

HL7 Middleware Framework for Healthcare Information System

Li-Fan Ko, Jen-Chiun Lin, Chi-Huang Chen, Jie-Sheng Chang, Faipai Lai
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

Tzu-Hsiang Yang, Po-Hsun Cheng, Chia-Chang Wen, Jun-Lian Chen
Information Systems Office
National Taiwan University Hospital
Taipei, Taiwan

Kai-Ping Hsu
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

Siao-Lin Hsieh
Computer Centre
National Chiao Tung University
Hsinchu, Taiwan

Abstract—In this paper, a new middleware framework developed for the Health Information System (HIS) rightsizing project of National Taiwan University Hospital (NTUH) is proposed. The framework is basically a Service-Oriented Architecture (SOA). Challenges as formatting complex medical information as well as integrating heterogeneous systems in our hospital are addressed by introducing HL7 and Web services standard into our framework. Finally, the performance of a operating HIS based on our framework is analyzed and presented to evaluate the efficiency of our design.

Keywords—HL7, Healthcare information system, web services, Service-oriented Architecture

I. INTRODUCTION

National Taiwan University Hospital (NTUH) owns about 2,000 beds for inpatients and serves approximately 8,000 outpatients a day. The hospital uses multiple information systems to manage patients' demographic data, medical records, equipments, and facilities, as well as pharmaceutical events occurred daily within the hospital, for examples: outpatient registrations, emergency treatments, laboratory results, etc. In addition, the systems keep tracking routines, activities regarding auditing, billing and accounting. These systems together form the NTUH healthcare information system (HIS).

Since the early 1980s, NTUH outpatient and inpatient information systems have been implemented based on the IBM PCS environment [8]. Other systems, such as the laboratory information system (LIS) or the emergency information system were based on heterogeneous systems such as Sybase, Oracle, OS/390, HP-UX and etc. As time and technologies advancing, the digitized, multi-media medical data as well as electronic administrative office environment have been dramatically demanding in the recent years. While introducing new systems and the cutting edge technologies into the legacy NTUH healthcare systems, it becomes increasingly difficult. Therefore, since December, 2003, NTUH has begun to rebuild

and restructure the systems in order to provide efficient and integrated services for not only electronic, educational environment establishment but patients, staff and faculty members' usages as well [5].

A modern middleware framework is designed to make our newly developed HIS available, scalable, robust, of high performance. In order to integrate different platforms and databases, a multi-tiered Service-Oriented Architecture (SOA) is chosen while HL7 standard is wildly adapted to cover all data exchange [9, 10]. Hence, the transition from the legacy system into the new one can be applied smoothly. Furthermore, the proposed framework treats all services as independent modules that can be hot-swapped without interrupting the whole system. This enables our new systems to be deployed and configured in the highest degree of flexibility. In the following part of this paper, we will describe the requirement, design, deployment and performance of our middleware framework.

II. SYSTEM REQUIREMENTS

The most important task of developing a modern HIS is to define and/or adapts architecture for exchanging various medical information and data across the system. In our new HIS architecture design, several requirements have to be met while choosing and defining the protocols to be used in our system.

First, our new system is designed to be operated by all outpatient service staff including clinical doctors, nurses, pharmacists, administrative personnel and etc. Hence, dissimilar and deviated user interface would be needed. Moreover, information from different medical staffs could be quite varied. The software architecture must be able to cover and handle all information.

Second, in NTUH, there are many existing medical information systems co-existed with the legacy HIS [7]. These systems are not going to be phased out in a short term. So that,

the new HIS must be able to communicate with these old systems, for example, the LIS. Furthermore, there are also some systems under developing while constructing the new HIS like medical PACS (Picture Archiving and Communication System). The ability of exchanging data between our new HIS and other new systems is also a must.

And last, although, we started developing our out-patient service system, the ultimate goal of the entire project is to design a new HIS that covers all healthcare services in NTUH including out-patient, in-patient and emergence services. Thus, the skeleton of our new HIS has to be able easily extended or re-used for further development.

