Journal of Medical Systems, Vol. 20, No. 4, 1996

Implementation of HL7 to Client-Server Hospital Information System (HIS) in the University of Tokyo Hospital

Kazuhiko Ohe[1] and Shigekoto Kaihara[1]

In developing a large-scale Hospital Information System (HIS), a client-server architecture has been gaining in popularity. It is important to introduce a standard message protocol that is independent both on the database structure and on the vender's proprietary platform. We introduced Health Level Seven (HL7) to our hospital information system. From our experiences, although we had to modify the original HL7 specifications in order to introduce the protocol to a client-server HIS especially in the area of order entry and record-oriented query, it was found that HL7 can be adopted in a client-server HIS.

KEY WORDS: Hospital Information System; HL7; client-server; communication standard; protocol.

INTRODUCTION

In developing a large-scale Hospital Information System (HIS), a client-server architecture has been gaining in popularity as it has in other information systems.[1-6] This architecture, which is based on collaborative communications between client's machines and a server, brings about good response, friendly man-machine interface, and flexibility of the client's applications. Since the communication between a client and a server is an essential and important component in this architecture, the application layer of the communication protocol often depends closely upon the database structure of the server. To develop a flexible environment for future computerized patient record systems (CPRs), however, it is recommended that the application layer of the client-server communication protocol should be as independent of both database structure and the clients' man-machine interface as possible, because there will be several multimedia servers from a variety of vendors and many clients from various or multiple platforms like Windows, UNIX, or MacOS. In this paper, we described our experiences acquired from the implemen-

[1]Hospital Computer Center, University of Tokyo Hospital, 7-3-1 Hongo, Bunkyo, Tokyo 113, Japan.

0148-5598/96/0800-0197$09.50/0

tation of Health Level Seven (HL7) into the University of Tokyo Hospital Information System.

BACKGROUND

In most of the 42 national university hospitals in Japan, the introduction of physicians' direct order entry systems, i.e., systems in which physicians enter most of the clinical orders by themselves, has been implemented both in the wards and the outpatient clinics, and this is an essential function of integrated HIS. In an encounter room for hospital outpatients, a physician must operate a computer terminal to check clinical data and to enter most clinical orders for laboratory examinations, radiology examinations, prescriptions, and so on while he or she is face to face with a patient. Since a physician often has to see over 10 patients an hour in most hospitals, response time and user friendly interface of the computer terminals are very important. Major vendors of HIS in Japan have been making efforts to discover solutions for these issues. As a result of these efforts, vendors have introduced an HIS which has a client-server architecture comprised of one mainframe as an HIS server and hundreds of personal computers as client workstations. In such a Japanese vendor's system, structures of the database on the server and the application layer of the communication protocol between clients and the server are tightly coupled with each other, and both are implemented according to the vendor's proprietary specifications. This situation is an obstacle to developing future CPR systems because (1) different vendors' HIS cannot exchange patient data with each other, (2) multi-vendor workstations cannot be introduced even if a user prefers a different vendor's workstation, (3) tight coupling between database structure and the vendor's proprietary communication protocol means the systems are both inflexible and incapable of extending the database structure. To resolve this situation, some standard protocol between a client and a server should be introduced. HL7 is one of the standard protocols often used in health care information systems.[7-10] We decided to adopt Health Level Seven in the client-server communication protocol in our new hospital information system, replacing the former HIS in the University of Tokyo Hospital.

INVESTIGATION OF APPLICABILITY OF HL7 TO CLIENT-SERVER HIS

HL7 is a protocol that was at first proposed as a standard communication protocol in the application layer (seventh layer in OSI communication model) between loosely connected independent computer systems in health care information systems.[11] However, HL7 could be applied to the communication protocol in a client-server HIS because it defines what kinds of data should be transferred by a certain message. The advantages of applying HL7 to a client-server HIS is that (1) on the server machine, a software module that is developed for the communication between the clients and the server also can be used for the communication between other auxiliary systems (departmental systems) and the server, (2) it is not necessary to create an original protocol only for the client-server communications, (3) open systems can be introduced because the client-server communications would be independent of their server architecture. The last advantage is especially important so that the HIS might be flexible and extendable to a future CPR system.

