

# XML Schema를 이용한 병원정보교환시스템 HIES의 확장

홍동완,<sup>0</sup> 노관준, 박진만, 김종덕, 윤지희  
한림대학교 컴퓨터공학과

(dwhong<sup>0</sup>, pillar, pjm, clrbell, jhyoon)@hallym.ac.kr

## Extension of a Hospital Information Exchange System using XML Schema

Dongwan Hong,<sup>0</sup> KwanJun Noh, JinMan Park, Jongduk Kim, JeeHee Yoon  
Dept. of Computer Engineering, Hallym University

### 요 약

최근 XML 문서는 여러 분야의 시스템에서 교환 표준으로 자리 잡고 있다. 의료 사업 분야에서도 병원 간 자료를 전송을 하는데 XML을 사용하는 것이 유용하며, HL7에서도 병원 간 외부 교환 표준으로 XML 문서를 선정하였다. 교환 매개인 XML 문서는 DTD를 이용하여 유효성을 검증할 수 있는 있으나, DTD는 구조 표현과 데이터 형식지원의 한계로 인하여 복잡 구조의 의료 데이터를 표현하기에 부족한 점이 있다. 병원정보교환시스템 HIES는 기존 병원 내의 시스템 구조를 유지한 채 병원 내 진료과 간이나 외부 병원 사이에 의료 데이터를 전산 공유할 수 있는 시스템으로, 대부분 복잡구조의 의료 데이터를 전송하게 된다. 본 논문에서는 XML DTD 기반으로 구축되었던 HIES를 XML Schema를 이용하여 확장함으로써, 전송 문서의 내부 구조 변경, 정보 공유 관리자 및 동적 스키마 처리, 정보 보안 분야에서 얻는 이점을 보인다.

### 1. 서 론

최근 XML(eXtensible Markup Language)은 다양한 사업 분야에서 공유 표준으로 자리 잡아가고 있다. XML은 문서의 구조를 나타내는 DTD(Document Data Type)를 기반으로 문서의 유효성(validation)을 검증받을 수 있다. 이와 같은 특성으로 인하여 DTD 기반의 XML 문서를 데이터 교환 표준으로 적용한 시스템이 급증하였다. 의료 사업 분야에서도 표준화 작업을 주요 업무 안으로 진행해 나가고 있는 상황이다. 현재 의료 데이터의 전자적 교환 표준으로 HL7(Health Level 7)[1]이 제안되었다. HL7은 데이터 표준, 메시지 표준, 문서 표준으로 이루어져 있으며 버전 3.0에서는 HL7 데이터를 외부 데이터 전송 및 표현(presentation)에 사용하도록 XML문서를 채택하여 DTD 구축 작업을 진행, 완료 단계에 이르렀다. 즉, 내외적으로 HL7을 기반으로 시스템을 구축하도록 권고하고 있는데 병원 내 진료과 간에는 HL7 메시지를 이용하여 의료 정보를 교환하며, 외부 병원과는 HL7 메시지를 XML 문서로 변환하여 의료 데이터를 교환할 수 있다. 하지만, 병원 내 시스템에 HL7 게이트웨이를 추가로 개발해야 한다.

HIES(Hospital Information Exchange System)[2,3]는 본 연구실에서 개발한 시스템으로 병원 내 기존 시스템의 변경 없이 외부 병원, 병원 내 각 진료과 및 HL7 데이터와의 정보 연계가 가능하다. 정보 공유, 전송을 위해 HIES는 내부 및 외부 메시지 교환으로 XML 문서를 사용하며, 웹 브라우저를 이용하여 시스템에 접근 가능하므로 웹을 사용할 수 있는 모든 병원에서 쉽게 사용, 가능하다.

XML DTD는 문서의 형태를 정의할 수 있으나, 지원되는 데이터 형식이 한정적이며, 임의로 사용자가 정의한 데이터 형식을 지원하기 어렵다. HIES는 의료 시스템에서 발생하는 모든 데이터를 처리 대상으로 하여, 환자 기초 정보, 의사의 진단 정보, 의료 장비(Modality)에서 산출된 데이터 등을 예로 들 수 있다. 이들 각각의 데이터는 유기적으로 결합되어 표현될 경우 의료 행위에 유용한 정보가 되는데, 이는 일반적으로 복잡구조의 데이터 형식이다. 또한 HL7 메시지는 HMD(Hierarchical Message

