

# Using a Health Level 7 Interoperability Bus to Support Legacy Systems in the Health Domain

Rafael Andrade<sup>1,2</sup>, Aldo von Wangenheim<sup>1</sup>, Alexandre Savaris<sup>1</sup>, Karine Petry<sup>1</sup>

<sup>1</sup>INCoD, Brazilian Digital Convergence Institute, Federal University of Santa Catarina—UFSC, Florianópolis, Brazil

<sup>2</sup>Federal Institute of Santa Catarina—IFSC, Florianópolis, Brazil

Email: [andrade@telemedicina.ufsc.br](mailto:andrade@telemedicina.ufsc.br), [awangenh@inf.ufsc.br](mailto:awangenh@inf.ufsc.br), [savaris@telemedicina.ufsc.br](mailto:savaris@telemedicina.ufsc.br), [karine@telemedicina.ufsc.br](mailto:karine@telemedicina.ufsc.br)

Received December 3, 2013; revised December 20, 2013; accepted December 30, 2013

Copyright © 2013 Rafael Andrade *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## ABSTRACT

In order to reduce the effort in the integration and actualization of heterogeneous healthcare legacy systems that should share a common database, we propose the creation of an interoperability bus using the HL7 standard—the HL7Middleware. This interoperability bus is an intermediate layer responsible for the communication between a database, health information systems and medical equipment, called HL7Server. Connected systems use the HL7 messaging semantics to store and retrieve data from the database. We validate our approach with respect to two different criteria: performance and integration costs. Benchmark tests were executed with and without the use of HL7Middleware and with different network bandwidths. These results demonstrated that the performance of the interoperability bus is higher when compared to traditional database access for larger volumes of data and when the bandwidth of the user is considerably lower than the bandwidth of the connection between HL7Server and database. The overall development and deployment cost was considered low and the reusability degree of wrapper code was considered high, thus suggesting a progressive reduction of the integration costs of additional services and subsystems of an organization.

**Keywords:** Middleware; Healthcare; Interoperability Bus; Legacy System; HL7; Telemedicine

## 1. Introduction

Nowadays, in a scenario of large hospitals or healthcare organizations, it is still common to find independent heterogeneous information systems and usually isolated from each other. These systems can be associated with medical devices, such as PACS (Picture Archiving and Communication System), RIS (Radiology Information System), or special purpose systems that were developed originally in a gradual and uncoordinated informatization process, performed without concern with the integration between all these information systems. This heterogeneity is composed of isolated legacy systems operating in parallel. Typically, this model disables the sharing and integration of healthcare information and, consequently, imposes the duplication of information, like the identification of a particular patient, resulting in redundant data and rework.

An example of an institution with such an informatization profile is the *Prof. Polydoro Ernani de São Thiago* University Hospital of the Federal University of Santa Catarina (HU/UFSC), in *Florianópolis, Santa Catarina*—

Brazil. The HU/UFSC (<http://www.hu.ufsc.br>) is a large public hospital founded in 1980 with 1600 professional staffs that receive over 11,000 patients per month and serve as the school-hospital attached to UFSC. The informatization process was started in 1984 when the hospital received a system composed of a super minicomputer model COBRA 480 and a set of terminals. This system was based upon a proprietary platform produced by *ComputadoresBrasileiros*, a military-owned Brazilian hardware producer [1]. The terminals were installed in the hospital, and quickly, several information systems were developed in a proprietary database management system (DBMS) solution for the COBRA SOD-operating system. This platform was enhanced and later replaced by various initiatives in several areas and administrative sectors from the hospital. With each of these solutions evolved, today the HU/UFSC has:

- A legacy hospital management system (HMS) that offers also patient admittance and electronic health record (EHR) services for about 55% of the hospital. This system was developed as a client/server solution

on the Gupta/Centura [2] platform;

- A web-based EHR system that interacts with different examination acquisition client tools and supports the hospital image services, covering obstetrics, radiology, colonoscopy, cardiology and others. This database is integrated into the EHR of other 401 health institutions distributed among 291 municipalities of the Santa Catarina State Public Telemedicine Network (RCTM) [3];
- A large set of isolated solutions, such as a Toxicology Emergency System, a Pharmacy Management System, and many others. All information systems operate in an isolated manner with data redundancy.

Our experience is that such a situation is much more common than one would expect, mainly in large health institutions with a long history of isolated informatization initiatives. It represents a major hindrance to an integrated informatization of the health institution as a whole. This occurs also because the substitution of legacy systems implies in deep changes in the operational culture and personnel retraining, besides errors and possible data loss during the data transfer.

