

# The LEX System : HL7을 사용하는 전자의무기록의 효율적인 교환과 공유를 위한 XML기반 통합의료환경의 구축

이 민 경<sup>†</sup> · 정 재 현<sup>††</sup> · 전 중 훈<sup>†††</sup> · 유 수 영<sup>††††</sup>  
김 보 영<sup>†††††</sup> · 최 진 옥<sup>††††††</sup>

## 요 약

LEX(Lifelong EHR based on XML)시스템은 이질적인 병원정보시스템을 가진 병원들이 HL7 메시지를 생성하고, 이를 통해 의료정보를 효율적으로 공유하고 교환할 수 있는 중앙의 임상데이터베이스(Central Clinical DataBase/CCDB)를 구축하는 XML 기반의 의료정보통합 프레임워크이다. 본 논문에서는 기존의 병원정보시스템에 독립적인 HL7 메시지 서버 연동방식을 제안하여 HL7 메시지를 생성하고, 생성된 메시지를 효율적으로 저장하고 검색하기 위한 환자중심의 XML데이터베이스 스키마를 제안한다. 또한 HL7 메시지를 새로운 구조의 XML문서로 변환하기 위한 DTD를 제안함으로써 기존의 방식보다 저장공간을 절약하고, 효율적인 환자중심의 검색을 지원하는 방안을 제시한다.

## A Medical Integration Framework based on XML for efficient exchange and sharing of Electronic Health Record using HL7

Minkyung Lee<sup>†</sup> · Jaeheon Cheong<sup>††</sup> · Jonghoon Chun<sup>†††</sup> · Sooyoung Yoo<sup>††††</sup>  
Boyoung Kim<sup>†††††</sup> · Jinwook Choi<sup>††††††</sup>

## ABSTRACT

The LEX system is a XML-based framework for medical information consolidation. The Lex makes it possible for heterogeneous HISs (Hospital Information Systems) exchange and share HL7 messages by storing the messages into a single Central Clinical Database. In this paper, we propose a HL7 message server independently interoperable from existing HIS to generate HL7 messages, and design an XML database schema suitable for storing and manipulating such data. We also propose a new DTD for efficient transformation of HL7 messages to XML documents for storage saving as well as supporting patient-oriented information retrieval.

**키워드 :** 전자의무기록(Electronic health record), HL7, XML, XML Database, DTD

### 1. 서 론

최근 인터넷을 기반으로 하는 웹의 급속한 발전으로 이제 정보는 인터넷을 통해서 교환되고 공유될 수 있는 개방된 환경에서 더 큰 가치를 지니게 되었다. 이러한 경향은 의료분야에도 예외 없이 적용되어 지역 네트워크를 기반으로 운영되어 오던 기존의 의료정보 시스템도 병원 및 타

기관들과의 데이터공유 필요성으로 인해 개방형 시스템으로의 전환이 요구되고 있다. 따라서, 의료정보의 교환과 공유를 위해 각 의료관련 시스템들에 대한 표준들이 제정되었으며, 거기에는 HIS(Hospital Information System)의 정보를 교환하기 위한 텍스트기반의 HL7(Health Level 7)을 비롯해, PACS(Picture Archiving & Communication System)를 위한 DICOM(Digital Imaging and Communication in Medicine), EDIFACT, X.12 등이 있다[1].

그 중에서도 HL7은 텍스트 기반의 의료정보들을 교환하기 위한 표준으로 이벤트가 발생하면 그에 따른 HL7 메시지가 생성되어 네트워크를 통해 다른 시스템들에 보내지는 병원 프로세스 중심의 특성을 지녔다. HL7 메시지는 데이

† 준 회원 : 한국국방연구원 연구원  
†† 준 회원 : 신구대학 컴퓨터정보처리과 교수  
††† 정 회원 : 명지대학교 컴퓨터공학과 교수  
†††† 준 회원 : 서울대학교 대학원 의공학교실  
††††† 정 회원 : 서울대학교 의과대학 의공학교실 연구원  
†††††† 정 회원 : 서울대학교 의공학 교수  
논문접수 : 2002년 3월 27일, 심사완료 : 2002년 7월 29일

터 전송의 최소 전송단위로 순차적인 세그먼트(segment)들의 집합으로 구성되어 있고, 그 세그먼트들은 데이터필드(data field)들의 집합으로 구성되어 있는데, 각각의 세그먼트는 그 기능에 따라 MSH, EVN, PID 등의 이름이 주어진다. 또, 각각의 데이터필드들은 실제 데이터를 담고 있는 부분으로 컴포넌트나 서브 컴포넌트들로 세분화되어 있다. 뿐만 아니라 HL7은 모든 정보를 다루기 위해, 대부분의 세그먼트와 데이터 필드들을 옵션으로 정의하였고, 경우에 따라서 중복을 허용하기도 한다[2]. 이러한 특성으로 인해 HL7 메시지를 이용해서 의료정보를 통합하는 대부분의 시스템에서는 HL7 메시지를 XML문서로 변환하며, 이를 위해서 HL7-SGML/XML SIG에서 제안한 DTD를 사용한다[3, 4].

비정형적이며 구조적인 데이터를 교환하는데 XML문서가 적합하긴 하지만 또 다른 문제는 HIS의 데이터 종류(DB 스키마)에 따른 의료정보의 다양성이다. 현재 각 병원들은 이질적인 HIS를 가지고 있기 때문에 같은 이름의 메시지라 하더라도 병원 자체 DB 스키마에 따라 이질적인 형태의 HL7 메시지가 생성되기 때문에, 이들 간에 직접적인 메시지 교환이 어렵다. 뿐만 아니라 현재 각 병원마다 저장되어 있는 개인 정보와 의무 기록들이 재정적, 시간적 낭비 없이 효율적으로 이용되기 위해서는 모든 병원들이 이용할 수 있는 중앙의 임상데이터베이스가 필요하다[6-8].