Description)[4,5]에서 처리하는데, HMD의 기능을 지원하는 METL(Message Element Type Language)[4,5]은 계층 구조의 메시지(Message Structure)와 복합(composite), 열거(list), 선택(choice), 기초(primitive), 메타 타입인 메시지(message)의 메시지 형식(Message Type)을 지원한다[5]. 이러한 데이터 구조를 XML DTD를 이용하여 표현할 경우 무리가 따른다. 하지만, XML DTD에 비해 XML Schema는 다양한 데이터 형, 관계(Relationship), 복합(Composite) 속성, 상속(inheritance) 등을 지원하므로 유연하고 확장 가능하며 복잡 구조의 의료 데이터를 표현하기에 적합하다[5,6]. 본 논문에서는 병원 정보 교환 시스템(HIES)에서 교환 매체로 사용되던 DTD 기반의 XML문서를 XML Schema[7]를 이용하여 확장하는데 필요한 내용을 기술한다.

### 2. 관련연구

#### 2.1 HIES

일부의 환자가 여러 진료 과에서 진료를 받아야 할 경우, 전산 공유가 가능하지 않은 병원에서는 진단 결과에 대해 의료인이 직접 검색이나 전송에 참여한다. 또한 의료 시설이 충분히 구비된 병원에 수탁진료(reference)를 요청할 경우, 환자의 과거 진료 내역을 첨부하여야 하는데 데이터 전송이 불가능할 경우 환자가 재검사를 받아야 하는 불편이 있다. HIES는 기존의 시스템 구조를 유지한 채 병원 내 각 진료과 및 병원 간 의료 데이터를 공유하기 위하여 개발된 시스템으로 데이터의 전송 매개로 XML 문서를 이용하고 있다. HIES를 이용하여 진단 데이터를 전송할 경우 데이터의 흐름을 그림 1에서 나타내고 있다. PACS(Picture Archiving and Communication System)에서 산출된 DICOM(Digital Imaging and Communications in Medicine) 형태의 자료나 의사가 진단한 정보를 갖고 있는 OCS(Order Communication System)의 데이터 등을 수입/수출하며, 각 진료과 및 외부 병원에 검색 결과를 XML 문서로 전송

하고, HL7 메시지 등을 포함할 수 있다. HIES는 정보공유관리자, 영상 압축/분할 모듈, 정보 추출기, HL7 변환기, 보안 모듈 등으로 구성된다[2,3].

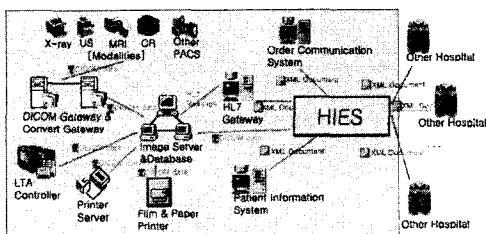


그림 1. HIES를 이용한 의료 데이터의 흐름도

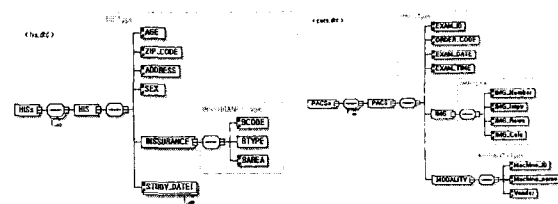
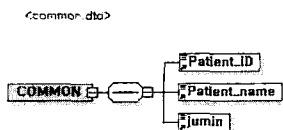
### 2.2 보안기법

HIES에서 정보 보안의 범주로 시스템 사용 및 접근 제어, 데이터 전송 등에 대해서 취급하였다. 시스템 사용과 접근 제어에 대해서는 취급할 데이터를 기초환자 정보, 진단 정보, 환자 건강 정보, 보험 정보, 처방전, 시스템 사용자 인증 정보로 분류하고 각 의료인의 등급을 두어 각 데이터의 접근 권한을 구분하였다[3]. 또한 정보 전송에서는 인터넷 상에서 데이터 전송 시 사용되는 SSL(Secure Socket Layer)/TLS를 사용하였다. SSL은 키 교환에 대해 handshake protocol, 교환 키 및 정보 확인을 위해 change cipher spec protocol, 문제 발생 시 alert protocol, 실제 자료의 암호화에 대한 레코드 프로토콜로 구성된다. SSL은 보안 단위로 세션을 이용하는데, 세션 기간에는 보안 서비스를 위한 세션키를 공유하고, 레코드 프로토콜에서 실질적인 보안 서비스를 제공한다. 추가적인 보안 사항으로 전자 서명에 기반한 challenge-response 기법을 사용하여 사용자 인증, 무결성 및 부인봉쇄 등을 제공한다. 사용자는 공개키 암호화 기법에 사용되는 자신의 공개키, 비밀키 쌍을 가지고 있어야 한다. 공개키 알고리즘은 RSA(Rivest-Shamir-Adleman)[8]를 사용하며, 해쉬 알고리즘은 SHA-1[9]을 사용한다.

