Performance Test of XDBs for Efficient XML Medical Document Management

Hangchan Kim[a], Minkuck Joh[a], Ilyeon Kim[b], Hyungkyu Roh[c], Ilkon Kim[a]

[a] Department of School of Computer Science, Kyungpook National University
[b] Department of Information and Communication, Kyungpook National University
[c] Graduate School of Information Security, Kyungpook National University

Abstract

As XML has become widely used as a first class format for data description and exchange, standardization of XML-based information exchanges protocol for message transfer between applications and information exchanges between hospitals is on going in the medical industry. As we anticipate that the use of XML document in medical information systems will increasingly expand, it has become an important issue to select a proper XDB for efficient management of medical XML document. This paper's authors implemented Patient Referral System with XDB that would be widely used among hospitals. In addition, using the XML document in the system, we implement performance test for two commercial XDBs.

Keywords:

XDB; XML; Referral System; HIS; HL7, CDA

Introduction

Due to rapid development of computer and information communication technology, computerization for medical institutions has been activated and private hospitals have fairly made progress in information digitization [1]. In addition, there is a movement that hospitals would exchange medical information with HL7 (Health Level 7), standard protocol for electronic data exchange in healthcare environment, based on these hospital information system [2].

HIS (Hospital Information System) using HL7 version 2.x protocol can be sufficiently constructed on the legacy RDBMS. But if HIS uses HL7 v3.0 protocol and CDA (Clinical Data Architecture) document which are going on standardization, the amount of XML document for application to exchange will rapidly increase. Therefore there is limit on the legacy RDBMS to store, retrieve and manage such medical XML document.

This paper recommends XDB (XML DBMS) to manage efficiently medical XML document based on HL7 version 3.0 messages and CDA. To test performance of XDBs, we implemented Patient Referral System and generated XML-based Patient Discharge Summary and stored it in XDB. We conducted a performance test on commercial XDBs, Cache v5.0 and Oracle 9i Release 2. But we used arbitrary defined format of Patient Discharge Summary because HL7 Working Group doesn't yet fully define standard of CDA.

We think that the result of this paper's performance test made an imperative guideline on selecting an XDB for medical institutions storage in future.

Background

HL7

It is not uncommon today for the average hospital to have installed computer systems for admission, discharge, and transfer; clinical laboratories; radiology; billing and accounts receivable, to cite a few. Often these applications have been developed by different vendors or in-house groups, with each product having highly specific information formats. As hospitals have gradually expanded information management operations, a concomitant need to share critical data among the systems has emerged. But it was not simple to fulfill the need because the lack of data and process standards between both vendor systems and the many healthcare provider organizations present a significant barrier to application interfaces [3].

As the figure 1 shows, HL7 v2.4 message currently used in common has a string-based and data-centric message structure whose field is divided by delimiters.

As time goes by, however, healthcare service industry needs more rigorous message structure than one of HL7 v2.x. To satisfy these needs, HL7 standard organization defined MDF (Message Development Framework) for standardization of HL7 version 3.0 and brought in XML technology. Figure 2 is an example of HL7 version 3.0 message which is generated from MDF.

And moreover, if all hospitals use different medical document formats to record medical information, it would be very complicate problems to exchange and process medical document among hospitals. Therefore HL7 standard institute defines CDA with HL7 Reference Information Model (RIM) and the HL7 Version 3 Data Types to solve these problems. The HL7 Clinical Document Architecture (CDA) is a document markup standard that specifies the structure and semantics of clinical documents for the purpose of exchange [4].

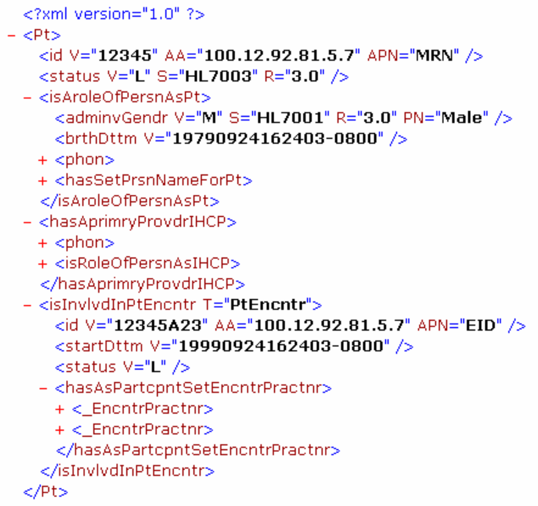
+

```
· MSH|^&~\|HOSPMPI|HOSP|CLINREG|WEST
  CLIN|199912121135-
  0600||RSP^K21^RSP_K21|1|D|2.4|

· QPD|Q21^Get Person
  Demographics^HL7nnn|111069|112234^
  ^^METRO HOSPITAL|^^^METRO
  HOSPITAL~^^^SOUTH LAB|

· PID|||112234^^^METRO
  HOSPITAL~98223^^^SOUTH
  LAB||Everyman^Adam||19600614|M||C|
  2101 Webster #
  106^^Oakland^CA^94612|
```