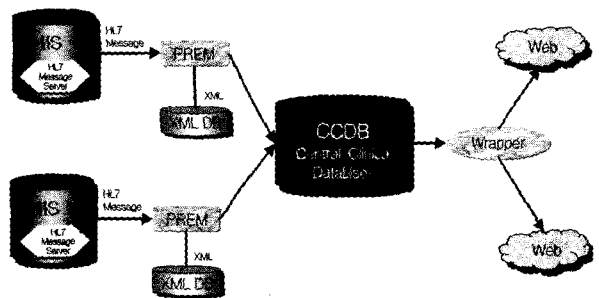
본 논문에서는 이질적인 HIS로부터 HL7 메시지를 생성하고, 이를 방대한 양의 중앙집중형 데이터베이스에서 효율적으로 의료정보를 검색할 수 있도록 XML문서로 변환하여 저장하는 시스템을 설계 구현하고자 한다. 이를 위해 첫째로, 기존의 병원정보시스템에 최소한의 변화를 주면서 클라이언트 어플리케이션에 독립적인 인터페이스 엔진의 개발이 필요하다. 둘째로, HL7 메시지가 HIS에 따라 여러 형태를 지니기 때문에 HIS에 독립적이면서, 환자중심의 검색1)을 할 수 있도록 변환하여야 한다. 기존에 제안된 DTD[3]는 하나의 HL7 메시지를 하나 XML문서로 1:1 변환한다. 따라서 환자가 내원해서 퇴원하는 프로세스 동안 이벤트 형식으로 많은 HL7 메시지가 발생되며, 이는 또한 많은 양의 XML문서로 변환되는 것을 의미한다. 이러한 XML문서를 그대로 저장할 경우, 환자정보의 대한 데이터가 중복되고, 대량의 XML문서들 중에서 원하는 정보를 검색하기가 어렵다. 따라서 이러한 문제를 해결하기 위해 HL7 메시지를 XML문서로 변환 할 때, 중복되는 데이터를 제거하고 환자중심의 효율적인 저장기 가능하도록 하기 위한 새로운 DTD가 필요하다. 새로운 DTD를 적용시킨 XML문서를 생성하기 위해서는 기존의 DTD와 제안한 DTD 간의 관계를 정의하는 변환규칙이 필요한데, 이는 사용하는 틀에 의존적이며, 다양할 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 LEX 시스템의 구조와 구현내용을 설명하고, 제 3장에서는 저장공간에 대한 실험 및 성능평가 결과를 제시한다. 마지막으로 제 4장에서는 결론을 맺는다.

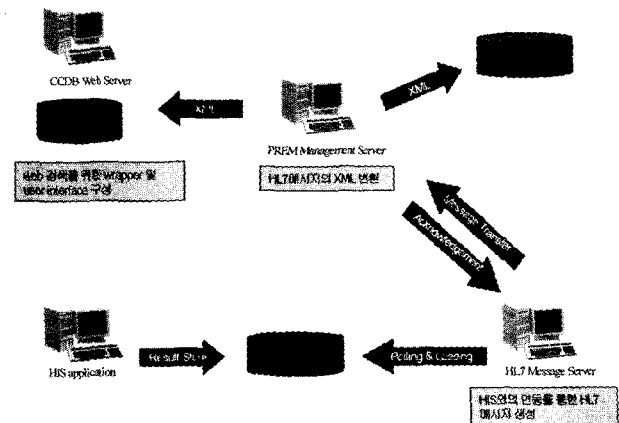
## 2. LEX 시스템

본 장에서는 LEX 시스템의 아키텍처와 각 컴포넌트들 간의 시나리오를 설명하고, 각 세부 컴포넌트의 구조와 구현내용을 설명한다.

### 2.1 아키텍처



(그림 1) LEX의 시스템 아키텍처



(그림 2) LEX의 컴포넌트 시나리오

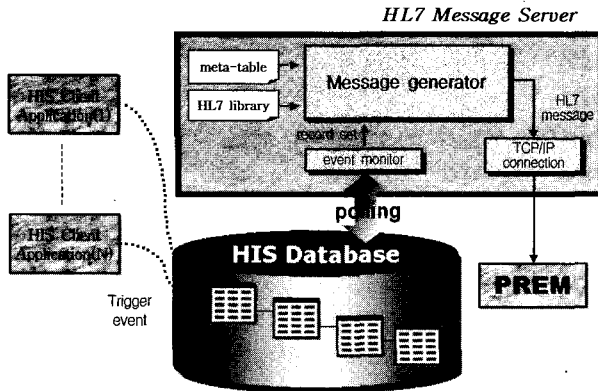
LEX는 크게 HL7 메시지 서버와 PREM 매니지먼트 서버로 구성되어 있다. HL7 메시지 서버는 환자가 내원하여 퇴원할 때까지 이벤트에 따라 HL7 메시지를 생성하고, 그것을 PREM 매니지먼트 서버에 전송하는 역할을 한다. PREM 매니지먼트 서버는 이를 XML문서로 변환하여 임시 데이터베이스인 XMLDB와 중앙임상데이터베이스인 CCDB에 저장한다. CCDB의 데이터는 Wrapper를 통해 사용자들에게 제공된다.

시스템 프로토타입의 구현은 (그림 2)의 컴포넌트 시나리오를 따르며, 이때 이벤트는 내원, 진찰, 수납, 검사, 예약, 약국의 순으로 발생한다고 가정하고, 그 중 발생할 수 있는

1) 환자 ID를 통해 환자별로 자신의 신상정보와 의료정보(등록, 진찰, 검사, 검사결과 등)를 검색하는 것을 의미한다.

메시지 타입을 PatientAdministration(ADT : 접수), OrderEntry(ORM : 검사 및 투약오더), ObservationReporting(ORU : 검사결과), Scheduling(SRM : 예약)의 4가지 타입으로 한정하였다.

### 2.2 HMS(HL7 Message Server)



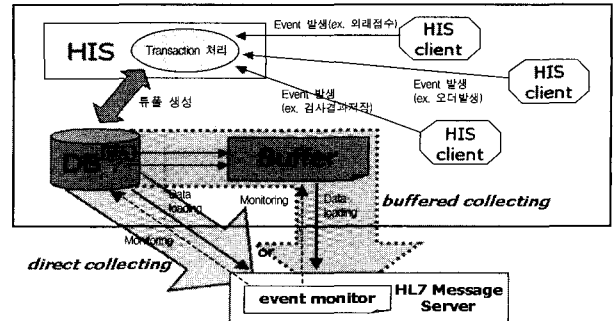
(그림 3) HL7 메시지 서버(HMS)의 구조

(그림 3)은 HL7 메시지 서버에 대한 구조로 크게 event monitor, message generator, TCP/IP connection 모듈로 구성되어 있다. Event monitor는 Timer를 이용하여 일정시간 간격(10초)으로 HIS 데이터베이스의 특정 테이블에 생기는 트랜잭션을 감시하여 HL7 트리거 이벤트의 발생을 감지하는 모듈이다. 일반적으로 HIS는 중앙집중형의 클라이언트/서버 기반의 시스템으로 응용 프로그램별로 사용자 인터페이스가 존재하며 HIS의 클라이언트가 표준화되어 있지 않다. 따라서 클라이언트에서 HL7 이벤트를 발생시켜 주고 이를 HL7 메시지 서버에서 감지하도록 하는 것은 시스템의 이식성 및 확장성과 유지보수에 많은 비용을 초래한다. 따라서 HIS의 변경 없이 클라이언트에 독립적이고 데이터베이스의 최소한의 변화만을 통해서 연동될 수 있도록 HL7 이벤트 식별 정보를 가지고 있는 특정 테이블을 폴링하여 데이터베이스의 트랜잭션을 감지하고 특정 HL7 이벤트가 감지된 시점에 HL7 메시지 서버가 동작하도록 하였다. 이때 폴링을 선택한 이유는 금융관련 서비스와는 달리 병원의 경우, 실시간 업무보다는 병원업무에 따라 필요한 시기에 배치 형태로 처리하는 경우가 빈번하기 때문이다. 긴급히 처리해야 하는 병원업무에 따라서는 폴링시기를 짧게 하거나 경우에 따라서는 실시간처리를 할 수 있겠지만, 본 연구에서는 시스템 성능을 고려하여 폴링방법을 사용하였다.