Health Level Seven (HL7) standard [1] is chosen to carry all information between Web UI servers and data exchange servers. HL7 is an ANSI standard widely used and adapted in many health institutions and medical instruments. Originally, HL7 was supposed to act as the seventh layer in Open System Interconnection (OSI) model. It means that HL7 was designed to be directly transmitted over transporting layer protocol, for example: TCP/IP.

Transmitting HL7 messages over TCP has its benefits, such as simple and easy to implement for small applications. But, for a large system like our HIS, using direct TCP connection means that we have to establish dedicated connection channels across the hospital network system. By doing this, we would have to open extra TCP ports for handling multiple and concurrent connections manually. These may not only increase the complexity of our system design and the risk of being attacked but also lose the robustness and flexibility of the HIS system.

To address these concerns, we decide not to transmit HL7 message directly over TCP socket. Instead, SOAP (Simple Object Access Protocol)/WSDL (Web Service Description Language) Web services is introduced into our system. SOAP/WSDL Web services, or in general, Web services, is a new generation mechanism enabling interoperation between software systems.

By combining multiple W3C XML (eXtensible Markup Language) standards in different parts of system communication, Web service allows message or data to be exchanged across network and program process. The most advantage of Web services is that there is almost no restriction of programming language for a system using Web services. Since Web services can be discovered, described and communicated by UDDI (Universal Description Discovery and Integration), WSDL and SOAP respectively [3, 4], any platform which can generate these XMLs may have the ability to consume Web services. The other benefit of using Web services is that it consumes and provides exchange data by SOAP or HTTP GET/POST mechanism. Because SOAP itself is also a kind of content carried by HTTP, Web services can exchange data by normal HTTP channel. This means that by using Web services, data exchange can be achieved without additional network configurations. Because of these, Web services technology is adapted in many major middleware solution architectures such as IBM WebSphere or Microsoft .NET Framework. Therefore, developing a Web services-

based system could be facilitated by well matured developing tools.

With these advantages, Web service is chosen in our HIS as the key role of middleware. We decide not to transmit HL7 directly over TCP, instead, we exchange HL7 message through Web services. The solution architecture lets our system be benefited by the comprehensive and flexible definition of HL7 but leaves the sophisticated connection problems to be handled by Web services.

III. SYSTEM DESIGNS

To realize our HL7/Web service architecture, we divide our middleware into three components.

A. HL7Message Modules

In our design, HL7 messages are utilized to format all information exchanged across systems. Since complex business logic process may be applied onto HL7 messages while inputting, presenting and data exchanging, an HL7 software library is implemented by the NTUH. We decided to create a DOM-like (Document Object Model) HL7 library because HL7 message is well structured. A DOM-like implementation could make it easier for developer to access elements in each message. First of all, a C#-based class library is generated by parsing the XML schema of HL7 version 2.5. Every element in HL7 definition is mapped as an object including fields, segments, and messages. Then, these objects are instantiated at service run-time.

When an HL7 message is needed, instance of the message is created and manipulated by program logic. Later, the message could be written out as an XML document [2] and prepared to be carried by SOAP. On the other hand, when an HL7 message is received from SOAP content, an instance of HL7 object could also be created by reading back the XML document via the same library, therefore, both sender and receiver could process the message in a symmetric manner.

B. HL7Central WebService

As we mentioned before, Web service is adapted in our system as middleware layer. In our design, we called this Web service layer: HL7Central. As it is named, the HL7Central, which accepts HL7 message as the only parameter of Web service call, is the single entry point of any HL7 message in our system. HL7Central service accepts Web service requests from different user-interface modules or external systems. In another word, this HL7Central is designed to act as the front-end of data exchange tier in our n-tier system architecture. Hence, any HIS application which needs to exchange information must form its data in HL7/SOAP and send its request to HL7Central. When an HL7 message is passed to HL7Central, the service process will perform the following process steps:

1. First, it roughly parses the message to get name and event type in HL7 definition.
2. According to the message name and event types, HL7Central dynamically loads and invokes correspondent subroutine to handle the HL7 message data exchange.