### 3. HIES의 확장

#### 3.1 XML Schema

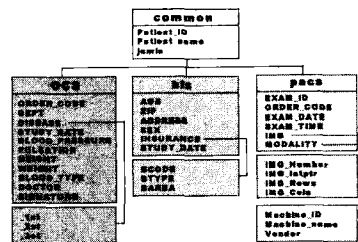
환자의 기초 정보 및 원부 데이터 등은 HIS(Hospital Information System), 의사가 진단한 내용은 OCS, 의료 장비 등에서 산출된 진단 정보는 PACS 등에 저장되어 있다. 만약 임의의 환자에 대한 진료 내역을 보고자 한다면 각 시스템에 있는 데이터를 통합하여 보이는 것이 효율적이다. 이와 같은 구조를 나타내기 위하여 진료 결과로 나타낼 항목을 XML DTD를 이용하여 그림 2에 예시한다. HIS와 PACS에서 사용되는 엘리먼트 중 공통으로 나타나는 항목인 환자번호, 환자이름, 주민등록번호는 중복 선언하지 않고 "common.dtd"에 저장하였다. OCS나 PACS에서 이를 공통으로 사용하기 위해서는 "common.dtd"를 "his.dtd"와 "pacs.dtd"에서 외부 파라미터 엘리먼트로 선언하는데, 이는 HIS와 PACS 관련 XML 문서 내에 중복 객체 선언으로 유효하지 않다. 이 문제는 xlink와 xpointer를 이용하여 해결할 수 있으나, 해결하더라도 차후, DTD 구조가 변경되었을 경우 많은 DTD 수정이 요구된다[10].



```
[his_dtd]
<?xml version="1.0" encoding="euc-kr" ?>
<ENTITY % common SYSTEM "common.dtd">
<ELEMENT HISs (HIS)+>
<ELEMENT HIS (AGE, ZIP_CODE, ADDRESS, SEX,
INSURANCE, STUDY_DATE)+>
<ELEMENT AGE (#PCDATA)>
.....
```

그림 2. HIS와 PACS의 XML DTD

그림 2의 XML DTD 구조를 XML Schema로 변경하면 그림 3과 같다. 그림 3은 환자번호, 환자이름, 주민번호와 같이 공통으로 쓰이는 속성을 상위 객체 "common"으로 정의하고, 하위 객체 "OCS", "PACS"에서 사용하도록 상속(inheritance)과 복합합(composition) 구조를 사용한다. 만약, 스키마 구조의 수정과 추가 작업이 발생하더라도 베이스 확장(extension base)을 이용하여 간단히 처리할 수 있다.



```
<xsd:element name="Patient">
  <xsd:complexType name="common">
    <xsd:sequence>
      <xsd:element name="Patient_ID" type="xsd:string" />
      <xsd:element name="Patient_name" type="xsd:string" />
      <xsd:element name="jumin" type="xsd:string">
        <xsd:pattern value="\d{6} - \d{7}" />
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="OCS">
  <complexContent>
    <extension base="common">
      <xsd:sequence>
        <xsd:element name="ORDER_CODE" type="xsd:string" />
        <xsd:element name="DEPT" type="xsd:string" />
        <xsd:element name="DISEASE" type="disease_seq" />
      </xsd:sequence>
    </extension>
  </complexContent>
</xsd:complexType>
```