On the other hand, before adopting HL7 in client-server communications, there are several problems to be considered. For example, there is a potential performance problem. In client-server systems, every time a user requests clinical data from the client computer, an HL7 message is sent to the server to be processed, and the response message will be returned from the server. A short turn-around time, or response time, is a very important factor for the system to be accepted by the users. However, since there are over 400 client workstations running at a time, the overhead of processing HL7 messages on the server cannot be ignored and might cause a slow response time. A second problem is derived from the fact that the necessary messages in the client-server communications depend on the man-machine interface on the client's side. For example, if users need a function for displaying a list of patient names admitted to one nursing unit, a message for requesting such data and the format of sequence of the result fields should be defined. The third and the most important issue to be considered is that current HL7 (ver.2.2) has not sufficiently defined record-oriented query/response messages, which are essential for client-server communications. A "record-oriented query/response message" means that the expected result of a query is not preformatted for display on the terminal, but rather a simple set of the requested fields. Because the client's program can display or arrange the resulting data on its screen, all the server has to do is respond to the requested data in simple format and not arrange the display format for the user's terminal. Considering that the kinds of query messages depend on the required functions on a client terminal, we suggest that record-oriented query/response messages in HL7 should be more flexible and as powerful as the server's SQL language is in order to satisfy diverse types of data retrievals.

IMPLEMENTATION OF HL7 AT THE UNIVERSITY OF TOKYO HOSPITAL

Required Modifications to HL7 in Our Client-Server Communications

Considering the several issues of HL7 stated above, we decided to adopt a protocol quite similar to HL7 in our hospital information system. Our protocol is the same as HL7 in the basic messages but has quite different modifications to let QRY/ORF messages support a diversity of record-oriented queries/responses. Small changes were also introduced into the order-entry messages for prescriptions and laboratory tests. Furthermore, several new messages and segments were defined to support user authentication, management of database of users, and record-oriented result data. In this development phase, we named our protocol Modified Messages of HL7 (MM7) to distinguish it from the original HL7.

Query/Response (QRY/ORF) Messages

Original HL7 defines QRY messages by the following three segments; an MSH segment for a message-header, a QRD segment for a query definition that indicates what kind of information is needed, a QRF segment for a query filter. On the other hand, in SQL a typical query sentence (SELECT sentence) has a list of requested fields and follows three clauses; SELECT a list of fields FROM table.database WHERE selection-condition-lists ORDER BY ordered-fields.

We thought that a semantic structure of query messages in HL7 should have compatibility with SQL so that it might be flexible enough to support diversity of record-oriented queries, and we redefined the QRD-segment to have the same information semantically as the FROM clause in SQL; we redefined QOF to have the same information as WHERE; we introduced a new QOD-segment that has the same information as ORDER BY. This definition should reduce both the tasks of developing conversion modules between SQL and HL7 and the overhead in the server process. Also in our definition, a pair of table name and database name is represented by a three-letter code in the field of what-subject-filter in the QRD segment to avoid incompatibility with the original HL7. This method is less flexible than SQL, but has enough flexibility to support most queries in health care information systems. A selection condition is also represented by a three-letter code (field identifier, operator, and string as a value). A QRF segment consists of one or more selection-conditions concatenated with the tilde letter (~). This representation supports only the AND condition. Each record-oriented query needs a definition of new segment(s) to represent the result data corresponding to the query.

Order Entry for Laboratory Tests and Prescriptions

Generally there are two ways to order laboratory tests for physicians in Japan. One is to indicate simply the name of the battery or set of test items that are often performed together. "Order of count of blood cells (CBC)" is an example of this, because CBC consists of the orders of red blood cells, white blood cells, platelets and so on. Another way, which is much more popular in our hospital, is to indicate each name of each test item one by one instead of indicating the name of the battery even if the test item is one element of a certain battery. The documentation of HL7 ver.2.2 describes the order entry message for laboratory tests as shown in Table I.