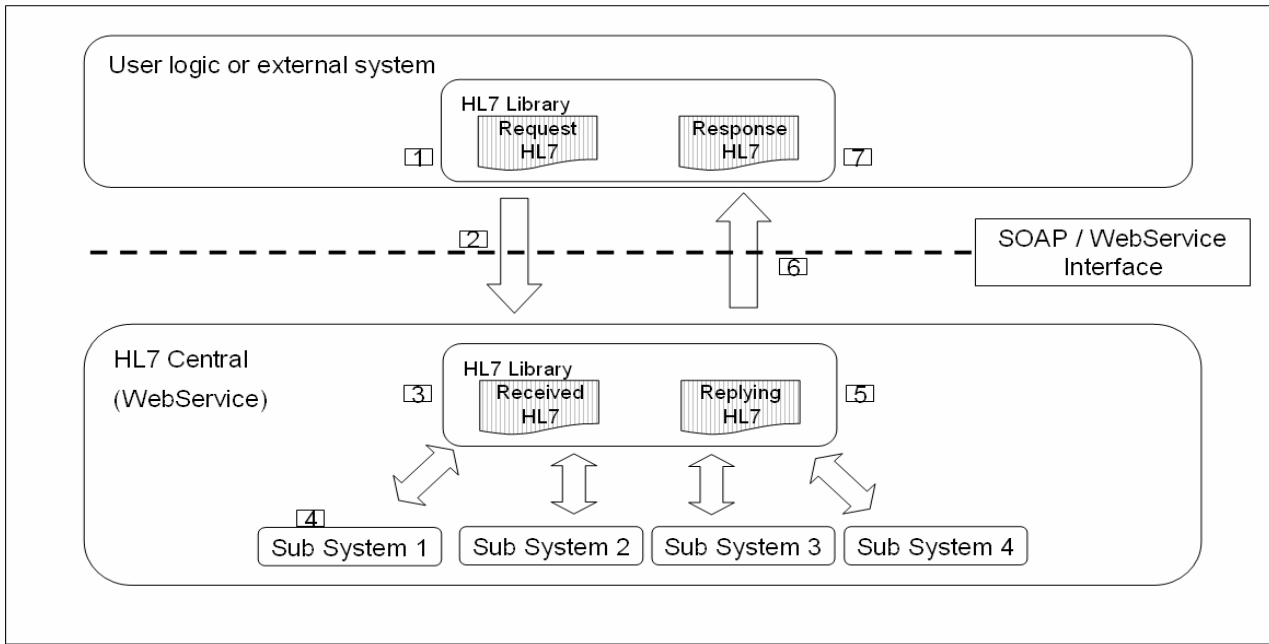


Figure 1. Relation of NTUH HL7 Framework Components. Note that numbers in rectangle represent steps of routine HL7 exchange process

3. Finally, HL7Central collects responding message for sub-routine and returns it to the service caller.

C. HL7 Sub-System Handlers

In last section, we have mentioned that HL7Central will dynamically invoke certain subroutine for handling HL7 message requests. These subroutines are arranged as several subsystems in our systems. Subsystems are divided by their functionality such as document management, finance, observation reporting, order entry, patient administration, resource scheduling and etc. Database connections are handled at these subsystems. Usually, when an HL7 message arrives at subsystem, it directly connects to our HIS database, performs data transaction, and mapping the result as response HL7 message. But in some cases, data exchange between our new HIS and other legacy databases may be required. For these cases, data synchronization between HIS databases and legacy by external data exchange server may be invoked by subsystem. However, the final result will still be processed by subsystems and returns as HL7 messages. Since all database connection is initiated from here, subsystems can be taken as the back-end of data exchange tier.