The use of an interoperability standard allows any system to interpret its local data using universal concepts and terminologies [4]. This definition motivates the idea of using a healthcare interoperability standard to integrate different systems and, at the same time, to use these concepts and terminologies to access data internally, operating on the own database. In order to represent requests and queries through message passing, a standard, such as HL7, maintaining the compatibility and consistency across systems and database, is employed.

Previous work has been done in the field of the use of an intermediate layer to allow communication with a database through specific XML queries [5-7]. More specifically, some authors employed UN/EDIFACT and HL7 v. 2.5 in order to allow the integration of healthcare systems [8,9].

Differently from previously published work, we propose a middleware functioning as an independent interoperability bus. It is based on the semantics defined by a standard, as an abstraction layer to represent and manage all database access. For this purpose, we chose HL7 v. 3.0 [10] for the semantic encoding because it is a standard used in application software, with a large set of services. The HL7 standard also shares data through the XML syntax for message encoding, thus facilitating the development of interfaces and wrappers for legacy systems.

The main objective of this work was to develop a workable solution approach in order to bridge interoperability problems through data and services integration model with low impact. Our approach should support the progressive integration of legacy applications into a cen-

tral system, acting as a “data hub” to all involved legacy systems. In order to send and retrieve healthcare information from a central database, an HL7Middleware was projected. This interoperability bus employs the semantic interoperability provided by the HL7 Standard. HL7 clients, legacy healthcare systems and medical devices can send messages based on the HL7 v.3.0 standard to access the EHR. For this purpose, the system acts as a message server and communication hub, the HL7Server. It allows the integration of legacy systems transparently to the database structure, ensuring a higher level of security and consistency between systems.

This new database access strategy offers also the advantage of speeding up the development of ever expanding healthcare information systems. Even if the integration of legacy systems is not an issue, our approach provides a database abstraction level, which allows for the change or addition of requirements without implying in the addition of database tables or field changes. When using the HL7Server, it is not necessary to rewrite the queries performed by new clients at the database level. Consequently, development risks and time can be reduced and the usage of a semantics-based interoperability philosophy can also render code more readable. Considering that a healthcare interoperability standard must provide syntax and semantics to describe the sharing of health information, we will present a new usage of the HL7 standard. In order to describe all database access services, independently if they are originated from legacy systems, we use HL7 clients to allow heterogeneous healthcare systems interoperability without the need to connect directly to a specific database.

## 2. Objectives

This work was performed with the main objective to develop interoperability architecture to allow the integration of legacy systems in health institutions in a simple, efficient manner. We created a security abstraction level that avoids the legacy systems receive data direct access from a shared database. The model was developed to allow the intra- and inter-institutional systems integration, supporting the integration of services through the Internet. In our studies we also developed a framework that allows the easy and fast development of interfacing structures for legacy systems. We called this architecture *HL7Middleware* and it is composed by two components: an *HL7Server*, which communicate a central data repository and different healthcare software and a set of *HL7Wrappers*, which implement the interface to particular legacy systems. In this paper we discuss the HL7Server architecture, the HL7Wrapper development philosophy, costs, performance and the overall communication concept.

Our approach was evaluated under two different viewpoints:

**1) Performance:** in order to measure if the introduction of an intermediate layer through the HL7Middleware would represent a bottleneck, we carried out a set of performance tests with and without the HL7Middleware. We implemented queries with different bandwidth limitations and we measured the response times;

**2) Integration costs:** in order to evaluate the integration costs of legacy systems using this approach, we monitored and measured the effort for the development, tests and deployment of HL7Wrappers. For this specific purpose, we integrated, in a monitored experiment, two different legacy systems with an inter-institutional central data repository.

### Validation Context

This approach was validated in a web-based EHR context called Integrated Telemedicine and Telehealth System (STT/SC), developed by HU/UFSC and used in various cities, mainly imaging services and also by the services from the RCTM. The STT/SC shares a common database with PACS and desktop systems developed at UFSC and used at various hospitals in the Santa Catarina state. As an integration and healthcare information sharing policy, it was defined that the HU/UFSC would progressively adapt all its services in order to transfer all healthcare data to the STT/SC.

## 3. Material and Methods

In this session we present the HL7Middleware architecture, the main features and the methods employed on our

approach.

