yang, T. H., Cheng, P. H., Yang, C. H., et al., A Scalable Multi-tier Architecture for the National Taiwan University Hospital Information System based on HL7 Standard. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*: IEEE Computer Society, 2006.
15. A New Blueprint for I.T. Available from: http://www.cio.com/article/10125/_A_New_Blueprint_for_I.T. 2005.
16. HL7 Messaging Standard Version 2.5. ANSI/HL7 V2.5-2003, 2003.
17. Steve, J., Toward an acceptable definition of service. *IEEE Softw.* 22:87–93, 2005.
18. Hsieh, S.-H., Hsieh, S.-L., Chien, Y.-H., et al., Newborn screening healthcare information system based on service-oriented architecture. *J. Med. Syst.* 2009.
19. Hsieh, S.-H., Hou, I. C., Cheng, P.-H., et al., Design and implementation of web-based mobile electronic medication administration record. *J. Med. Syst.* 2009.
20. Mishelevich, D. J., and Slyke, D. V., The IBM patient care system: an overview of its software. *SIGMIS Database*. 11:64–75, 1980.
21. Abul-Hamayel, M. A., and El-Halmoushi, N. S., Moving mainframe VM users to a distributed UNIX system (KFUPM VM rightsizing experience). *Int. J. Netw. Manag.* 7:339–350, 1997.
22. Newcomer, E., and Lomow, G., Service-oriented architecture. *In Understanding SOA with Web Services*. Addison Wesley, pp. 10–13, 2005.
23. What Is Service-Oriented Architecture. Available from: http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html. 2003.
24. Migrating to a service-oriented architecture, Part 1. Available from: http://www.ibm.com/developerworks/library/ws-migratesoa/. 2003.
25. Service-oriented architecture. Available from: http://en.wikipedia.org/wiki/Service-oriented_architecture. 2007.
26. Service Oriented Architecture—SOA. Available from: http://www-306.ibm.com/software/solutions/soa/. 2007.
27. Web Services Glossary. Available from: http://www.w3.org/TR/ws-gloss/.
28. Ko, L.-F., Lin, J. C., Yang, T. H., et al., HL7 Middleware Framework for Healthcare Information System. *In 8th International Conference on e-Health Networking, Application & Services (HEALTHCOM 2006)* New Delhi, India, 2006.
29. HL7 Standards. Available from: http://www.hl7.org/Library/standards.cfm. 2009.
30. Session-State Modes. Available from: http://msdn.microsoft.com/en-us/library/ms178586.aspx. 2008.
31. National Health Insurance (NHI) IC Card (traditional Chinese). Available from: http://www.nhi.gov.tw/IC_Card/news/news.asp?menu=9&menu_id=175. 2009.
32. ActiveX. Available from: http://en.wikipedia.org/wiki/ActiveX. 2009.
33. Smart Client Architecture and Design Guide. Available from: http://msdn.microsoft.com/en-us/library/ms998506.aspx. 2009.