These subsystems together form the basic services provided by our new HIS. To give an idea of what services can be covered by our new system, a part of our subsystem modules and its functionalities are listed in Table 1.

In summary, these three components form the solution of our framework as Fig 1. . The HL7 library is utilized in both client and Web services sides while multiple subsystems are leveraged by HL7Central. A typical HL7 message exchange process can be briefly described as steps illustrated in Fig. 1: (1) A HL7 message is formed, manipulated by HL7 library; (2) the request HL7 is written out as XML and sent by SOAP to

HL7 Web service; (3) HL7Central reconstructs received message then dispatches it to dedicated HL7 message handler subsystems; (4) The corresponding subsystem performs necessary processes for incoming HL7 message; (5) The result of HL7 message exchange are encapsulated in replying HL7 message by our library; (6) The replying HL7 is sent back to the client application as the result of Web service; (7) The client reform the response HL7 by the same library, hence, the desired information is retrieved and ready to be rendered for different UI or specified usages.

TABLE I. SUBSYSTEM MODULES OF NEW HIS IN NTUH

Subsystem names	Functionalities
Finance	Billing and charging information of medical services. Cost analysis service for hospital operation management
Pharmacy	Basic medicine information and drug controlling services
RadScheduling	Scheduling of all medical imaging services. Ex: radiology, CT, MRI...
Order	Medical order recording and analysis services
Observation Report	Laboratory report retrieving service
Patient Registration	Patient registration and clinical service schedule services
Scheduling	Clinical staff service time scheduling

Note that these are only a part of subsystems in our HIS

IV. DEPLOY AND IMPLEMENTATION

The NTUH HIS is deployed in subnet of the private hospital network. The server farm is protected by a firewall, and is separated from end user machines, as well as the legacy information systems. The 4-tier architecture of NTUH HIS is

shown in Fig. 2. The four tiers are web browsers, web application servers, data exchange servers, and databases. Components and flows shaded with gray color are implemented based on the proposed framework. The HL7/Web service middleware framework provides a solid solution for data communication between tire and data exchange tire.

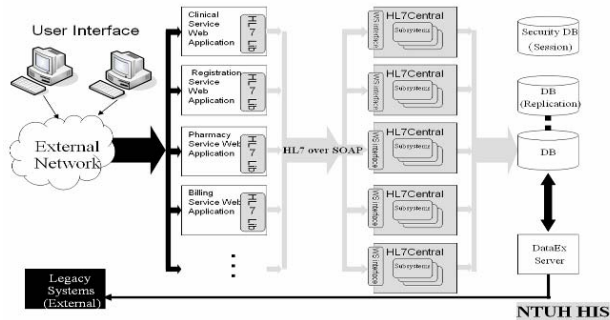


Figure 2. NTUH HIS system architecture Note

To increase the performance of the NTUH HIS, a farm of multiple HL7Central servers are deployed. All of them are configured running in load balance mode or failover mode to ensure the system availability. The system configuration can be adjusted with interrupting client applications that are currently serving. Also, any new or modified service can be slated into the system and reuse the existing Web service interfaces easily.

V. PERFORMANCE ANALYSIS

The new HIS of NTUH entered fully service on 2006/1/2. Before the system was deployed, the most concerning and arguable issue is whether the overhead of XML Web services could slow down the data communication speed between servers of different tiers. After running our new HIS in NTUH for three months, the collected data indicates our new HIS works smoothly without any unacceptable response delay.

In Table 2, number of out-patients serviced from 2006/02/13 to 2006/02/18 is listed. Average round-trip response time of HL7 data exchange is demonstrated in Fig 3. From these statistics, it shows the average response time for single transaction is less than 0.2 sec, a very reasonable speed for any end-user.

TABLE II. OUT-PATIENT STATISTIC FROM 2006/02/13 TO 2006/02/18.