HL7 트리거 이벤트가 감지되면 해당 HL7 메시지 구성에 필요한 모든 데이터 항목들을 meta-table의 매핑 함수 검색키 정보와 함께 레코드 셋 형태로 HL7 메시지 서버 상의 메모리에 적재하고 message generator 모듈을 호출한다. Message generator는 메모리상의 레코드 셋을 바탕으로 생성할 메시지의 타입과 구조를 결정한 후, 메시지의 각

컴포넌트에 해당하는 데이터 항목들에 대해서 meta-table 검색키 정보를 이용하여 적절한 매핑 함수를 찾아 호출하는 것을 반복적으로 수행하는 방식으로 완전한 HL7 메시지를 생성하는 모듈이다.

이때 HL7 메시지 서버가 HIS 데이터베이스로부터 데이터를 가져오는 방법은 아래 (그림 4)와 같이 buffered collecting과 direct collecting의 두 가지 방법이 있다.



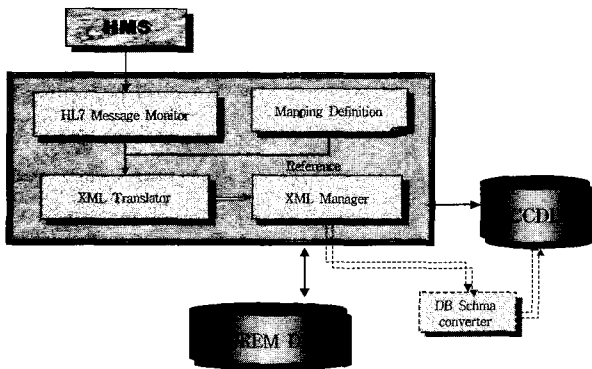
(그림 4) HL7 메시지 서버에서 HIS 데이터베이스의 데이터를 수집하는 방법

첫째, 중간 저장소를 통한 데이터의 수집(buffered collecting) 방법은 HL7 메시지 구성에 필요한 모든 HIS 데이터를 통합하여 저장/관리할 수 있는 중간 저장소(buffer)를 통해서 HL7 메시지 서버의 event monitor가 DB 트랜잭션을 감시하고 HL7 이벤트를 감지하도록 하는 방법이다. 이때 HIS 데이터를 중간 저장소에 복사/저장하기 위한 방법으로는 트리거를 이용한다. 이와 같은 방법은 원무시스템과 같이 원무관련 이벤트에 대해서 발생하는 데이터베이스 트랜잭션이 오직 하나의 테이블에 하나의 튜플로만 일어나는 경우에 유용한 방법이며, HIS 데이터베이스를 직접 접근함으로써 발생 가능한 위험 요소를 제거하고 현 HIS의 원활한 운영을 보장할 수 있다는 장점이 있다.

둘째, HIS DB 테이블의 직접 접근 및 조인 연산을 통한 데이터의 수집(direct collecting) 방법은 HL7 메시지 서버의 event monitor로 하역금 트랜잭션이 발생하는 여러 테이블 중에서 HL7 이벤트 식별 정보를 가지고 있는 특정 테이블을 감시함으로써 해당 HL7 이벤트를 식별하고 조인 연산을 통해서 메시지 구성에 필요한 모든 HIS 데이터를 수집하는 방법이다. 이 방법은 검사시스템과 같이 HIS 클라이언트 어플리케이션에서 발생한 이벤트에 대한 트랜잭션이 HIS 데이터베이스의 여러 테이블에 걸쳐서 하나 이상의 튜플을 생성하는 경우에 유용하다.

### 2.3 PMS(PREM Management Server)

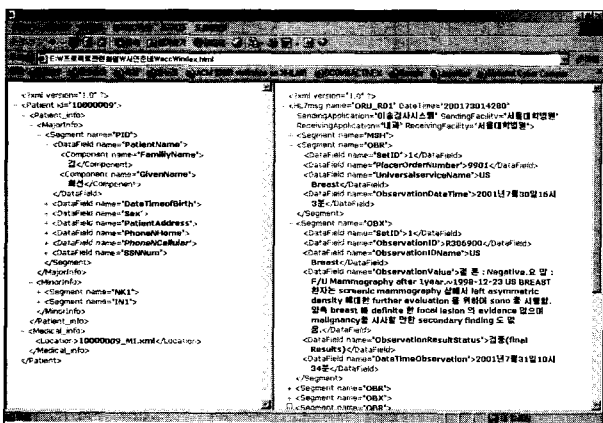
(그림 5)는 PMS의 구조로 크게 HL7 Message Monitor, XML Translator, XML Manager, DB Schema Converter 모듈로 구성되어 있다.



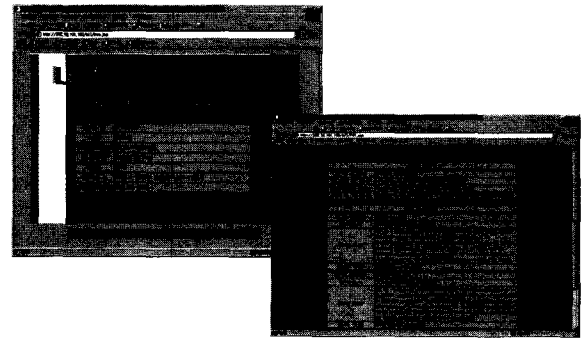
(그림 5) PREM 매니지먼트 서버(PMS)의 구조

HL7 Message Monitor는 HL7 메시지가 감지되면, 이를 파싱하여 메시지의 타입과 이벤트종류인 MSH의 HL7 event Identification 정보와 PID의 Patient ID를 추출하고, 파싱된 메시지와 함께 XML Translator에게 보낸다. XML Translator는 HL7 Event Identification에 근거하여 해당 변환규칙화일(.ecc)을 찾아 XML로 변환하는데 이용한다. 예를 들어 ADT^A04 메시지가 감지되었을 경우, HL7 Event Identification에 해당하는 MSH의 ADT^A04 값을 이용하여 여러 Mapping Definition file중에 ADT^A04와 관련된 화일을 찾아 HL7 메시지를 환자신상정보(Patient Information)인 PID.XML과 환자의료정보(Medical Information)인 MI.XML로 나누어 변환한다. 변환 후 이를 XML Manager에 전송하면 PREM DB와 CCDB로 저장한 후, 사용했던 HL7 메시지를 삭제한다. DB Schema converter는 PREM의 DB와 CCDB의 스키마가 다를 경우, PREM DB에 전송되는 XML 문서의 구조를 CCDB의 스키마에 적합하도록 변환하여 저장하는 모듈이다.

(그림 6)은 “김희선”이라는 환자가 내원하여 발생한 HL7 메시지가 PMS에 전송되었을 때, 환자신상정보와 환자의료정보로 분리한 후의 XML문서로 변환하는 화면이다. (그림 7)은 CCDB에 저장 후, 웹에서 환자 ID로 환자 신상정보와 환자정보를 검색한 화면이다.