### 3.1. Architecture

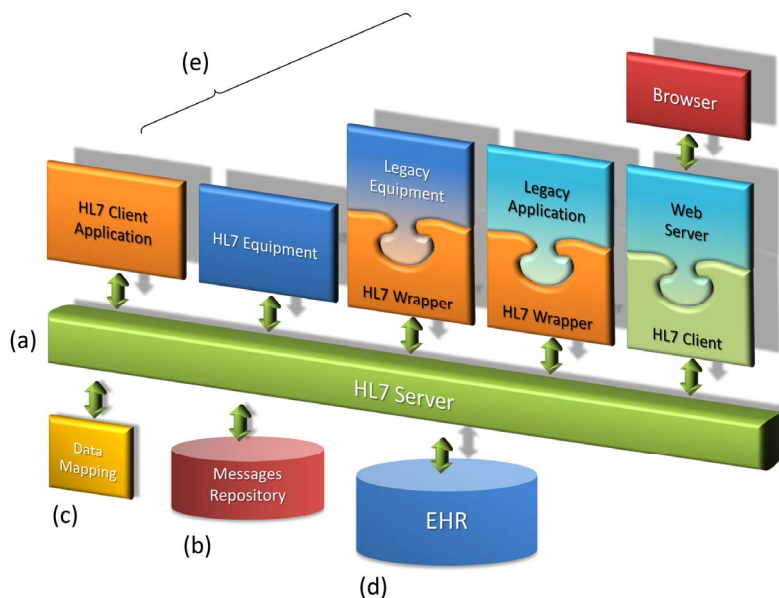
We developed an architecture adherent to the HL7 standard. HL7 employs a messaging concept to represent events associated to the healthcare process, such as, e.g., *patient admission*. In HL7 v.3.0 messages are coded as XML documents. These documents are built accordingly to a generic information model called the Reference Information Model (RIM) [8].

The main components of our architecture are described in **Figure 1**. The semantics defined by the HL7 standard are maintained in the database access approach of our HL7Middleware. The HL7Middleware architecture consists of: (a) a server that operates as an HL7 messaging bus; (b) a set of message templates and stored procedures representing database access functions associated to each message category; (c) a mapping of messages, fields and stored procedures to a database; (d) a database or set of databases; and (e) client systems of different kinds.

#### 3.1.1. HL7Server

The HL7Server message server operates as an abstraction layer between a client system and a database. The HL7Server is responsible for receiving and interpreting HL7 messages, performing database operations and sending responses to the client system as HL7 messages. This server is the only instance able to include, alter or retract data from the database.

The HL7Server implements a queue for incoming messages and communicates through sockets. Each time a new message arrives at the queue, a parsing process is



**Figure 1.** Schematic vision of the HL7Middleware concept.

started which verifies the syntactic consistency. The type of all incoming well-formed messages is then identified and the kind and structure of information to be retrieved or stored are identified in a mapping table. Parameters are generated for the associated stored procedure and the server opens a connection to the database. The procedure is executed and the connection is immediately closed. Depending on what is returned, a mapping of the data to a return HL7 message is performed through the inclusion of data elements in the adequate message template. The message is included in a send queue and sent when a connection to the HL7 partner can be established [10].

### 3.1.2. HL7 Clients

Any system that implements HL7 v.3.0 messaging can act as a client, from native HL7 clients to HL7-compliant equipment or other software can connect through adaptors. Even web-based applications can rely on the HL7Server: in this case the web-server acts as an HL7 client and implements HL7 messages whenever data has to be retrieved or stored.

The client/server communication between non HL7-compliant legacy systems and the server has to be mediated by adaptors, which translate the output of these systems to HL7 syntax and back. These adaptors work and are employed in the same way as do database wrappers. In programming, a wrapper is a component, which changes the interface of an existing component in order to increase its compatibility with various other systems, so we called our adaptors HL7 wrappers. HL7 wrappers can be implemented in two different ways, depending on how much access one has to the legacy system:

- **HL7 extension wrappers** can be implemented when access to the source code is possible and the legacy platform supports implementing TCP/IP communication. In this case, an extension to the code of the legacy system or an external library can be developed and used to compose, send, receive and parse HL7 messages;
- **HL7 external wrappers** can be employed whenever access to the source code is not available or the platform of the legacy system does not support networking programming. In this case, file-based communication is performed with an external wrapper process, that monitors a directory tree or local database where the legacy system records its actions.

