

CDA 문서를 관계형 데이터베이스에

저장 관리하기 위한 시스템

박수진^o 김일곤* 조훈** 곽연식**

경북대학교 컴퓨터학과*

경북대학교 의과대학 의료정보학교실**

sjply@hotmail.com^o, ikkim@knu.ac.kr, hunecho@knu.ac.kr, yskwak@knu.ac.kr

System for storing and managing CDA documents in the relational database

Soojin Park^o Ilkon Kim Hune Cho YeonSik Kwak

Department of Computer Science, Kyungpook National University*

Department of Medical Informatics, Kyungpook National University School of Medicine**

요 약

본 논문에서는 병원간의 정보를 교환하기 위한 목적으로 정의된 의료 분야의 표준기관인 HL7에서 제안하는 Clinical Document Architecture(CDA)문서를 관리하기 위한 시스템을 제안한다. CDA Manager라는 이 시스템은 각 병원의 관계형 데이터베이스인 CDA local repository에서 CDA문서를 저장, 검색, 수정의 기능을 제공한다. 본 논문에서는 XML로 기술된 CDA문서의 특성을 살려 테이블을 설계하고 검색속도를 향상시켰다.

1. 서 론

의료 분야의 전산화에 따른 많은 메시지와 문서를 처리하고 여러 다양한 병원이나 관련된 기관에서 서로 정보를 교환하기 위한 표준화 작업이 활발하게 이루어지고 있다. Health Level 7(HL7)은 병원간의 정보를 교환하기 위한 목적으로 설립된 표준화 기관으로 병원 업무와 관련된 다양한 메시지와 의료 문서를 정의하고 있다[1]. HL7 표준들 중에 퇴원기록지나 진단서와 같은 의료 문서의 구조를 정의하는 것을 Clinical Document Architecture(CDA)라고 한다. 이것은 일반적인 텍스트 문서가 아닌 계층적인 구조를 가지고 문서를 구조적으로 표현할 수 있는 XML로 표현된다. CDA 문서는 일반적인 XML문서와 달리 포맷이 정해져 있으므로 저장 방법에 차이가 있다. 본 논문에서는 CDA local repository에 CDA 문서를 저장, 수정, 검색하기 위한 방법을 제시한다. 우선 2장에서 현재까지 제시된 XML-to-relational mapping 기술과 CDA에 대해 간략히 소개하고 3장에서 CDA문서 관리를 위한 Manager의 설계를 설명한다. 4장에서는 구현방법과 평가를 설명하고 마지막으로 5장에서 결론과 향후 과제로 마무리를 짓겠다.

2. 관련 연구

2.1 XML-to-relational mapping

현재까지 존재하는 여러 가지 XML-to-relational mapping 기술은 세 가지 방법으로 분류한다[2].

첫째로 일반적인 기술은 DTD나 XML schema를 사용하지 않고

문서의 element, attribute와 문서의 구조를 저장하는 방법으로 Edge, attribute, universal로 나뉜다[3]. 가장 간단한 것은 모든 attribute를 하나의 테이블에 저장하는 것으로 Edge테이블에 저장하는데 이것은 Edge(source, ordinal, name, flag, target)으로 구성된다. 같은 이름을 가지고 있는 attribute는 하나의 테이블에 저장하는 방법으로 이것은 AttributeName(source, ordinal, flag, target)으로 구성된다. 문서의 모든 attribute를 하나의 universal table에 저장하는 방법은 attribute 테이블의 outer join의 결과라고 할 수 있다. 이는 아래와 같이 구성된다.