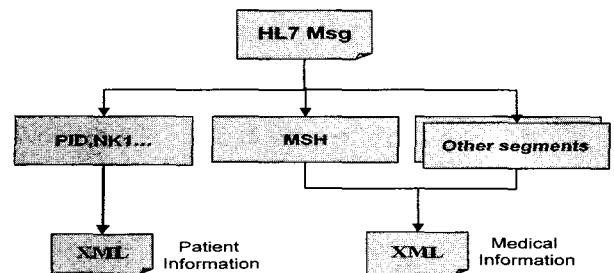


(그림 6) HL7 메시지를 XML문서로 변환한 화면



(그림 7) 환자신상정보와 의료정보의 검색 예

HMS로부터 전송된 HL7 메시지는 환자신상정보와 환자의료정보로 분리하여 HL7 메시지의 구조를 변경한 후 XML문서로 변환하는데, 이는 HL7 메시지가 트리거 이벤트를 일으킨 환자에 대한 신상정보를 가지고 있기 때문에, 한 환자가 퇴원할 때까지 발생하는 대량의 HL7 메시지에는 환자신상정보가 계속 중복되기 때문이다. 뿐만 아니라 검색 속도의 향상을 위해 HL7 메시지에서 중복되는 환자신상정보(patient information)와 지속적으로 생성되는 환자의료정보(medical information)로 분리하여 XML로 변환하여 저장한다.



(그림 8) HL7 메시지를 XML문서로 재구성

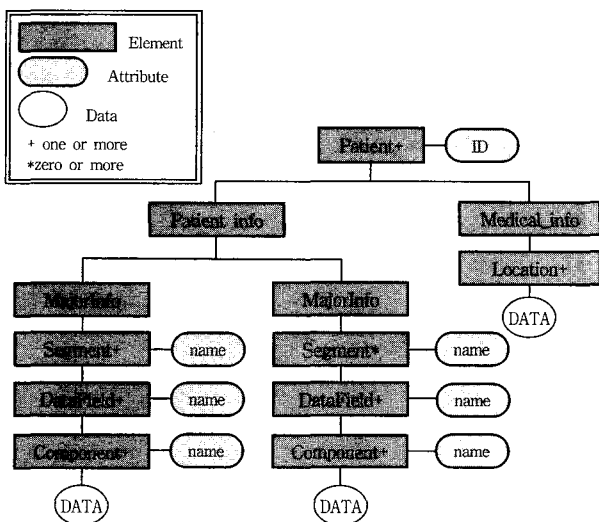
(그림 8)과 같이 환자신상정보(patient information)의 경우, 포함되는 세그먼트가 필수인지 선택인지에 따라 주정보(PID세그먼트)와 부정보(그 외에 선택적으로 중복되는 NK1, AL1세그먼트 등)로 분리한다. 이와 같이 필수인 주정보와 상대적으로 중요도가 떨어지는 정보를 나눔으로써, 트리구조를 가진 XML 데이터구조에서 환자검색속도를 증가시키고, 정보의 갱신을 보다 효과적으로 수행할 수 있게 한다. 환자의료정보(medical information)는 메시지 자체의 정보를 포함하고 있는 MSH세그먼트와 나머지 세그먼트들(주정보 및 부정보를 제외)로 구성된다.

다음은 환자신상정보와 환자의료정보로 분리하여 변환하기 위한 DTD를 설명하고, 변환한 XML문서를 환자중심의 검색하기 위한 XML 데이터베이스의 스키마를 설계한다.

### 2.3.1 HL7 메시지에서 환자신상정보

환자신상정보는 메시지들을 저장하기 위한 대상을 확인

할 뿐만 아니라, 검색을위해 사용되어지는 중요한 정보이다. 초기에 환자가 등록될 때는 환자신상정보가 XML문서로 변환되어 데이터베이스에 저장되지만, 한번 저장된 후에는 갱신이 없는 한, PID의 Patient ID를 기본키로 이용하여 환자신상정보를 검색하거나 변경한다. 만약 HL7 메시지를 변경 없이 XML문서로 변환할 경우, Patient ID를 찾기 위해서는 5 레벨 아래 있는 노드를 검색해야만 한다[3]. 따라서, 빠르고 효율적인 검색을 지원하기 위해 (그림 9)와 같이 환자마다 유일한 값을 가진 Patient ID를 환자신상정보의 가장 상위 엘리먼트에 ID속성을 지닌 애트리뷰트로 정의한다. 또한 지속적으로 증가하는 의료정보에 대한 확장을 고려하여, 환자의료정보 데이터베이스를 나누어 저장하고 관리할 수 있도록 설계한다. 즉, 해당 환자의 의료정보가 있는 데이터베이스의 위치와 이름을 MedicalInfo의 Location이라는 엘리먼트의 DATA로 저장하는 것이다. 예를 들어, 많은 의료정보가 발생하여 여러 데이터베이스에 나누어 저장했을 경우, 특정 환자에 대한 의료정보를 검색할 때, 환자 ID로 검색한 후 Location이 가지고 있는 데이터베이스의 위치와 이름을 가지고, 환자의료정보 데이터베이스를 검색하게 되는 것이다. 이때, 환자의료정보 데이터베이스의 이름을 환자아이디와 일련번호나 환자아이디와 년도 등의 규칙을 정해 놓으면 보다 효과적인 검색 및 유지보수가 가능하다.



(그림 9) 환자신상정보의 DTD구조

```

<!ELEMENT Patient (Patient_info, Medical_info)>
<!ELEMENT Patient_info MajorInfo, MinorInfo>
<!ELEMENT MajorInfo (Segment)>
<!ELEMENT MinorInfo (Segment*)>
<!ELEMENT Segment (DataField+)>
<!ELEMENT DataField (Component+)>
<!ELEMENT Component (#PCDATA)>
<!ELEMENT Medical_info (Location+)>
<!ELEMENT Location (#PCDATA)>
    
```

```

<!ATTLIST Component name CDATA #REQUIRED>
<!ATTLIST Segment name CDATA #REQUIRED>
<!ATTLIST Patient id ID #REQUIRED>
<!ATTLIST DataFieldname CDATA #REQUIRED>
    
```

(그림 10) 환자신상정보 DTD

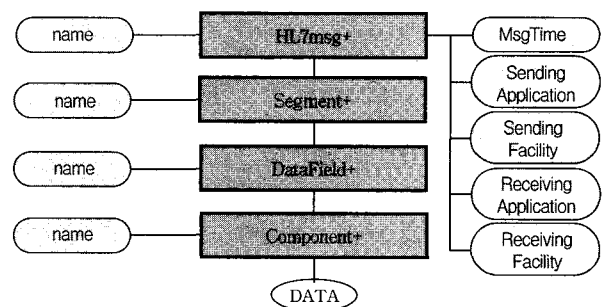
위의 (그림 10)은 데이터베이스에 저장될 환자신상정보의 DTD로, 실제 HL7 메시지는 XML로 변환시 환자신상정보와 환자의료정보로 분리하여 저장하지만, Location에 환자의료정보 데이터베이스의 위치와 이름을 저장하기 때문에, 논리적으로 보면 환자신상정보와 환자의료정보는 연결되어 있다.