### 3.1.3. Message Repository and Database Abstraction Philosophy

Due to the complexity of HL7 messages, which could be considered a limiting factor in the development of wrappers and communication routines, an *HL7message template* was generated for each messages supported by the HL7Server [10]. A message template is an XML docu-

ment containing special comments that indicate which information has to be filled in each HL7 message element. This allows an adaptor developer to abstract from the syntax of the message and to focus on the semantics of the message elements. The HL7Server manages a *dynamicrepository* of HL7 message templates and links to their associated *stored procedures*, as shown in **Figure 1**, which allows the vocabulary to be dynamically extended during runtime, without restarting the server. A stored procedure is a set of SQL commands that has been compiled and stored on the database server.

Once it has been “stored”, client applications can execute it over and over again without sending it to the DBMS again and without compiling it. These aspects of the architecture also provide an abstraction from the actual database, since the database is visible only through the vocabulary implemented by the HL7Server.

Additionally, the HL7Server uses a Data Mapping module, which employs XML documents to relate message elements to fields and stored procedures of a given database, as shown in **Figure 1**.

## 3.2. Performance Evaluation

In order to compare the performance of this interoperability bus against a traditional architecture we employed two different test settings that emulate the behavior of our EHR system [11]. One system operated directly on the PACS database and another routed all communication through the HL7Server. In order to measure the latency time between request and response, we defined four different network bandwidths (128 kbps, 400 kbps, 1 Mbps and 1 Gbps). Each request represents a set of 100 accesses into the database and was sent in batch mode for image retrieval and patient data storage.

For a better performance evaluation under different kinds of data compositions, we selected four different examination modalities for each of the batches of 100 access requests, presenting different image sizes and quantity: Computed Tomography (CT): 25 examinations, containing approximately 48 slices each; Nuclear Medicine (NM): 25 examinations, containing approximately six images each; Electrocardiogram (ECG): 25 examinations, containing approximately two images each; Endoscopy (SC): 25 examinations, containing approximately two images each.

For guarantee most similar database access times for both settings, the HL7Server was installed in the same machine as the web server.

### 3.2.1. Set #1: Without HL7Middleware

In this scenario we simulate various browsers accessing and querying our system in a controlled manner. We developed an HTTP client for the purposes of this valida-

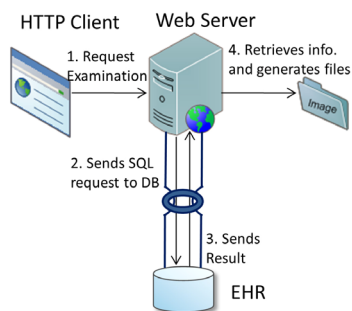
tion experiment. This client emulates a browser accessing the HTTP server, as shown in detail in **Figure 2**, and send 100 examination result access requests under each of the four bandwidths.

The processing sequence in **Figure 2** is:

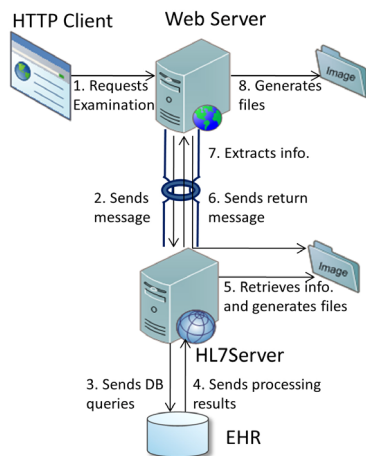
- 1) **Request examination data:** sends examination ID from the examination list to the web server. The examination list contained 100 IDs from 4 different modalities;
- 2) **SQL request to DB:** a PHP procedure at the web server opens a connection to the database, composes and performs an SQL query;
- 3) **Sends result:** result of the SQL query is sent to the web server;
- 4) **Retrieve infos:** the web server retrieve results, decodes and generates local files for each image. Resulting data is sent as HTML data to HTTP client.

### 3.2.2. Set #2: Using the HL7Middleware

For the tests using the *HL7Middleware*, we employed a web server that contained an embedded HL7 client, implemented in PHP, responsible for the communication with the *HL7Server*. The data requests were performed by the same HTTP client, which sent 100 examination IDs under each of the four bandwidths, emulating a web browser. **Figure 3** shows the communication during this test in detail.