Universal (source, ordinal₁, flag₁, target₁, ordinal₂, flag₂, target₂,... ordinal_n, flag_n, target_n)

두번째 방법은 DTD나 schema에 존재하는 모든 element를 각각 테이블로 만들어 표현하는 것으로 DTD를 참조하는 것은 fixed mapping이라 하고 스키마를 참조하는 것은 flexible mapping이라 한다. 반복되는 하위 element는 하나의 독립된 테이블에 저장하고 반복되지 않는 것은 parent테이블에 저장한다. 널 값을 허용하는 필드를 사용하면 optionality, 여러 가지 테이블을 사용하거나 널 값을 허용하는 universal테이블을 사용하는 경우에는 choice로 표현한다. 세번째 방법은 사용자의 요구에 따라 다양한 방법으로 저장하는 것으로 commercial system에서 사용된 예가 있다[4].

본 논문에서 데이터베이스에 저장하고자 하는 CDA문서는 XML문서이지만 나름의 특성을 가지고 있어 이 문서를 잘 다루기 위해 위의 세 가지 방법과는 다르게 테이블을 구성하였다.

2.1 CDA

HL7의 Clinical Document Architecture(CDA)는 문서 교환을 목적으로 한 임상문서를 구조적으로 정의한 문서 표준으로써 이것은 텍스트, 이미지, 사운드와 다른 멀티미디어 콘텐츠를 포함한다[5]. 임상문서는 아래와 같은 특징을 가진다.

- Persistence: 임상문서는 정해진 시간 동안 변하지 않고 문서 그대로를 유지해야 한다.
- Stewardship: 임상문서는 정해진 사람이나 기관에 의해 유지해야 한다.
- Potential for authentication: 임상문서는 법적으로 인증 받을 수 있는 정보들로 이루어진다.
- Wholeness: 임상 문서의 인증은 문서의 한 부분이 아닌 문서 전체에 대해서 적용 된다.
- Human readability

CDA문서는 XML로 표현하고 RIM에서 derive되었으며 HL7 version 3 데이터 타입을 사용하고 완전한 CDA는 계층적인 구조를 가진다. 이것은 HL7메시지의 한 segment나 element에 MIME type으로 인코딩되어 교환된다. 2000년에 CDA release 1이 정해지고 현재 release 2가 진행중인데 둘 사이에는 구조적으로 큰 차이점을 가진다. 본 논문에서는 CDA release 1을 기반으로 하였기 때문에 이에 대해서 설명하도록 한다.

CDA release 1 문서의 구조는 header와 body로 나누어지고 CDA 레벨에 따라 서로 다른 스키마를 가지게 된다. CDA header는 <clinical_document_header>라는 element 내에서 정의되는데 문서를 구분하고 인증, 환자, 진료, 의사에 관한 정보를 포함하고 body는 clinical report를 포함한다. 현재 CDA release 1은 CDA header와 Level One body만 스키마로 정의되었다.

CDA에서 레벨이란 문서의 콘텐츠 포함 범위에 따라 정해지는데 레벨이 높아짐에 따라 더 세부적이고 구체적인 내용을 포함한다. 아래 그림은 CDA release 1 level one 문서의 간략한 예를 보여준다. 문서의 header에는 provider와 patient의 정보를 포함하고 body에는 comment를 포함한다.

```
<?xml version="1.0" ?>
<!DOCTYPE levelone [View Source for full doc type... ]>
<levelone>
  <clinical_document_header HL7-NAME="document_service_as_clinical_document_header"
    T="service" PIM-VERSION="0.9B">
    <provider HL7-NAME="has_service_actor" T="service_actor">
    <patient HL7-NAME="has_service_target" T="service_target">
    <local_header ignore="all" descriptors="MyLocalTag">
  </clinical_document_header>
  <body confidentiality="CONF">
    <section>
      <caption><Comment></caption>
      <paragraph>
        <comment>최심실의 크기는 정상이며 systolic function은 정상으로 calculated EF는 46%. 좌심방, 우심방 및 우심방의 크기는 정상이고, 좌심실벽의 두께는 정상이며 regional wall motion abnormality는 관찰되지 않음. 승모판 및 대동맥판막은 정상기능을 가지고 있으며 trivial TR이 동반되어 있음. TR velocity로 부터 추정된 肺동맥속기압은 mmHg로 정상임. </comment>
      </paragraph>
    </section>
  </body>
</levelone>
```