실제 환자신상정보는 PatientInfo의 하위 엘리먼트로 저장하는데, HL7이 필수정보를 제외한 정보들을 옵션으로 정의해 놓았기 때문에 세그먼트, 데이터필드의 종류나 개수가 일정하지 않다. 따라서 세그먼트의 이름과 데이터필드의 이름을 엘리먼트로 직접 정하지 않고, Segment와 DataField라는 이름의 엘리먼트를 만들고, name이라는 애트리뷰트를 만들어, 실제 이름을 name 애트리뷰트의 값으로 저장하였다. 이로 인해 정의하지 않은 세그먼트, 데이터필드들이 생기거나 이름이 변경되어도 쉽게 추가 할 수 있도록 데이터베이스의 유연성을 높였다.

### 2.3.2 HL7 메시지에서 환자의료정보

환자의료정보는 여러 기관에서 새로운 정보가 계속적으로 발생되기 때문에 이를 효율적으로 검색할 수 있도록 (그림 11)과 같이 애트리뷰트를 추가하였다. 일반적으로 환자는 의료기록을 날짜별로 검색하기 때문에, MSH세그먼트 내에 위치하는 메시지 발생시간을 HL7msg의 MsgTime 애트리뷰트로 정의하였다. 또한 ID속성을 부여함으로써 메시지를 구분할 수 있는 기본 키 역할을 한다.

만약 환자가 같은날 여러 곳에서 진료를 받는다면 진료실에서 발생하는 검사오더에 관한 메시지(예 ORM001), 검사실에서 발생하는 검사결과에 관한 메시지(예 ORUR01)가 여러 개 발생하게 된다. 따라서 메시지를 보낸 곳(검사결과 보낼 곳)을 알려주는 SendingApplication, SendingFacility와



(그림 11) 환자의료정보의 DTD구조

받은 곳(검사요청 받은 곳)을 알려주는 ReceivingApplication, Receiving Facility를 새로운 애트리뷰트로 정의하였다.

```

<!ELEMENT HL7msg (Segment+)>
<!ELEMENT Segment (DataField+)>
<!ELEMENT DataField (Component+)>
<!ELEMENT Component (#PCDATA)>

<!ATTLIST HL7msg MsgTime ID #REQUIRED>
<!ATTLIST HL7msg SendingApplication CDATA #REQUIRED>
<!ATTLIST HL7msg SendingFacility CDATA #REQUIRED>
<!ATTLIST HL7msg ReceivingApplication CDATA #REQUIRED>
<!ATTLIST HL7msg ReceivingFacility CDATA #REQUIRED>
<!ATTLIST HL7msg name CDATA #REQUIRED>
<!ATTLIST Segment name CDATA #REQUIRED>
<!ATTLIST DataField name CDATA #REQUIRED>
<!ATTLIST Component name CDATA #REQUIRED>
    
```

(그림 12) 환자의료정보 DTD

환자신상정보와 마찬가지로, HL7 메시지와 세그먼트, 데이터필드들이 여러개 올수 있으므로, name이라는 애트리뷰트를 두어 데이터베이스의 확장을 대비해 유연성을 높였다. 위의 (그림 12)는 환자의료정보 DTD이다.

2.3.3 XML 데이터베이스 스키마 설계

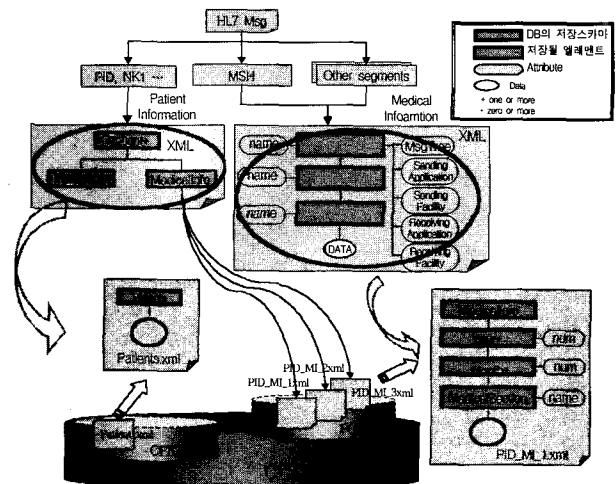
하나의 HL7 메시지를 두 개의 XML문서로 변환하는 과정은 HL7 메시지들을 효율적으로 저장하고 검색하기 위한 기초 작업으로 이를 기반으로 하여 XML 데이터베이스 스키마를 설계하고자 한다. 먼저 CCDB(Central Clinical DataBase)를 개념적으로 통합환자정보 데이터베이스(CPIDB : Central Patient Information DataBase)와 개인임상정보 데이터베이스(PCIDB : Personal Clinical Information DataBase)로 나누어 설계한다.

의료정보는 환자중심으로 검색하기 때문에 하나의 테이블에 한 환자에 대한 모든 정보를 가지고 있는 것이 검색에 효율적이다. 하지만 일반적으로 모든 환자들의 신상정보에서 특정 Patient ID를 찾아야 하기 때문에, 모든 환자의 신상정보를 하나의 테이블에 저장하는 것이 더 효율적이다. 반면 의료정보의 경우는 지속적으로 대량 생성되기 때문에 환자별로 하나의 테이블을 만들고 거기에 연이어 저장하는 것이 효과적이다. 즉, CPIDB는 물리적으로 모든 환자들의 신상정보를 하나의 XML파일(하나의 테이블)에 저장하고, PCIDB는 환자별로 의료정보를 저장하기 위한 XML파일(하나의 테이블)을 갖게 되므로 여러 개의 XML파일을 저장하는 형태를 갖는다. 물리적으로는 분리되어 있지만, 환자 ID로 검색한 환자의 신상정보에는 그 환자의 의료정보에 대한 위치를 알려주는 Location 엘리먼트가 있기 때문에 PCIDB에서 그 환자의 의료정보를 쉽게 찾을 수 있다. 따라서 환자의 신

상정보와 의료정보가 하나로 묶여 있으며, 또한 모든 환자가 하나의 데이터베이스로 묶여 있는 형태가 되는 것이다.

아래의 (그림 13)은 HL7 메시지 발생 후 XML 데이터베이스로 저장되는 과정을 보여주는 것으로, HL7 메시지 발생 후 환자 정보는 XML문서로 변환하여 CPIDB에 있는 Patients.xml의 Patients 루트(root)밑에 하위 엘리먼트로 저장하고, 의료정보는 변환 후 PCIDB에 있는 PID\_ML\_1.xml의 MedicalInfo루트 밑에 저장하는데, 이때 Year와 Month 엘리먼트를 두어, 년도 및 월별로 메시지를 분류하고, 그 밑에 MedicalSection 엘리먼트를 두어, 메시지타입별로 분류하여 저장한다. Year와 Month의 num 애트리뷰트에는 해당 년도와 월을 저장하며, MedicalSection의 name에는 메시지 타입 이름(PatientAdministration, OrderEntry 등)을 저장한다. 검색이 가능하다는 장점이 있다.