According to this message protocol, the first method can be represented but the latter cannot, because the message has no repeatable segment which indicates every order item. This problem may be derived from the difference between Japanese and American physicians in the way they order laboratory tests. At any rate, we had to introduce new repeatable ZOX-segments to indicate each test item in one order, just as the ORF message for reporting test results has repeatable OBX-segments to indicate each test result in one order. Through examinations of all kinds of physicians' clinical orders including laboratory tests, radiological examinations, prescription orders, dietary orders, and so on, we found that there is one

Table I. The Structure of the Order Entry Messages.[a]

ORM: General Order Messages (from reference 11, Chapter 4.2.1)	
MSH	Message Header Segment
[	
PID	Patient Identification
]	
{	
ORC	Common Order Segment
[	
	Order Detail Segment OBR, etc. (OBR, RXO, . . .)
	[{NTE}]
	[
	{
	OBX Results Segment for sending referential results
	}
	]
]	
]	
}	

[a]ORM messages cannot include repeatable order items within the block of order detail segment. OBXs are the result data to be sent for the receiving system, not the ordered items.

common semantic structure which can represent all kinds of orders. The structure is shown in Table II. Every order can be represented by a common order segment (ORC), domain specific segment (OBR for tests, RXO for prescriptions, ZDT for dietary, and so on) and a repeatable ZOX segment which indicates the contents of the order. Although ZOX is a multi-purpose segment, the basic structure is the same among all kinds of orders. The advantage of introducing one common semantic structure like this is to decrease the task of developing a similar but different software module to interpret each distinct message and to allow easy adoption of another order entry message for a new kind of clinical order. Table III shows an example of a message for ordering laboratory tests.

Development of Processing Modules for HL7 in C++ Language

Using the specifications stated above, the processing program for the modified HL7 was developed in C++ language on Solaris 2.3 operating systems. Each segment and message is defined by C++'s class definition, and a class has its own member functions, which can generate a segment string or message string under the HL7 encoding rules and can decode it from a received string. By this technique, to send a certain HL7 message to the server and receive the response, a developer has only to create one instance of a message class, set necessary parameters into the member-data of the class, call the "generating and sending HL7 to the server" member-function of the class, and call the "receiving from the server and decoding" member functions. A whole message is generated as following; a message-generating function of a message class calls each segment-generating function of each seg-

Table II. Repeatable ZOX Segments Were Introduced for Sending the Ordered Items.[a]

ORM: General Order Messages (modified in our hospital)

```
MSH       Message Header Segment
[
  PID     Patient Identification
]
{
  ORC     Common Order Segment
    [
          Order Detail Segmen OBR, etc. (OBR, RXO, . . .)
          [
            {
            ZOX order item segment
            }
          ]
          [{NTE}]
          [
               {
               OBX Results Segment for sending referential results
               }
          ]
    ]
}
```

[a]A ZOX segment has the following five fields; SetID, Value-Type, Item-ID, Item-SubID, Item-Value. This structure is compatible with the structure of an OBX segment.

Table III. Example of a Message for Ordering Four Items of Laboratory Test[a]

MSH	Message Header Segment
PID	Patient Idenification
ORC	Common Order Segment
OBR	Detail Segment, e.g., information of blood sample.
ZOX	Order item1, e.g., WBC
ZOX	Order item2, e.g., RBC
ZOX	Order item3, e.g., Ht
ZOX	Order item4, e.g., Hb

[a]Information of each item is included separately in each ZOX segment.

ment and further the segment-generating function calls the field-generating functions and so on.