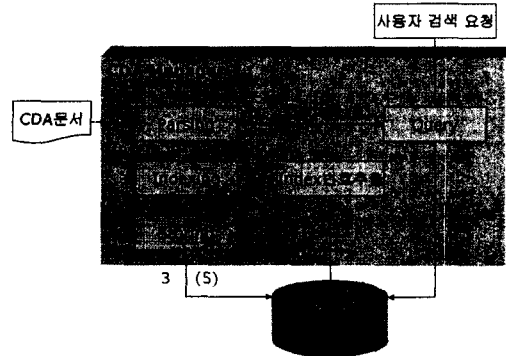
<그림 1. CDA 문서의 예>

3. 시스템 설계

3.1 CDA Manager

CDA Manager는 생성된 CDA 문서를 CDA local repository에 저장, 검색, 수정의 기능을 제공한다.

우선 CDA문서가 입력으로 들어오면 그 문서를 parsing하여 버전 번호를 확인하여 처음으로 입력되는 문서인지 수정되는 문서인지를 확인한다. 처음으로 입력되는 문서일 경우에는 index값을 추출하고 문서에 적합하게 설계된 테이블에 저장한다. 수정되는 문서일 경우에는 이미 저장된 문서의 인덱스를 추출하여 그 위치에 데이터를 입력한다. 사용자가 검색하는 경우 요청에 적합한 쿼리를 테이블로 보내고 그 결과를 웹에 보여준다.



<그림 2. CDA Manager 설계>

3.1.1 CDA 문서의 저장

CDA 문서는 특성상 문서의 원본을 유지하여야 한다. 즉 문서를 쪼개서 정보만 데이터베이스에 저장하는 것이 아니라 문서 전체를 저장하게 된다. 이런 경우 데이터베이스에서 문서 전체를 한 필드에 저장하게 되면 문서의 검색이 불가능해지기 때문에 문서에서 검색에 필요한 값을 추출하여 각 필드에 저장하게 된다.

CDA문서를 생성되면 이를 CDA Manager에게 보내고 CDA Manager는 이 문서를 파싱하여 인덱스를 추출한다. CDA문서는 XML의 형식으로 되어 있으나 일반적인 XML과는 달리 모든 element에 대해 검색할 필요가 없고 특정 element의 값으로 문서를 확인할 수 있어 아래에 있는 CDAINDEX table의 필드 값만을 저장한다.

<표 1. CDAINDEX table>

index	Patient id	Patient name	Provider id	doc type cd	Encoun ter tmr	Final impre ssion	com ment
-------	------------	--------------	-------------	-------------	----------------	-------------------	----------

index, Patient id, Patient name, Provider id, Document type cd와 Encounter tmr은 문서를 확인하는데 필요한 값이므로 not null이고 Final impression과 Comment는 문서의 검색에만 사용되는 필드로 null도 허용한다.

CDA문서는 CDADOC table에 저장되는데 CDAINDEX table의 index값과 CDADOC의 index가 일치하도록 하여 사용자가 검색할 때 index값이 일치하는 CDA문서를 가져 올 수 있게 한다. CDA 문서 전체는 clob의 형태로 저장되고 문서에 포함되는 이미지는 blob의 형태로 저장된다. CDA문서는 not null이고 이미지는 문서 자체에 없을 수도 있기 때문에 null을 허용한다.

<표 2. CDADOC table>

index	CDA document	IMAGE
-------	--------------	-------

문서의 타입 코드를 정의하는 테이블을 하나 두고 CDAINDEX 테이블의 doctypecd와 연결하여 문서타입으로도 검색이 가능하게 한다. 문서 타입 코드는 현재 의료계에서 사용되는 LOINC 코드로 표기한다.