(그림 13)에서 Patients.xml은 CCDB에 저장된 통합환자정보 데이터베이스이며, 그에 대한 스키마는 (그림 14)의 DTD 밑에 환자신상정보 DTD를 추가한 형태이고, PDI\_ML\_1.xml은 개인임상정보 데이터베이스 중 하나로, 스키마는 (그림 15)의 DTD밑에 환자의료정보 DTD를 추가한 형태이다.



(그림 13) HL7 메시지가 변환 후 데이터베이스에 저장되는 형태

```

<!ELEMENT Patients (Patient+)>
    
```

(그림 14) CPIDB의 patients.xml의 DTD 추가 부분

```

<!ELEMENT MedicalInfo (Year+)>
<!ELEMENT Year (Month+)>
<!ELEMENT Month (MedicalSection+)>
<!ELEMENT MedicalSection (HL7msg+)>
<!ATTLIST Year num CDATA #REQUIRED>
<!ATTLIST Month num CDATA #REQUIRED>
<!ATTLIST MedicalSection name CDATA #REQUIRED>
    
```

(그림 15) PCIDB의 PID\_ML\_1.xml의 DTD 추가 부분

### 3. 실험 및 성능 평가

#### 3.1 실험 목적

본 실험은 HL7 메시지를 환자신상정보와 환자의료정보를 분리하여 중복되는 부분을 제거하였을 때, 저장공간의 효율성을 알아보기 위한 것으로, 발생된 HL7 메시지에서 중복을 허용하여 저장하였을 때와 중복을 제거하고 저장하였을 때를 비교하였다. 먼저 가정을 통해 데이터분석을 하여 그 효율을 예측하였고, 시뮬레이션 결과와 비교하였다.

#### 3.2 실험 방법 및 모델

실험방법은 한 환자가 내원하여 퇴원할 때까지 타입별로 여러 메시지가 발생하면, 첫 번째는 그 메시지들을 그대로 저장하고, 두 번째는 본 논문에서 제안한 대로 중복되는 세그먼트들을 제거하고 저장한 후 저장하였다. 시뮬레이션 10일 동안 반복 실험하였고 정확한 실험을 위해 smpl 시뮬레이션 라이브러리를 사용하였다[5].

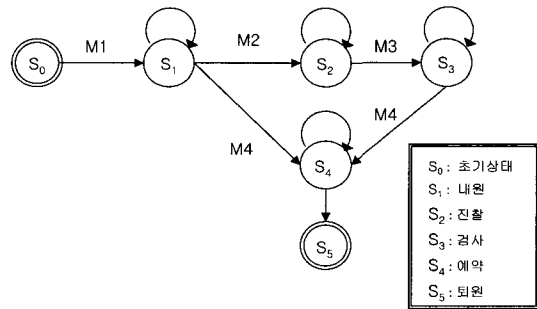
실험에서는 한 환자가 내원하여 퇴원할 때까지 타입별로 메시지들의 수와 종류를 랜덤하게 생성하였다. 메시지 타입은 연구범위와 동일하며, 각 타입에서 Query/Response에 해당하는 메시지들은 제외하였다.

환자가 내원하여 발생하는 메시지들은 (그림 16)의 실험 모델과 같이 2가지 경우로 시뮬레이션 하였다. 첫째는 S0의 초기 상태에서 내원(S1) 후 진찰(S2)과 검사(S3)를 받은 후 예약(S4)을 하고 퇴원(S5)하는 경우이고, 둘째는 내원(S1)하여 예약(S4)한 후 퇴원(S5)하는 경우이다. 본 실험에서는 실제 병원의 흐름을 고려하여 첫 번째 경우와 두 번째의 경우를 70 : 30 정도의 비율로 발생시켰고 각 상태로 이동하는데 걸리는 지연시간은 없다고 가정하였다. 이때 환자가 내원하는 시간은 exponential distribution을 사용하였고 S1, S2, S3, S4의 각 상태에서 발생할 수 있는 최대 메시지는 M1(PA와 관련된 메시지 그룹)에서 49개, M2(OE와 관련된 메시지 그룹)에서 12개, M3(OR과 관련된 메시지 그룹)에서 15개, M4(SC와 관련된 메시지 그룹)에 25개로 총 101개이다. 각 타입에서 발생하는 메시지들은 병원의 환경과 유사하도록 만들기 위해 PA의 A04, OE의 O01, OR의 R01 메시지는 환자마다 필수적으로 발생시켰고, 나머지 메시지는 필요에 따라 가중치를 두었다. 이때 각 메시지들의 중복 발생은 허용하지 않았다.

메시지의 생성을 위해 사용한 세그먼트의 종류와 데이터 필드의 크기는 HL7 v2.3을 기반으로 정의하였다. 각 메시지는 필수세그먼트와 옵션세그먼트로 나누어진다. 필수 세그먼트는 발생할 수 있는 데이터필드들의 최대 데이터양을 계산하여 최대(데이터양/2)~최대 데이터양 사이의 uniform Random Distribution 값으로 세그먼트의 크기를 결정하였다. 이때 필수 세그먼트의 길이를 50% 이상으로 산정한 이

유는 필수 세그먼트의 필수 데이터 필드들이 적어도 50% 이상의 정보를 가지고 있을 것이라 가정했기 때문이다. 이에 반해 옵션세그먼트는 0~최대(데이터양/2) 사이의 uniform Random Distribution을 적용하였다.

필수세그먼트와 옵션세그먼트에서 세그먼트나 필드의 중복은 고려하지 않았다. 또한 uniform Random Distribution을 사용한 이유는 세그먼트와 데이터필드의 수가 가변적이고, 데이터필드의 데이터양도 가변적이기 때문에 같은 타입이라 하더라도 환자에 따라 다른 형태, 다른 개수의 메시지들이 발생할 수 있으며, 거기에 어떤 규칙을 적용하기 어렵기 때문이다.