Figure 1- Example Version 2.4 message



```
  <?xml version="1.0" ?>
- <Pt>
    <id V="12345" AA="100.12.92.81.5.7" APN="MRN" />
    <status V="L" S="HL7003" R="3.0" />
  - <isAroleOfPersnAsPt>
      <adminvGendr V="M" S="HL7001" R="3.0" PN="Male" />
      <brthDttm V="19790924162403-0800" />
    + <phon>
    + <hasSetPrsnNameForPt>
    </isAroleOfPersnAsPt>
  - <hasAprimryProvdrIHCP>
    + <phon>
    + <isRoleOfPersnAsIHCP>
    </hasAprimryProvdrIHCP>
  - <isInvlvdInPtEncntr T="PtEncntr">
      <id V="12345A23" AA="100.12.92.81.5.7" APN="EID" />
      <startDttm V="19990924162403-0800" />
      <status V="L" />
    - <hasAsPartcpntSetEncntrPractnr>
      + <_EncntrPractnr>
      + <_EncntrPractnr>
      </hasAsPartcpntSetEncntrPractnr>
    </isInvlvdInPtEncntr>
  </Pt>
```

Figure 2 - Example of Version 3 message in a possible XML style

XML DBMS

We can define an XDB as a database management system that stores, updates, and retrieves XML documents. In addition to using the XML document as its base unit, an XDB provides a set of tools to manage the XML documents. These tools conform to a logical data model that abstracts the logical structures from the physical structures. There are no requirements for how an XDB is expected to physically store XML documents. Some XDBs are built on an object database, others might use compressed files with an indexing scheme, and still others might be built on top of a relational database. At this time XDBs can be classified into two basic types (with a third on the horizon): native and XML enabled [5].

A native XML database (NXDB) is simply one that was designed from ground up to store XML documents. It might make use of a preexisting technology such as object-oriented data storage technology such as object-oriented data storage techniques, but its mission is to store, retrieve, and update XML documents.

In the second type, an XML-enabled database (XEDB), extensions are added to a preexisting database management system to support XML documents. An XEDB can be built on top of an existing object-oriented or relational database management system. An XEDB provides a mapping layer between the XML documents and its database structures as well as support for XML-based tools to retrieve and update XML documents.

We put Cache version 5.0 and Oracle 9i Release 2 on a performance test. Cache has the properties of XEDB and Oracle 9i Release 2 is a convergence of the two other types.

System description

Referral System design

Figure 3 is the framework of Patient Referral System we implemented.

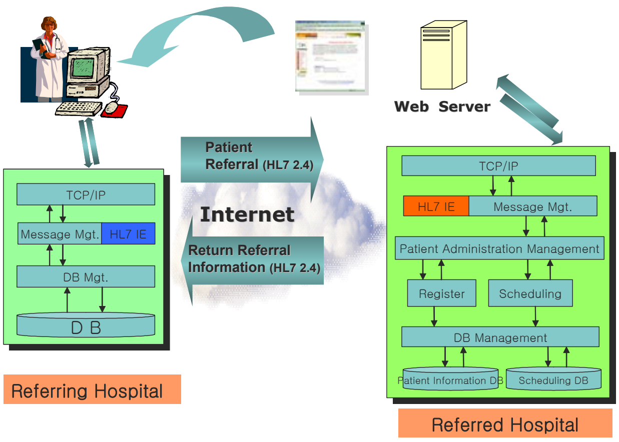


Figure 3 – Framework of Patient Referral System

Let's suppose that Primary Care Physician in referring hospital wants to send a referral to a specialist working for referred hospital. Using HL7 Interface Engine, Massage Management module generates HL7 version 2.4 message carrying patient information and sends it to a referred hospital through the Internet. Then Massage Management module in the referred hospital receives the message and understands it by the HL7 Interface Engine and then passes it to Patient Administration Management module. This module processing the message to reserve encounter with specialist and patient information is stored in database.