**Figure 2. Test set #1—without HL7Middleware.**



**Figure 3. Test set #2—employing HL7Middleware.**

The processing sequence in **Figure 3** is:

- 1) **Request examination data:** sends examination ID from the examination list to the web server. The examination list contained 100 IDs from four different modalities;
- 2) **Send message:** an HL7 message is composed, a socket is created and a connection to the HL7Server established. Message is sent;
- 3) **Send DB query:** server accepts connection, receives HL7 message and parses it, fills in parameters and calls the associated stored procedure;
- 4) **Send back processing result** of the stored procedure;
- 5) **Retrieve data and generate file:** parses query results decodes images and saves in a public directory;
- 6) **Send return message:** composes return HL7 message and sends to the HL7 client;
- 7) **Extract infos:** receives return message, parses XML and retrieves images from a public directory;
- 8) **Generate file:** web server generates local files for each retrieved image. Resulting data is sent as HTML data to HTTP client.

### 3.3. Wrapper Development Effort

In order to acquire data about the effort involved in the development of HL7 wrappers, we monitored the development effort of two different wrappers connecting two different systems: an **external HL7 wrapper** connecting the legacy hospital management system (HMS) to the *HL7Server*; an **extension HL7 wrapper** connecting a PACS client for Unix/Linux platforms that is used in different hospitals of the RCTM to acquire image data from non-DICOM devices, to the *HL7Server*.

The requirements for wrapper development in these cases represented what we considered typical examples for each kind of wrapper: (a) an *external wrapper* for a legacy system developed in an obsolete and limited programming language using communication through files and (b) an *extension wrapper* for a modern but non-HL7 UNIX application developed in C++. We considered that monitoring the development, tests and deployment effort associated with these two wrappers should be representative of the development effort associated with wrapper implementation for our approach.

The effort was measured through standard software measurement techniques and quantified in terms of person/hour.

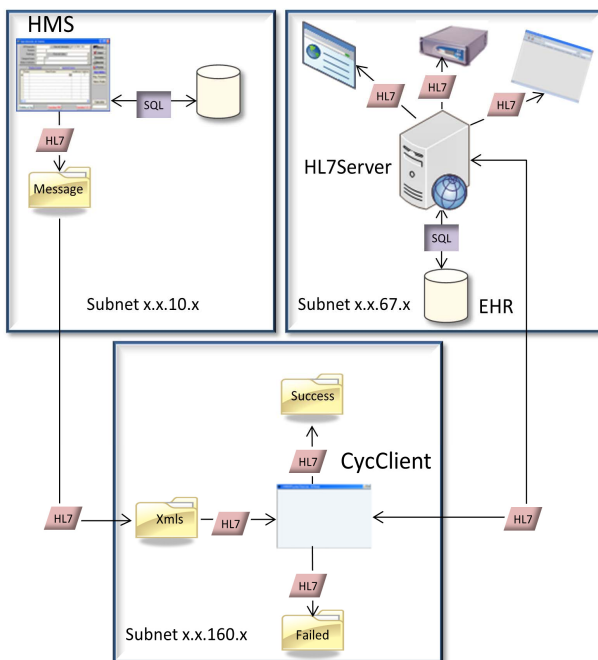
#### 3.3.1. The External Wrapper

In order to implement message sending and receiving in the HMS, we developed an external HL7 client, running as an independent process, to perform communication with the *HL7Server*. This was necessary because the Gupta/Centura platform does not offer support for ex-



licit sockets programming. The communication process is described below and depicted in **Figure 4**:

- **Message composition:** as Gupta/Centura does not support XML document manipulation such as XPath, we use a set of HL7 message templates, where a specially developed static procedure is responsible for inserting values at runtime.
- **Message coding:** as the HMS Sybase database employs the CP-850 encoding scheme to store information, it is necessary to translate it to the UTF-8 encoding scheme used in HL7 messages. The new message generation procedure of the HMS only fills the message template with adequate values; the HL7 client performs the transcoding afterwards.
- **Message saving and transfer:** as the HMS operates on a different sub networks than the *HL7Server*, a directory was created that is constantly monitored by a process that copies all messages generated by the HMS on the x.x.10.x subnet to the x.x.160.x subnet in a directory called “XmIs”;
- **Message sending and receiving:** a special HL7 client (*CycClient*) was developed to act as awrapper and send the messages in XMLs to the *HL7Server* and receive responses. For this purpose the *CycClient* monitors the XmIs directory and reads, parses and sends any well formed messaged found there. Messages received from the *HL7Server* are either stored in a directory called “Success” or “Failed”, depending on if they represent a true response from the server or an error message.