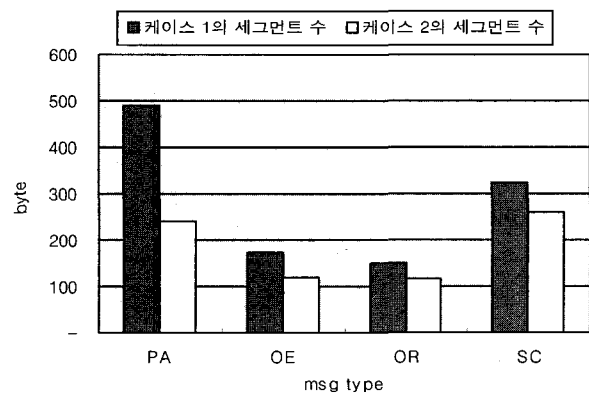
(그림 16) 실험 모델

#### 3.3 실험 결과

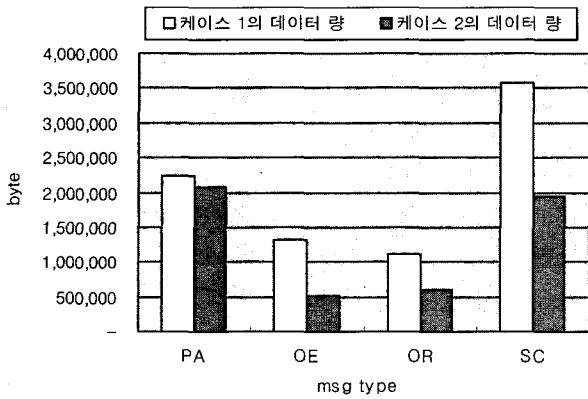
<표 1>은 101개의 메시지 수에서 모든 데이터가 최대로 발생되었다고 가정하였을 경우, HL7 메시지를 그대로 저장

<표 1> 케이스1과 케이스2의 메시지 타입별 세그먼트수와 데이터양 분석

	PA	OE	OR	SC	합 계
메시지 개수	49	12	15	25	101
케이스1의 세그먼트수	487	161	143	322	1,113
케이스2의 세그먼트수	237	114	112	261	724
케이스1의 데이터양	2,231,150	1,300,102	1,055,097	3,582,331	8,168,680
케이스2의 데이터양	2,082,244	502,244	590,268	1,955,299	5,130,055



(그림 17) 세그먼트수 비교



(그림 18) 데이터양 비교

한 케이스 1과 변환을거쳐 중복을 제거한 케이스 2에 대해 비교분석한 결과이다. 또한 (그림 17)은 케이스 1과 케이스 2에 대해 세그먼트수를 비교하였고, (그림 18)은 데이터양을 비교하였다. 분석결과 세그먼트 수는 약 35% 감소하였고, 저장공간은 약 38%정도를 절약할 수 있었다.

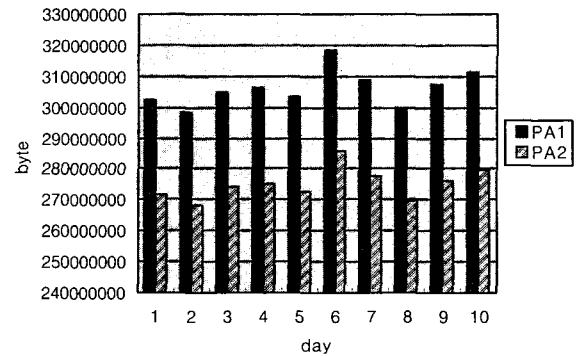
실제 시뮬레이션에서는 10일동안 총 13169명이 내원하였고 생성된 약 68만개의 메시지들을 그대로 저장할 경우 약 10기가바이트를 차지하였고, 환자신상정보에 해당하는 세그먼트들을 제거할 경우 약 6기가바이트를 차지하였다. 시험에서 발생한 데이터들은 각 환자에 따라 메시지들을 타입별로 분리하여 메시지의 수와 필수 및 선택 데이터들의 양을 저장하였고, <표 2>와 <표 3>은 하루 동안 타입별로 저장된 총 세그먼트 수와 데이터양을 합산한 결과이다. (그림 19)부터 (그림 22)까지는 각 타입별로 케이스 1과 케이스 2에서 발생한 데이터양을 비교한 것이다. 시뮬레이션 실험결과 환자신상정보에 해당하는 세그먼트들을 제거하면 제거하지 않는 것에 비해 약 37%의 저장공간을 절약할 수 있는 것으로 나타났으며 이는 가정을 통해 예측한 결과와 유사한 효율을 보임을 알 수 있었다.

<표 2> 10일간 발생한 타입별 데이터양 (케이스 1의 경우)

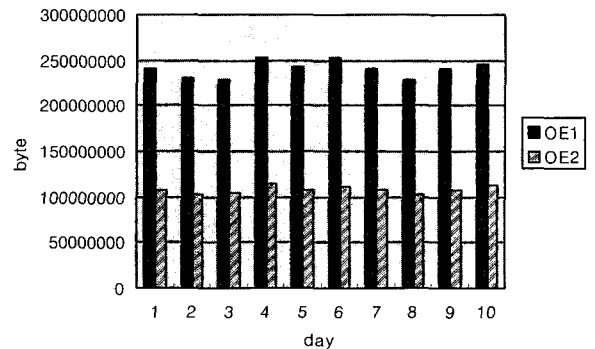
day	PA1	OE1	OR1	SC1	일일 합계
1	302,401,092	240,453,984	117,361,611	346,617,158	1,006,833,845
2	298,436,457	231,434,962	110,144,247	337,801,341	977,817,007
3	304,700,887	229,115,358	111,084,070	352,907,577	997,807,892
4	306,342,731	252,575,000	114,360,158	358,819,154	1,032,097,043
5	303,470,478	243,342,428	112,455,321	341,140,641	1,000,408,868
6	318,321,953	252,728,444	120,491,751	373,847,402	1,065,389,550
7	308,993,479	241,248,114	121,243,246	358,171,811	1,029,656,650
8	299,816,493	229,939,754	112,755,134	361,994,531	1,004,505,912
9	307,435,285	240,790,740	115,141,538	344,707,738	1,008,075,301
10	311,436,708	245,225,496	109,481,772	364,835,990	1,030,979,966
타입별 합계	3,061,355,563	2,406,854,280	1,144,518,848	3,540,843,343	10,153,572,034

<표 3> 10일간 발생한 타입별 데이터양 (케이스 2의 경우)

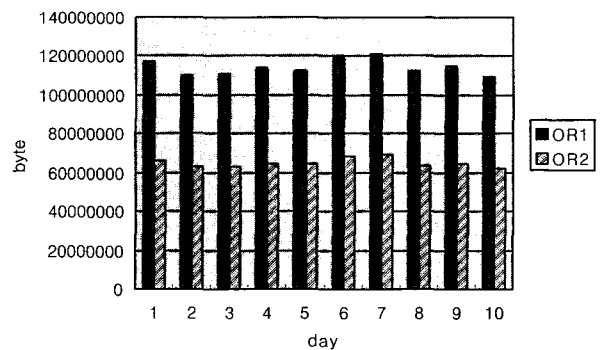
day	PA2	OE2	OR2	SC2	일일 합계
1	271,632,659	107,991,836	65,945,510	189,430,479	635,000,484
2	267,906,680	103,005,752	62,718,037	186,588,213	620,218,682
3	274,019,418	103,670,174	63,088,136	192,769,902	633,547,630
4	275,165,377	114,240,418	64,834,082	197,358,914	651,598,791
5	272,763,129	107,833,042	64,163,558	187,343,237	632,102,966
6	285,943,075	111,759,438	68,675,414	206,355,424	672,733,351
7	277,823,543	108,220,222	69,207,479	197,017,638	652,268,882
8	269,487,352	103,111,294	63,858,095	198,534,215	634,990,956
9	276,317,383	108,147,732	64,718,024	190,122,976	639,306,115
10	279,932,749	112,461,116	62,332,302	198,444,902	653,171,069
타입별 합계	2,750,991,365	1,080,441,024	649,540,637	1,943,985,900	6,424,938,926