In general, the referred hospital sends the result report of the patient's visit back to the referring physician. If the referring hospital doesn't have HL7 Interface Engine installed, the referring physician can refer a patient and confirm the result report of the patient's visit through web.

When medical information is shared through Internet, security solution is necessary. This paper's referral system has medical information document go through encryption process to prohibit a malicious hacker from eavesdropping the medical information. When medical document is transported by HL7 Interface Engine and TCP/IP protocol, XML encryption and signature are used. On the other hand when medical document is transported by web server and HTTP protocol, SSL protocol is used to encrypt the medical information.

Conceptual database design

Figure 4 is an ER-Diagram (Entity Relation Diagram) that models structure of the legacy relation database of HIS (Hospital Information System). The legacy relation database contains basic information affiliated with Patient Discharge Summary. We suppose that Patient Discharge Summary is generated from the legacy relation database of HIS. All entities directly related to Patient Discharge Summary have relationships with 'adinfo' entity. 'adinfo' entity models table that contains the data generated whenever a patient is referred to the referred hospital. 'ICD9CM', 'ICD10' and 'LOINC' entities have information about disease and diagnosis. These entities of figure 4 have just code and name as attributes, omitting several other attributes.

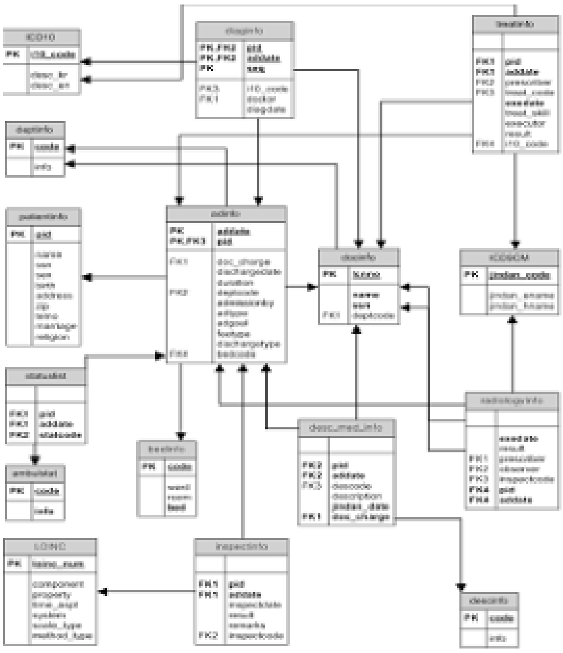


Figure 4 - ER Diagram of Legacy Hospital Database

Figure 5 is a class diagram which is designed with intention of storing XDB with Patient Discharge Summary. This class diagram is designed with intention of extracting related data from legacy relation database, storing XDB with them as XML format and extracting them from XDB as XML document.

PDS (Patient Discharge Summary) class stands for Patient Discharge Summary and has seven attributes. PBI (Patient Basic Information) class stands for basic information about a patient and PAI (Patient Admission Information) stands for admission information for a patient. Patient Diagnosis Information, Narrative Medical Information, Chief Treatment Information, Chief Observation Information, and Chief Radiology Observation Information class stand for information various about diagnosis and treatment for the patients and may form data as list structure. PDS (Patient Discharge Summary) class has five list data structures about information of diagnosis and treatment for a patient. PDS class's seven attributes are a class-embedded type of class or class-list structure.