**Figure 4.** Steps for communication process between HMS and HL7Server.

### 3.3.2. The Extension Wrapper

The *CycDCM* is an image and video capturing application that acts as a Radiological Information System (RIS)/DICOM client. It codes the captured examination data together with demographic data informed by the operator as DICOM objects and sends to a central PACS, the *CycDCM*. The server automatically replicates all demographic data in the EHR database of the STT and also generates JPEG or MPEG renderings of the examinations and stores in the EHR database. The *CycDCM* can also be used as a RIS client for providing on-site findings reports. In this case, it sends the data directly to the EHR database employing an SQL query.

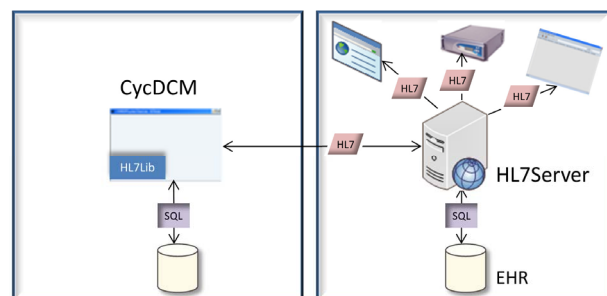
Here we implemented an HL7 client as a library. All data intended for the EHR database will be stored directly by the HL7Server, instead using the PACS for direct database access, as shown in **Figure 5**.

## 4. Results

The benchmark tests showed the result that the performance of the HL7Middleware is superior if compared with traditional database access for larger data volumes and when the user’s network bandwidth is considerably inferior to the bandwidth of the HL7 server-database connection. **Table 1** shows measured mean latency times when using traditional database access, comparing to the HL7Middleware. The traditional database access showed considerably lower latency times when a Gigabit bandwidth was employed. As bandwidth deteriorates, HL7-Middleware performance becomes comparatively better.

**Figure 6** shows the mean total latency times for the 128 kbps, 400 kbps, 1 Mbps and 1 Gbps bandwidths.

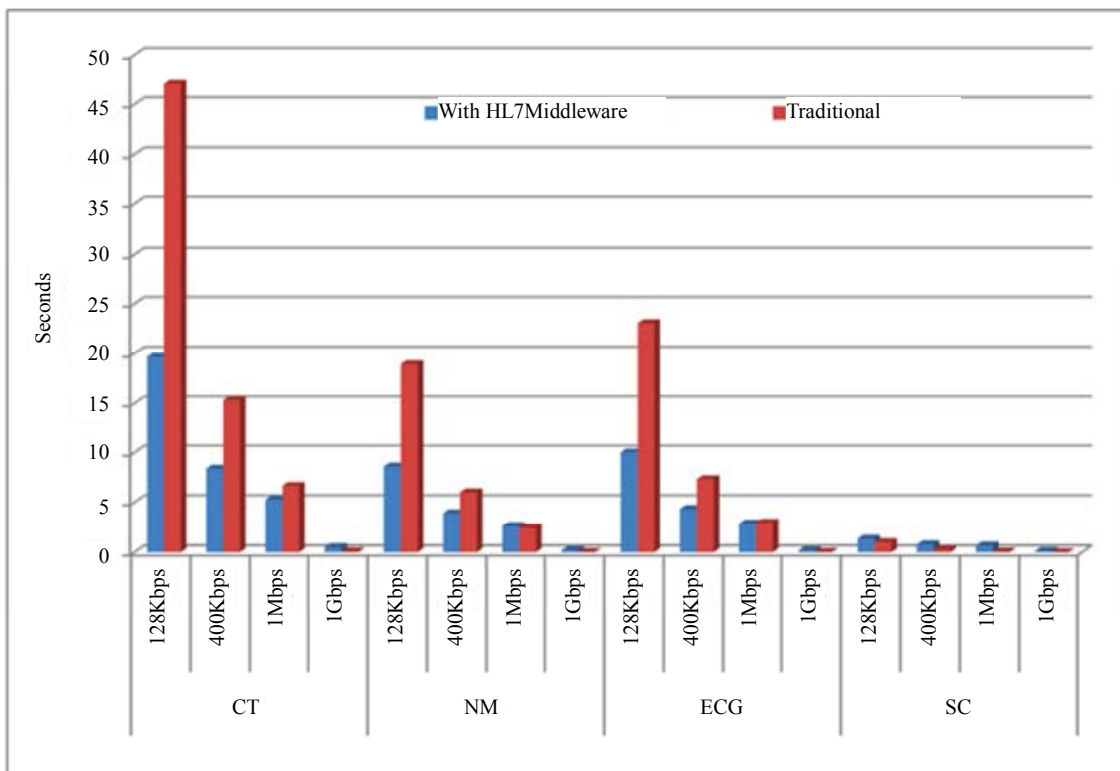
For each image retrieved from the database, the HL7Server converts the data from base64 representation to binary, creates, stores a file in a public directory and includes a link into the return message sent to the HL7 client. The client system must access the image and copy it locally. The bandwidth represents an additional bottleneck that is not present when using the traditional database approach because the DB client performs this process locally after the data is received.