그림 3. XML Schema 표현

#### 3.2 정보 공유

HIES에서 정보 공유는 정보 공유 관리자(Information Sharing Manager)가 담당한다[2,3]. 기존 정보 공유 관리자는 스키마나 데이터 충돌(conflict)이 발생할 경우, 지식 베이스(knowledge-base)의 정보를 참조한다. 스키마 충돌은 자료를 공유하는 과정에서 각 사이트의 내부 구조가 상이한 경우에 발생하는 것으로 테이블 이름이나 속성 이름의 차이가 원인이 될 수 있으며, 데이터 충돌의 경우는 예를 들어, 같은 환자의 이름 "홍길동"과 "KilDongHong"에 대해 다르게 인식하기 때문이다. HIES는 이와 같은 충돌 발생 시 지식 베이스에 있는 요소 값(factor id)을 검색한다. 동일한 요소 값을 갖는 결과들을 추출하여 충돌이 해결될 때까지 결과 속성이나 데이터 값을 대치(replace)시킨다. 하지만 요소 값을 알아내는 루틴이나 충돌 해결 루틴은 순차적인 작업으로 XML Schema의 네임스페이스와 속성을 이용하면 개선된 결과를 얻을 수 있다. 그림 4는 효과적으로 충돌 문제를 해결하기 위한 함수 "find\_id" 알고리즘이다. XML 문서에 <name id="1004"> Magarette </name>과 같

이 속성이 정의되어 있다고 가정하자. 요소 값은 속성인 "1004"로 대체되며, 외부 시스템과 자료 공유를 할 수 없는 경우 속성 값 "1004"와 같은 속성 필드 리스트를 산출한다. 리턴된 속성 필드 리스트의 값들을 대체시켜 충돌을 해결할 수 있다. 또한 name이라는 속성에 들어가는 인스턴스에 대한 데이터 충돌도 고려해야 한다. "name" 속성은 환자의 이름으로도 표현될 수 있고, 의료 장비(Modality)의 이름으로도 표현될 가능성이 있다. 이와 같은 경우 고유의 속성을 나타내는 네임스페이스를 비교한다면 같은 의미를 가진 필드인지를 알 수 있다.

```
function find_id(attr_id, namespace_list) : match_item
global_find_check = FALSE;
if not match a attr-name in other-side database {
    make_id_list(attr_id);
}
else
    if(compare(namespace_list))
        global_find_check = TRUE;
while(global_find_check == FALSE) {
    if(compare(attr_id, id_in_list)) {
        replace(id, id_in_list);
        global_find_check = TRUE;
    }
    else
        next(id_in_list);
}
if(global_find_check == FALSE)
do_alter_table_routine(redefinition_schema);
```

그림 4. 함수 find\_id

3.3 동적 스키마

환자가 진료를 받기 위해 내원했을 경우 초진/재진/입원 등으로 구분할 수 있다. 만약 입원 환자의 경우 키, 몸무게, 혈액형 등의 진료 기초 정보가 필요하지만, 일반 내원 환자의 경우 진료 결과에 이 데이터를 포함하지 않아도 무방하다. 또한 진단 결과를 입력할 때 그림 3에서 진단1, 진단2, 진단3으로 정의된 항목에 데이터가 기입된다. 하지만, 환자가 3회 이상 내원할 경우 데이터를 입력할 엘리먼트가 존재하지 않게 된다. 이와 같은 경우 XML Schema에서 제공하는 동적 스키마를 이용하면 효율적이다. 동적 스키마를 이용한 입원 환자 처리는 다음과 같다.

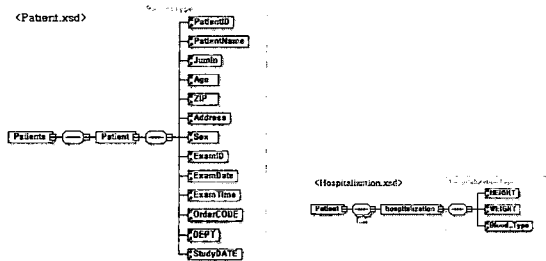


그림 5. 동적 스키마 표현을 위한 스키마 조각

```
<?xml-stylesheet href="Patients.xsd" type="text/xsl" />
<?xml-stylesheet href="Hospitalization.xsd" type="text/xsl" />
<xsl:template match="Patients">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match="Patient">
  <p>
    <xsl:if test="//Patient/hospitalization">
      <xsl:value-of select="Height"/>
      <xsl:value-of select="Weight"/>
      <xsl:value-of select="Blood_Type"/>
    </xsl:if>
  </p>
</xsl:template>
```

그림 6. 동적 스키마 처리 XSL

그림 5의 "hospitalization.xsd"는 키, 몸무게, 혈액형 항목을 포함하는 스키마 조각이다. 그림 5의 스키마 조각은 그림 6에서 표현된 XSL(eXtensible Style Language)을 이용하여 추가된다. "xsl:if" 문을 이용하여 XML 문서의 구조를 검사하는 데

"//Patient/hospitalization" 조건이 참이면 "hospitalization" 스키마 조각의 엘리먼트를 반환하여 "Patient"에 동적으로 추가한다[11]. "//Patient"와 "/hospitalization"은 그림 4의 "patient.xsd"와 "hospitalization.xsd"에 표현되고 있다. 또한, 그림 5에서 스키마 조각을 동적으로 추가하였는데 환자의 키를 알 수 없어, "HEIGHT" 항목을 입력하지 않았다고 가정하자. 이는 스키마 조각에 해당되는 엘리먼트 데이터를 XML 문서 내에 입력하지 않은 것이지만 유효한 문서로 검증된다[7,11]. 따라서 각 의료 업무 단위에 대해 세세히 정의, 분류하지 않고, 포괄적인 스키마 조각을 정의하여 사용할 수 있다.