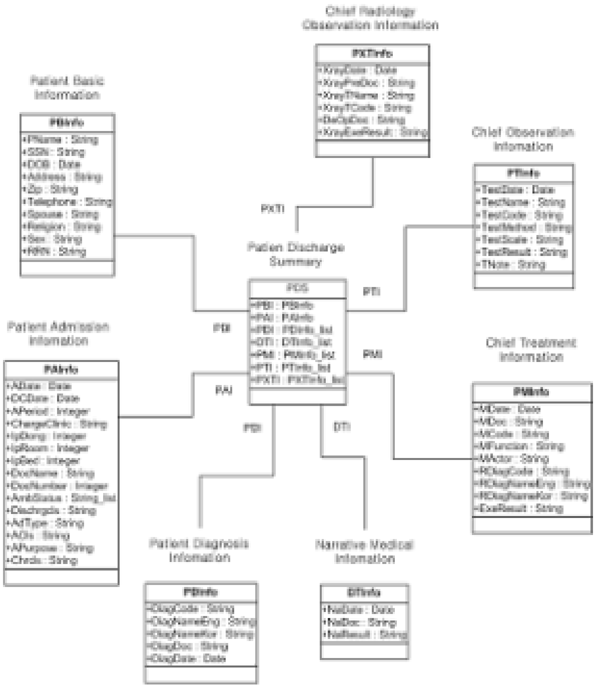
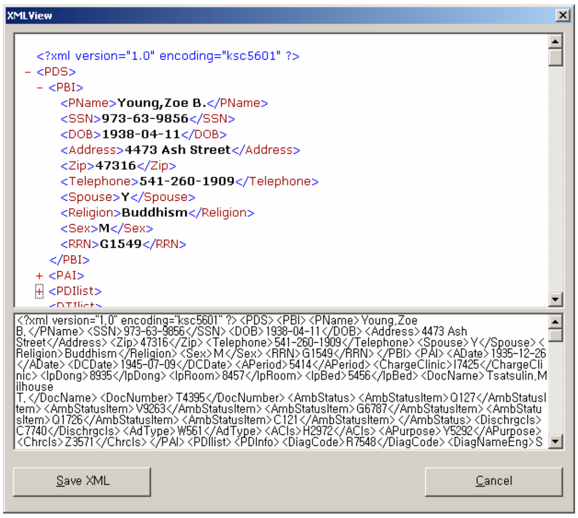


Figure 5 - Patient Discharge Summary Class Diagram

We made XMLViewer application that extracts data related to Patient Discharge Summary from the legacy relation database and maps them to PDS class to store XDB. And furthermore, XMLViewer application can extract a Patient Discharge Summary from XDB and show it with XML format like figure 6.



```
  <?xml version="1.0" encoding="ksc5601" ?>
- <PDS>
  - <PBI>
      <PName>Young,Zoe B.</PName>
      <SSN>973-63-9856</SSN>
      <DOB>1938-04-11</DOB>
      <Address>4473 Ash Street</Address>
      <Zip>47316</Zip>
      <Telephone>541-260-1909</Telephone>
      <Spouse>Y</Spouse>
      <Religion>Buddhism</Religion>
      <Sex>M</Sex>
      <RRN>G1549</RRN>
    </PBI>
  + <PAI>
  + <PDIlist>
```

```
<?xml version="1.0" encoding="ksc5601" ?> <PDS> <PBI> <PName> Young,Zoe
B. </PName> <SSN> 973-63-9856</SSN> <DOB> 1938-04-11 </DOB> <Address> 4473 Ash
Street</Address> <Zip> 47316</Zip> <Telephone> 541-260-1909</Telephone> <Spouse> Y</Spouse> <
Religion> Buddhism</Religion> <Sex> M</Sex> <RRN> G1549</RRN> </PBI> <PAI> <ADate> 1935-12-26
</ADate> <DCDate> 1945-07-09</DCDate> <APeriod> 5414</APeriod> <ChargeClinic> I7425</ChargeCli
nic> <IpDong> 8935</IpDong> <IpRoom> 8457</IpRoom> <IpBed> 5456</IpBed> <DocName> Tsatsulin,M
ilhouse
T. </DocName> <DocNumber> T4395</DocNumber> <AmbStatus> <AmbStatusItem> Q127</AmbStatusI
tem> <AmbStatusItem> V9263</AmbStatusItem> <AmbStatusItem> G6787</AmbStatusItem> <AmbStatu
sItem> Q1726</AmbStatusItem> <AmbStatusItem> C121</AmbStatusItem> </AmbStatus> <Dischrgcls>
C7740</Dischrgcls> <AdType> W561</AdType> <ACls> H2972</ACls> <APurpose> Y5292</APurpose>
<Chrcls> Z3571</Chrcls> </PAI> <PDIlist> <PDInfo> <DiagCode> R7548</DiagCode> <DiagNameEng> S
```

Figure 6 – XMLVeiwer showing XML-based Patient Discharge Summary

After XML-based Patient Discharge Summary is generated in this way, it goes through encryption and signature process. And then it is included in HL7 message and transferred to the referring hospital. If HIS of referring hospital can not process HL7 message, a referring physician can see a Patient Discharge Summary through web.

Patient Referral through Web

Figure 7 is a web page by which referring physician can select a department and a specialist. Physician can see this page through web-browser after entering basic patient information and description of medical examination result.