Hardware and Software Configuration of the HIS

In the University of Tokyo Hospital Information System, two mainframes (Fujitsu FACOM M1600/8 for clinical services and M760/10 for administration) share a 90GB hard disk as the HIS servers. The M760 is only for registrations and real time accounting of all the patients, and the M1600 is for all other real-time clinical services including retrievals of laboratory data, prescription data, and scheduling data of patients' next visits, as well as the storage of all kinds of orders for laboratory tests, prescriptions, radiology examinations, pathophysiology tests, meals, and so on. Not only physicians but co-medical staff use a UNIX workstation (Fujitsu S-4/CL, 48MB memory, 540MB hard disk, 17-in. color display) as a client terminal, which is an OEM model of SUN-Classic from the Sun Microsystems Co. Ltd. The mainframe database for clinical services is managed by Relational Database Management System (RDBMS), and the originally developed message-processing software module running on the mainframe processes the messages between the RDBMS and the clients. Since an HL7-based message protocol, which is called MM7 in this paper, is adopted between the server and the clients, the software module takes the role of converting a client's request messages from MM7 format into SQL format and vice versa. The same message protocol is also implemented between the clinical server and two auxiliary systems: the radiological information system and the pathophysiological examination system. On every client workstation, client application programs, which we newly developed in cooperation with Fujitsu Co. Ltd. to serve as a user-friendly interactive man-machine interface for the medical staff, are running under a SOLARIS 2.3 UNIX operating system including X-Window and Motif environments. Through an object-oriented analysis of required messages in the client-server HIS, MM7 message processing software was coded in C++ language.

Experience of Performances Through 18 Months' Operation

The new hospital information system in the University of Tokyo Hospital has been running since July 1994. In an average day we see 2,500 outpatients in about 250 examination rooms with one client workstation in each room. Seventy percent of the outpatients come in the morning. There are about 800 beds on 23 floors, and there are a total of 69 client workstations. The average response time for each user action is about 5 sec and the traffic of the messages on a particular day is described in Fig. 1. As the figure shows, our HIS environment requires the server to process about four messages in 1 sec. The load ratio of the CPU for processing MM7 messages was over 25% in the first year, and this load ratio was reduced to 17% by fine tuning the program.

DISCUSSION

Adopting a standard message protocol that is not proprietary is important to develop a vendor-independent client-server system. Although SQL is one of the solutions, expressions coded in SQL depends on the semantic structure of the database on a server. If a structure of a table is changed on a server, SQL statements

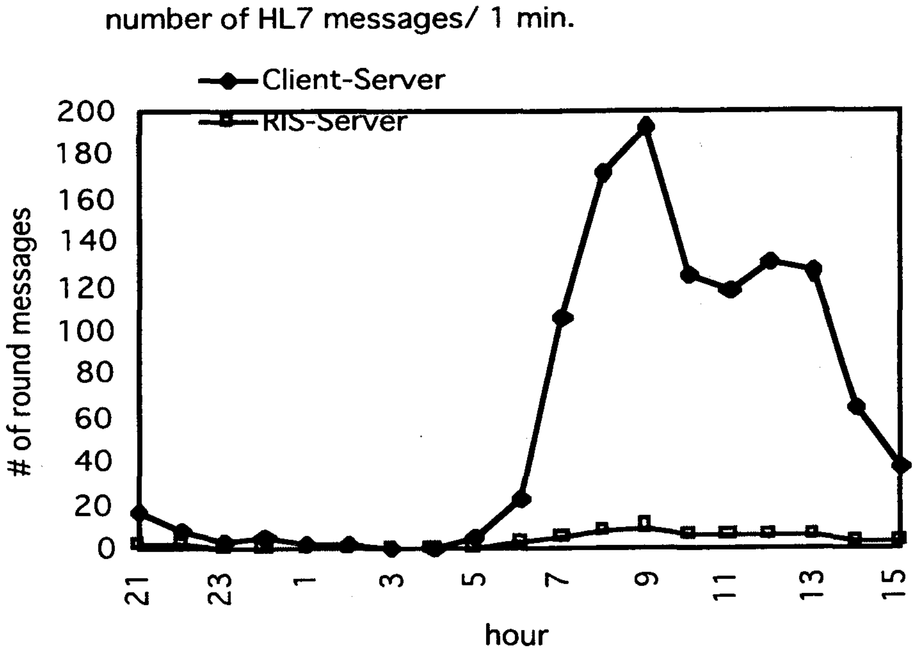

Fig. 1. The vertical axis shows the number of messages that are processed in one minute. The peak-time is from 9 to 10 o'clock, when the server has to process about 200 clients' messages in one minute. For comparison to the communication between auxiliary systems, the number of messages between radiology information system (RIS) and the server is also depicted in this figure.