3.4 정보 보안

그림 3의 전체 구조에서 OCS는 마지막 필드로 "SIGNATURE"를 갖고 있다. 이는 의사가 진단을 내린 후 전자 서명이 들어가는 항목으로 사용자 인증, 메시지 무결성, 부인 봉쇄를 제공할 수 있다[3]. XML 전자 서명은 W3C에서 XML Signature 표준 스펙[12]을 제공하고 있다. 본 시스템에서는 XML 문서 전체에 대해서 암호화 할 수 있으나, 일부분을 서명함으로써 전자 서명 생성 및 검증 등에서 효율을 높일 수 있다. "SIGNATURE" 필드는 비밀키 암호화 기법을 이용하여 처리한다. 암호화 및 암호화 키 전송에는 공개키 시스템을 사용한다. 공개키 시스템은 공개키와 비밀키로 이루어져 있으며, 안정성을 높이기 위해 대칭키 암호화 기법에 의해 한 번 더 암호화되어 있다. 비밀키는 사용자로부터 받은 비밀번호를 이용하여 해쉬하고, 해쉬된 값을 키로 하여 암호화된 비밀키로 복호화한다.

4. 결론 및 향후 연구과제

의료 정보 시스템의 연구 및 활용 범위가 급진전하고, 사용 범위가 전자 확대됨에 따라 시스템의 내부 구조도 점점 복잡해지고 있다. 본 논문에서는 XML Schema를 이용한 HIES의 기능 확장에 대하여 논하였다. 복합 구조의 의료 정보 데이터를 보다 유연성 있게 표현 가능하며, 효율적인 정보 공유 방식 및 시스템 사용 중에 동적으로 데이터 구조를 변경하는 기법 등을 도입하였다. 현재 XML DTD를 XML Schema로 확장하는 변환기의 동작은 반자동 형태로 차후 자동에 가까운 효율적인 변환 알고리즘을 개선하고자 한다.

참고문헌

[1] "Health Level 7," <http://www.hl7.org>  
 [2] 홍동완, 유지희, 남궁숙, "XML문서를 이용한 병원정보교환 시스템" 대한의료정보과학회지, Vol. 7, No. 2, 2001.  
 [3] 홍동완, 주한규, "의료 정보 시스템의 정보 보안," 한국정보과학회 학술발표논문집, Vol 28, No. 2, 2001.  
 [4] R. Dollin, P. Biron, "HL7 Recommendation : Using XML as a Supplementary Messaging Syntax for HL7 Verion 3.0," HL7 XML SIG, 1999.  
 [5] Health Level 7 HIMSS 99, <http://www.hl7.org/special/demo/Samples.htm>  
 [6] Kawaguchi, K., "W3C XML Schema Made Simple," XML.com., 2000  
<http://www.xml.com/pub/a/2001/06/06/schemasimple.html>  
 [7] XML Schema Part 0: Primer  
<http://www.w3.org/TR/2000/WD-xmlschema-0-20000407/>  
 [8] R. Rivest, A. Shamir, L. Adleman, "A method for digital signature and public-key cryptosystems," Communications of the ACM, Vol. 21, pp. 120-126, 1978.  
 [9] FIPS 180-1, "Secure hash standard," Federal Information Processing Standards Publication 180-1, U.S. Dept. of Commerce / NIST, 1995.  
 [10] D. Lee, W. W. Chu., "Comparative Analysis of Six XML Schema Languages," ACM SIGMOD Record, 29(3), pp. 76-87, 2000.  
 [11] K. Wang, "Natural Language Enabled Web Applications," pp. 15-22, NLPFRS, 2001.  
<http://www.alphaworks.ibm.com/formula/dynamicxmlforjava>  
 [12] W3C, XML-Signature Syntax and Processing, Candidate Recommendation, 19-April-2001.  
<http://www.w3.org/TR/xmlsig-core>