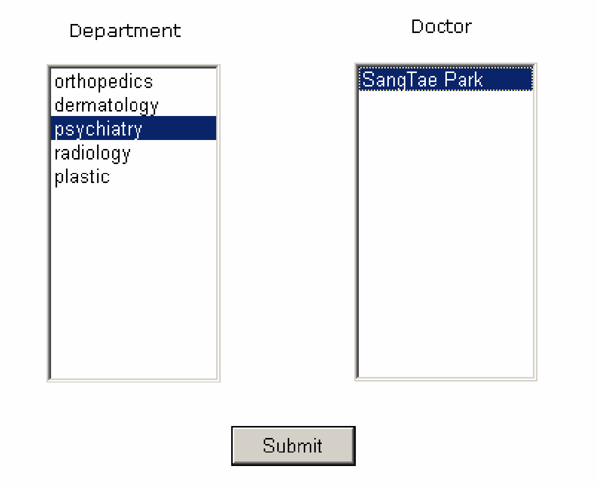
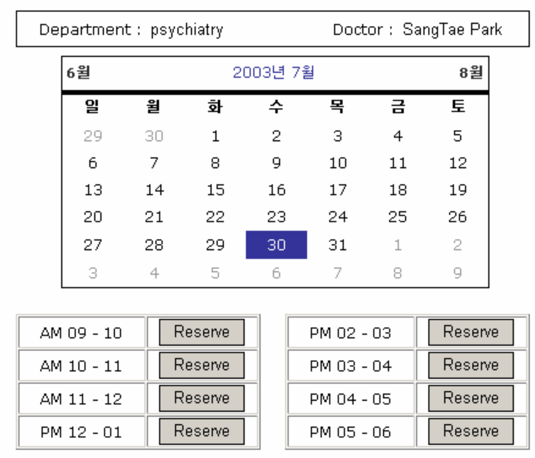


Figure 7 – Web page for selecting a department and a specialist

After that, web page changes to just like figure 8. As figure 8 shows, the specialist schedule shows up according to the department and the specialist selected at prior web page. Referring physician can make a reservation for encounter with the specialist by clicking unreserved time on patient visit day.



Department : psychiatry Doctor : SangTae Park

6월			2003년 7월			8월
일	월	화	수	목	금	토
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

AM 09 - 10	Reserve	PM 02 - 03	Reserve
AM 10 - 11	Reserve	PM 03 - 04	Reserve
AM 11 - 12	Reserve	PM 04 - 05	Reserve
PM 12 - 01	Reserve	PM 05 - 06	Reserve

Figure 8 – Web page for selecting date and time for encounter with a specialist

Performance test of XDBs

To test the performance against XDBs, XML documents, Patient Discharge Summary are used. Authors generated 10,000 XML documents and stored it in XDBs. Test environment follows below.

- OS : Windows 2000 Professional
- CPU : 700Mh, dual
- Memory : 1G

Cache v5.0 and Oracle 9i Release 2 are given the performance test. Actually these two XDBs are more widely used than anyone else. Let's observe the properties of these two DBMS. The Cache has the properties of enabled-XML DB and supports object-oriented programming. On the other hand, Oracle 9i Release 2 has the properties of Native-XML DB and enabled-XML DB and supports XPath.

The performance of the two XDBs is tested for below three search methods.

1. Index Search
2. Normal Search
3. Substring Search

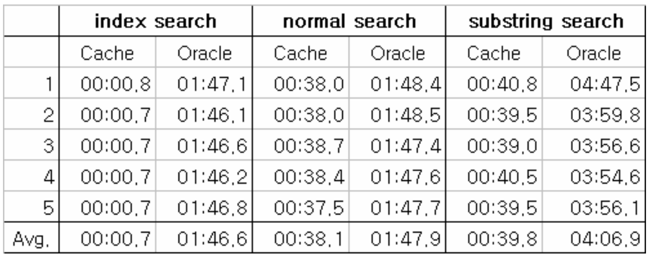
The method of Index Search makes a search for fields to which is given an index. On the other hand, the method of Normal Search makes a search for fields to which is not given an index. And the method of Substring Search makes a search for fields which has the target string as a substring.

The test programs using these three search methods are coded with Visual Basic and C# on the .NET platform. The program for Cache performance test is coded with Visual Basic on .NET platform because Cache provides COM object enabling a programmer to use Visual Basic. One the other hand, the program for Oracle performance test is coded with C# on .NET platform. The speed difference between Visual Basic and C# can be ignored because these programs were coded as console applications.