associated with the table have to be modified in the client program. We needed a message protocol that did not depend on the changes of the semantic structure of the server's database and HL7 was the solution we adopted. Since it was found that the query/control features of HL7 were not powerful enough to use in the client-server environment, reconstructions and extensions to the query messages were required. Further, HL7 proposes different message structures for different auxiliary order entries: e.g., each message for diet order entry, laboratory order entry, and prescription order entry has a different definition of segments. This feature not only requires the time-consuming task of coding for the several kinds of programs that correspond to each structure of the order entry messages, but also requires a definition of the new message structure whenever a new auxiliary order entry system is introduced. To avoid this problem, we developed a common order entry message that is abstracted from all order messages. It might be controversial whether the modification was essential or not, though the idea was implemented in Hamamatsu Medical School in Japan as well as in our hospital. Even though the adoption of an HL7-based message protocol into a client-server HIS brought us several problems, it also brought us many advantages. The most useful advantage is that any computer using any platform can access the clinical database and only requires the C++ communication modules to be transported. In fact, we could make use of this feature to develop a Common Gateway Interface (CGI) program for the World Wide Web (WWW) server that serves patients' data to physicians' WWW browsers. From these experiences, we recognized that a standard, vendor-independent, efficient message protocol for a client-server HIS is necessary to develop future high-performance CPR systems.

REFERENCES

1. Willard, K. E., Sielaff, B. H., Connelly, D. P., Integrating legacy laboratory information systems into a client-server world. *Methods Inf. Med.* 34(3):289-296, 1995.
2. Chueh, H. C., Barnett, G. O., Client-server, distributed database strategies in a health-care record system for a homeless population. *J. Am. Med. Inform. Assoc.* 1(2):186-198, 1994.
3. Sauquet, D., Jean, F. C., Lemaitre, D., *et al.*, The HELIOS unification bus: A toolbox to develop client/server applications. *Comput. Methods Programs Biomed.* 45 Suppl:S13-22, 1994.
4. Pham, T. Q., Young, C. Y., Tang, P. C., *et al.*, Implementing a physician's workstation using client/server technology and the distributed computing environment. *Proc. Annu. Symp. Comput. Apl. Med. Care.* 626-360, 1994.
5. van Mulligen, E., Timmers, T., Beyond clients and servers. *Proc. Annu. Symp. Comput. Appl. Med. Care.* 546-550, 1994.
6. Huff, S. M., Haug, P. J., Stevens, L. E., *et al.*, HELP the next generation: A new client-server architecture. *Proc. Annu. Symp. Comput. Appl. Med. Care.* 271-275, 1994.
7. Hammond, W., The status of healthcare standards in the United States. *Int. J. Biomed. Comput.* 39(1):87-92, 1994.
8. Hammond, J. E., Berger, R. G., Carey, T. S., *et al.*, Report on the clinical workstation and clinical data repository utilization at UNC Hospitals. *Proc. Annu. Symp. Comput. Appl. Med. Care.* 276-280, 1994.
9. Hettinger, B. J., and Brazile, R. P., Health Level Seven (HL7): standard for healthcare electronic data transmissions. *Comput. Nurs.* 12(1): 13-16, 1994.
10. Shafarman, M. J., Meeks-Johnson, J., Jones, T., *et al.*, Implementing a record-oriented clinical lab interface using HL7 version 2.1 at Indiana University Hospital. *Proc. Annu. Symp. Comput. Appl. Med. Care* 511-515, 1991.
11. Health Level Seven, Health Level Seven Ver.2.2 Documentation, 1994.