**Figure 5.** Steps in the communication process between CycDCM and HL7Server.

**Table 1. Comparing HL7Middleware with traditional database access.**

Modality	Bandwidth	HL7Middleware	Traditional	Difference	Ratio
CT	128 kbps	19.66	47.15	-27.49	0.42
	400 kbps	8.38	15.33	-6.95	0.55
	1 Mbps	5.30	6.67	-1.37	0.80
	1 Gbps	0.51	0.09	0.41	5.54
NM	128 kbps	8.60	18.93	-10.33	0.45
	400 kbps	3.92	6.03	-2.11	0.65
	1 Mbps	2.61	2.42	0.19	1.08
	1 Gbps	0.20	0.05	0.16	4.47
ECG	128 kbps	10.01	23.07	-13.06	0.43
	400 kbps	4.34	7.34	-2.99	0.59
	1 Mbps	2.86	2.97	-0.11	0.96
	1 Gbps	0.18	0.05	0.13	3.82
SC	128 kbps	1.36	1.00	0.35	1.35
	400 kbps	0.79	0.28	0.50	2.79
	1 Mbps	0.67	0.08	0.60	8.75
	1 Gbps	0.13	0.01	0.12	21.57



**Figure 6. Mean total latency times.**

**Table 2. Development effort to implement the wrappers.**

System	Kind	Effort	Description
HMS	Legacy application + HL7Wrapper	182 man/hour	150 man/hour by develop the CycClient; 20 man/hour for build the messages; 12 man/hour for deployment and testing
CycDCM	Legacy application + HL7Wrapper	46 man/hour	16 man/hour for build the messages; 30 man/hour for development the library.
STT	HL7ClientApplication	30 man/hour	For develop an application of <i>HL7Client</i> attributes.

Since the HL7Server is connected to the database through our local gigabit network, the database response time to queries from the HL7Server is considerably lower. Even with the additional processing generated by HL7 messaging, the HL7Server performance exceeds direct access to the database by users connected through bandwidths such as 1 Mbps or 400 kbps.

The *HL7Server* converts the images to the final binary format and the images are sent in base64 encoding. This explains why connections using the HL7Middleware were faster, even if more processing is involved. The traditional database access was faster only when DB client/DBMS bandwidth was similar to the HL7 client/HL7Server bandwidth.

In order to quantify the effort of the integration of a legacy system using the HL7Middleware, we recorded the development and deployment time of this wrapper. The total effort to develop and deploy the necessary connection software has been about 182 persons/hour. About 150 persons/hour of this effort were considered to be directly reusable on the integration of other services at the hospital. This process indicates a probable progressive reduction of the integration of further services or subsystems in an organization. **Table 2** shows the necessary effort to be developing the wrapper.

## 5. Discussion and Conclusions

In many clinics, hospitals and large health care networks, which have a quite old computer process and grow wildly, to change these systems is not easy. Due to the large amount of heterogeneous systems, the difficulty in the change and also other services that are essential and can not be stopped, connecting these different systems is a major challenge of computer systems. An advantage of the healthcare domain is to possess a general-purpose of a communication channel represented by the HL7 standard. We consider the better practical path to be an integration approach for such legacy systems. Traditional HL7-based solutions are intended to be peer-to-peer solutions.

We developed a slightly different HL7-based client-server model expressed as an HL7 interoperability bus, where all communications between systems are routed through a central server and also key information can be centrally stored for common access. This approach offers

a general-purpose interface where legacy systems can be connected. Wrappers that monitor these systems or, when possible, are invoked to perform integration of legacy systems, and make possible also the systems integration where source code is not available and modifications cannot be performed on the system. The test results showed that the performance of the HL7Middleware model is satisfactory. But an unexpected side effect was achieved. Due to the more compact image-encoding scheme used, the communication was faster using the HL7Middleware when bandwidth represented a bottleneck.