The program for performance test is executed with five different data selected from database. The test was put on these five data with three search type. And then table 1 shows the test result.

Figure 10 is a chart that is generated from table1 and represents two XDBs's average search time for specific search method.

Table 1 – Results of performance test against XDBs



	index search		normal search		substring search	
	Cache	Oracle	Cache	Oracle	Cache	Oracle
1	00:00.8	01:47.1	00:38.0	01:48.4	00:40.8	04:47.5
2	00:00.7	01:46.1	00:38.0	01:48.5	00:39.5	03:59.8
3	00:00.7	01:46.6	00:38.7	01:47.4	00:39.0	03:56.6
4	00:00.7	01:46.2	00:38.4	01:47.6	00:40.5	03:54.6
5	00:00.7	01:46.8	00:37.5	01:47.7	00:39.5	03:56.1
Avg.	00:00.7	01:46.6	00:38.1	01:47.9	00:39.8	04:06.9

In the case of Oracle 9i Release 2, XPath is used for a data search because Oracle 9i Release 2 is a native XML database (NXDB). In this case, although an index is given to a field, there is no improvement in search speed. On the other hand, because Cache is an XML-enabled database (XEDB), in the case of Index Search, there is considerable improvement in search speed. Cache supports an object-oriented programming method. By the way, if a programmer use this method instead of XPath, search speed of Cache is better than Oracle's.

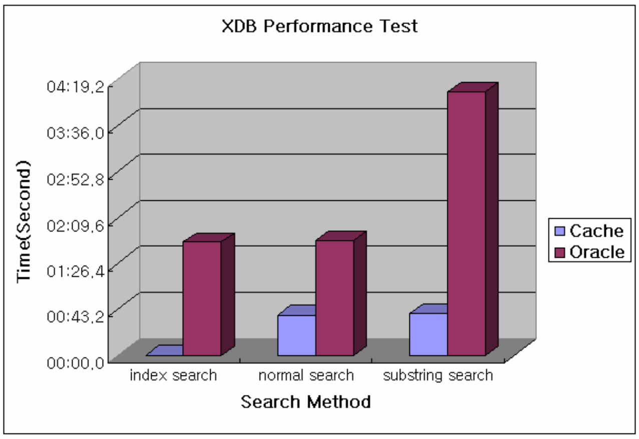


Figure 9 – chart of search time result of performance test against Cache and Oracle

Conclusions

Until now, this paper introduced the Patient Referral System implemented by the authors and reports the result of performance test against two XDBs, Cache version 5.0 and Oracle 9i Release 2. As the result says, Cache supporting object – oriented programming method searches XML document faster than Oracle using XPath does.

Future work will be devoted to test the other XDBs and implement Patient Referral System using HL7 version 3.0 Interface Engine, after HL7 version 3.0 standard protocol is confirmed.

Acknowledgments

This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, in Republic of Korea (02-PJ1-PG6-HI03-0004).

References

[1] Mandle KD, Kohane IS. Healthconnect: Clinical grade patient-physician communication. In Proceedings, AMIA Annual Symposium 1999

[2] Sooyung Yoo, Boyoung Kim, Jinwook Choi, Jaeheon Cheong, Jonghoon Chun, "Development of HL7 Message Server with Laboratory Common View Layer", Proceedings of The 29th KISS Spring Conference, 2002

[3] HL7 Working Group, HL7 STANDARD VERSION 2.4, 2000

[4] Gorge W. Beeler, Stan Huff, Wesley Rishel, et al, Message Development Framework, HL7 Working Goup,1999

[5] Greg Mable, "The Next Generation Database – XDB", http://www.sys-con.com/xml/articleprint.cfm?id=421

[6] Ben Chang, Mark Scardina, Stefan Kiritzov, "ORACLE 9i XML Handbook", Oracle Press, 2002

[7] Kevin Williams, Michael Brundage, Patrick Dengler, Jeff Garbriel, Andy Hoskinson, "PROFESSIONA XML Databases", Wrox Press Ltd, 2001

[8] Ronald Bourret, "XML and Databases", http://www.rpbourret.com/xml/XMLAndDatabases.htm

[9] H.X.Mel, Doris Baker, "Cryptography Dcrypted", Addison-Wesley, 2001

[10] HL7 Clinical Data Architecture Release 2.0 Cleveland draft, 2003

[11]Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database System", Addison-Wesley, 1997