Patient Date	Total outpatient completed	Total outpatient registrations
2006/2/13 Mon	6838	7966
2006/2/14 Tue	7183	8285
2006/2/15 Wed	6903	7929
2006/2/16 Thu	5978	6974

2006/2/27 Fri	5992	7012
2006/2/28 Sat	1624	1912

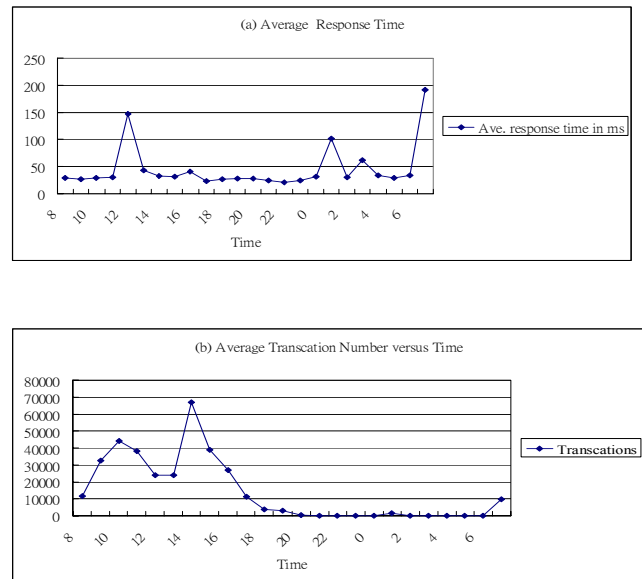


Figure 3. Average response time and transaction numbers from 2006/2/13 to 2006/2/18 (a) average response time (in ms); (b) average transaction numbers

VI. CONCLUSION AND FUTURE WORKS

In this paper, a middleware framework for Healthcare Information System using HL7 standard and state of the art XML Web service is presented. The HL7/Web service framework has successfully provided an effective solution for the challenges of integrating various and heterogeneous information systems while developing a large scale HIS. A new HIS based on the proposed framework is developed and currently serving about 8,000 outpatients a day in NTUH. Further development of the framework is also under study including using HL7 3.0 in our system and migrating our inpatient user requirements into the new framework.

REFERENCE:

- [1] "Health Level Seven (HL7,)" HL7 Standard v.2.5, 2003.
- [2] HL7 Recommendation: Using XML as a Supplementary Message Syntax for HL7 Version 2.3.1, Health Level Seven Organization <http://www.hl7.org>
- [3] World Wide Web Consortium (W3C), "SOAP specifications," <http://www.w3.org/TR/soap/>, Feb, 2006.
- [4] World Wide Web Consortium (W3C), "WSDL Note Version 1.1," 15 March, 2001, <http://www.w3c.org/TR/wsdl/>
- [5] T.H. Yang, P.H. Cheng and etc, "A Scalable Multi-tier Architecture for the National Taiwan University Hospital Information System based on HL7 Standard," CBMS 2006, the 19th IEEE International Symposium on Computer-Based Medical Systems.
- [6] B. Blobel, "Authorization and Access Control for Electronic Health Record System," Int'l J. of Medical Informatics, vol. 73 pp251-257
- [7] P. H. Cheng, C. H. Yang and etc, " Application of HL7 in a Collaborative Healthcare Information System," Proceeding of the 26th Annual International Medicine and Biology Society, pp 3354 – 3357, Sept, 2004

- [8] J. David and Don Van Slyke, "The IBM Patient Care System: An Overview of its Software," SIGMIS Databases, 1980, Vol. 11, pp. 64 - 75
- [9] Tung Bui and Alexandre Gachet, "Web Services for Negotiation and Bargaining in Electronic Markets: Design Requirements and Implementation Framework", Proceedings of 38th Hawaii International Conference on System Sciences, 2005
- [10] B. Blobel and M. Holena, "Comparing Middleware Concepts for Advanced Healthcare System Architectures," Int'l J. of Medical Informatics, vol. 46, no. 2, pp. 69-85, Sept. 1997.