The time spent on the wrapper development and deployment was considered low because of the benefits provided by the standardization of database access. *HL7 Middleware* architecture allows for programming language, database and operating system independence. The interoperability bus also allows the integration with medical equipment that already uses the HL7 standard and heterogeneous healthcare systems interoperability.

From the necessary effort point of view, it is important to observe that the external HL7 wrapper *CycClient* that had to be developed is a generic HL7 client tool able to read XML document files composed by legacy systems. This client sent HL7 messages to an HL7 server using the HL7 application protocol. The development was necessary for the integration between the STT and the HL7Server, but the 150 persons/hour efforts involved in its development must not be spent again to the implementation of further communication channels. Only the 32 persons/hour effort for the coding of additional messages would be necessary to implement more services. This turns out to be a very interesting effort.

## REFERENCES

- [1] P. B. Tigre and P. Evans, "Going Beyond Clones in Brazil and Korea: A Comparative Analysis of NIC Strategies in the Computer Industry," *World Development, London*, Vol. 17, No. 11, 1989, pp. 1751-1768. [http://dx.doi.org/10.1016/0305-750X\(89\)90198-8](http://dx.doi.org/10.1016/0305-750X(89)90198-8)
- [2] P. Sanjiv, "Using SQLWindows and Centura: Techniques for Building Client/Server Solutions," Wiley, Hoboken, 1996.
- [3] A. von Wangenheim, C. L. Barcellos Jr., H. M. Wagner



- and C. C. Gomes, "Ways to Implement Large Scale Telemedicine: The Santa Catarina Experience," *Latin-American Journal of Telehealth*, Vol. 3, No. 1, 2009, pp. 364-377.
- [4] A. Onabajo, I. Bilykh and J. Jahnke, "Wrapping Legacy Medical Systems for Integrated Health Network. Migration and Evolvability of Long-Life Software Systems," In: *Proceedings of Net Object Days Workshop on Migration and Evolvability of Long-Life Software Systems*, Springer Lecture Notes, Erfurt, 2003.
- [5] F. M. Al-Wasil, W. A. Gray and N. J. Fiddian, "Establishing an XML Metadata Knowledge Base to Assist Integration of Structured and Semi-Structured Databases," In: *Proceedings of the 17th Australasian Database Conference*, 49, ACM International Conference Proceeding Series, Hobart, Vol. 170, 2006, pp. 69-78.
- [6] S. Chan, T. Dillon and A. Siu, "Applying a Mediator Architecture Employing XML to Retailing Inventory Control," *The Journal of Systems and Software*, Vol. 60, No. 3, 2002, pp. 239-248.  
[http://dx.doi.org/10.1016/S0164-1212\(01\)00095-4](http://dx.doi.org/10.1016/S0164-1212(01)00095-4)
- [7] S. R. Collins, S. Navathe and L. Mark, "XML Schema Mappings for Heterogeneous Database Access," *Information and Software Technology*, Vol. 44, No. 4, 2002, pp. 251-257.  
[http://dx.doi.org/10.1016/S0950-5849\(02\)00015-0](http://dx.doi.org/10.1016/S0950-5849(02)00015-0)
- [8] L.-F. Ko, et al., "HL7 Middleware Framework for Healthcare Information System," In: *Proceedings of HEALTHCOM 2006 8th International Conference on e-Health Networking, Applications and Services*, New Delhi, 17-19 August 2006, pp. 152-156.
- [9] Y. Xu, D. Salquet, E. Zapletal, D. Lamaitre and P. Degoulet, "Integration of Medical Applications: The 'Mediator Service' of the SynEx Platform," *International Journal of Medical Informatics*, Vol. 58-59, 2000, pp. 157-166.  
[http://dx.doi.org/10.1016/S1386-5056\(00\)00084-8](http://dx.doi.org/10.1016/S1386-5056(00)00084-8)
- [10] "HL7, Health Level Seven Web Site," 2012.  
<http://www.hl7.org/about>
- [11] J. Wallauer, D. Macedo, R. Andrade and A. von Wangenheim, "Creating a Statewide Public Health Record Starting from a Telemedicine Network," *IT Professional*, Vol. 10, No. 2, 2008, pp. 12-17.  
<http://dx.doi.org/10.1109/MITP.2008.23>