(그림 19) PA의 케이스별 비교

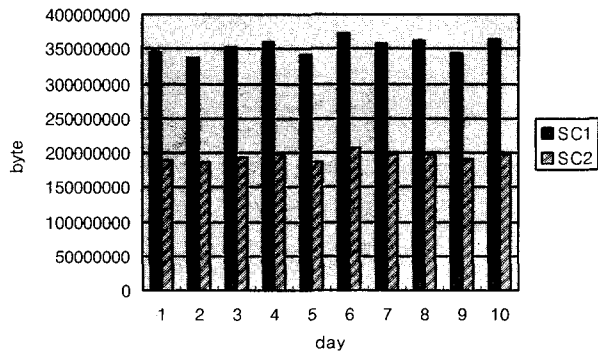


(그림 20) OE의 케이스별 비교



(그림 21) OR의 케이스별 비교





(그림 22) SC의 케이스별 비교

## 6. 결 론

본 논문에서는 병원정보시스템에 HL7 메시지 서버를 연결하여 현존하는 HIS 시스템에 최소한의 영향을 주면서 HL7 메시지를 생성하기 위한 방법을 제안하였다. 또한 이질적인 형태의 HL7 메시지들을 효율적으로 저장하고 검색할 수 있는 환자중심의 XML 데이터베이스 스키마를 제안하였다. HL7 메시지들을 새로운 DTD에 기반을 둔 XML 형태로 변환함으로써 효율적인 저장과 검색이 가능하게 되었다. 또한 웹을 통해 환자 개인의 임상 기록에 접근할 수 있는 의료 환경을 구축하여 환자중심의 검색이 가능함을 보였다.

실험 결과, HL7 메시지의 중복을 제거하여 저장하는 경우 약 37%의 공간을 절약할 수 있음을 알았고, 이는 빈번하게 이벤트가 발생하는 실제 병원환경에서 디스크 접근을 줄일 수 있다는 측면에서 시스템의 성능을 보장하는 중요한 열쇠가 되리라 사료된다. 본 논문에서는 HL7 메시지에 국한하여 시스템을 설계하고 프로토타입을 구현하였으나 앞으로는 여러 의료 관련 시스템들에서 발생하는 모든 의무기록을 통합하여 공유할 수 있는 연구가 필요할 것이라 예상된다.

## 참 고 문 헌

- [1] 안철범, 나연복 "XML을 이용한 통합 의료정보 시스템 구현", 한국정보과학회 가을학술발표논문집(1), 제27권 제2호, pp. 24-26, 2000.
- [2] HL7 organization, HL7 standard version 2.3, health level seven, Inc, 1997.
- [3] HL7 SGML/XML SIG, "XML as an Interchange Format for HL7 V2.3 Messages," <http://www.mcis.duke.edu/standards/HL7/committees/sgml/WhitePapers/hl7v2/hl7xml2.zip>.
- [4] W3C, Extensible Markup Language(XML)1.0, <http://www.w3.org/TR/2000/REC-xml-20001006.html>, Oct., 2000.
- [5] Paul A. Fishwick, "Simulation Model Design and Execution," pp.29-60, pp.129-159.
- [6] Sooyoung Yoo, Boyoung Kim, Seungbin Han, Youngchul Lee, Jinwook Choi, M. D., Ph. D, Jaehoon Cheong, Minkyung Lee, Jonghoon Chun, Ph. D, "Automatic RIM (Reference

Information Model) Wrapper for LEX : Lifelong Electronic Health Record Based on XML," AMIA, 2001.

- [7] Sooyoung Yoo, Boyoung Kim, Seungbin Han, Jinwook Choi, Minkyung Lee, Jonghoon Chun, Jaehoon Cheong, Sang-goo Lee, Won Gyu Lee, "A Framework to Integrate Electronic Health Record based on HL7 and XML : The LEX System," CJKMI, 2001.
- [8] 유수영, 김보영, 한승빈, 최진욱, 정재현, 이민경, 전종훈, "의료기관 정보 공유를 위한 HL7 메시지 서버 프로토타입 설계 및 구현", 대한의료정보학회 대한의료정보학회지, 제7권 제3호, pp.1-8, 2001.



### 이 민 경

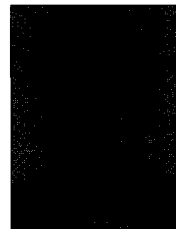
e-mail : mkl@mju.ac.kr

2000년 명지대학교 컴퓨터공학과 졸업 (학사)

2002년 명지대학교 대학원 컴퓨터공학과 졸업(공학석사)

2002년~현재 한국국방연구원 연구원

관심분야 : XML, 데이터베이스, 객체지향방법론



### 정 재 현

e-mail : jhjung@shingu.ac.kr

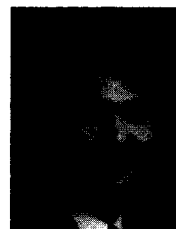
1993년 서울대학교 자연과학대학 계산 통계학과 졸업(이학 박사)

1995년 서울대학교 자연과학대학 대학원 전산학과 졸업(이학 석사)

1997년 서울대학교 공과대학 전기컴퓨터 공학부 박사과정 수료

2002년~현재 신구대학 컴퓨터정보처리과 재직

관심분야 : 데이터베이스, 정보검색, 웹데이터



### 전 종 훈

e-mail : jchun@mju.ac.kr

1986년 덴버대학교 전산과학(학사)

1988년 노스웨스턴대학교 전산과학(석사)

1992년 노스웨스턴대학교 전산과학(박사)

1995년 센트럴 오클라호마대학교 조교수

1995년~현재 명지대학교 컴퓨터공학과 부교수

관심분야 : 데이터베이스, 정보검색, 의료정보



### 유 수 영

e-mail : yoosoo0@snu.ac.kr

2001년 명지대학교 컴퓨터공학과 졸업 (학사)

2002년~현재 서울대학교 의과대학 대학원 의공학교실 석사과정

관심분야 : 데이터베이스, 의료정보, XML, 표준화



**김 보 영**

e-mail : bykintw@chol.com

1998년 전북대학교 컴퓨터학과 졸업  
(학사)

2000년~2001년 전능메디컬소프트웨어(주)

2001년~현재 서울대학교 의과대학 의공  
학교실 연구원

관심분야 : 데이터베이스, 소프트웨어 공학, 의료정보, XML



**최 진 옥**

e-mail : jinchoi@snu.ac.kr

1987년 서울대학교의과대학 졸업(학사)

1993년 서울대학교 대학원 의학박사  
(의공학)

1996년~현재 서울대학교 의과대학  
조교수(의공학)

관심분야 : HL7, 의료정보의 Semantic IR