<표 3. Document type code table>

LOINC	description
-------	-------------

3.1.2 CDA 문서의 검색

저장된 CDA문서는 병원에서 의사나 간호사 등의 provider나 관계된 사람들이 검색을 하여 원하는 문서를 볼 수 있다. 문서는 환자의 id, 환자의 이름, provider id, 문서 종류, 방문 날짜, 병명이나 특정 단어와 관련된 키워드 검색이 가능하다. 환자나 문서의 정보에 의한 검색은 CDAINDEX table에서 정의한 필드에 대해서 인덱스를 생성하여 수행하고 키워드 검색은 Final Impression과 Comment의 내용을 text search를 통하여 원하는 단어가 있는 문서를 보내주게 된다. 검색된 문서가 여러 개일 경우는 각각을 선택할 수 있는 항목을 주어 사용자의 편의를 고려한다.

3.1.3 CDA 문서의 수정

CDA문서를 수정하는 방법은 두 가지가 있는데 하나는 CDA builder에서 수정된 값에 대하여 문서를 다시 생성하는 것과 다른 하나는 문서를 불러와서 사용자가 직접 값을 바꿔 저장하는 것이다.

첫 번째의 경우 CDA Manager에 입력된 문서는 <version_nb> element의 값으로 저장할 위치를 판단하게 된다. 가장 처음으로 생성된 문서는 <version_nb v = 1>, 후에 수정된 문서는 <version_nb v = 2>로 정의한다. CDA Manager는 version_nb의 값이 1이면 index를 추출하여 저장하고 값이 2이면 검색을 통하여 이미 저장된 문서가 있는 index값을 추출해 그 위치에 새로운 문서를 update한다.

두 번째 방법은 사용자가 문서를 바로 수정 할 수 있는 인터페이스를 구성하여 각 element의 값을 수정하고 저장 할 수 있도록 한다.

4. 구현 및 평가

구현 환경은 아래와 같다.

<표 4. 구현환경>

Operating System	Windows 2000 professional
Language	Java version 1.4.1
Database	Oracle 9i
Parser	Oracle XML parser
JDBC	JDBC Level4(oracle)

CDA문서의 구조는 일반 XML문서와는 달라 테이블 구성도 기존에 사용된 방법과 차이가 있고 저장할 때 한번의 파싱 과정을 거치기 때문에 바로 저장하는 것보다 약간의 시간이 걸리지만 검색

에서는 CDA문서의 특성을 잘 살려 XML 데이터베이스를 사용하지 않고도 기존에 널리 사용되던 관계형 데이터베이스에 맞게 테이블을 구성하여 XML문서에 대한 검색이지만 XDB에서의 검색 보다 훨씬 빨랐다.

<표 5. 검색 수행 시간 비교>

	RDB(인덱스추출)	XDB
환자 ID	00:00.0781250	01:46.1406250
	00:00.0625000	01:46.1875000
진료결과 (심장병)	00:00.0935000	01:46.2425000
	00:00.1237500	01:46.5625000

위 결과는 oracle 9i에서 만개의 데이터에 대해 환자의 아이디와 심장병을 가진 환자를 검색 할 때 본 논문에서 제시한 인덱스를 추출하여 검색하는 방법과 oracle에서 제공하는 XDB로 검색하는 방법의 속도를 비교한 것이다. XDB를 사용하면 검색할 때마다 문서를 전부 파싱하기 때문에 시간이 오래 걸리지만 원래 RDB의 검색 방법을 사용하면서 테이블은 CDA문서에 적합하게 구성하여 검색시간도 단축하고 관리도 관계형 데이터베이스에서의 방법을 그대로 사용할 수 있다.

5. 결론 및 향후과제

본 논문에서는 의료 표준 기관인 HL7에서 제안한 CDA를 저장, 관리하는 시스템을 개발하였다. HL7은 현재 병원마다 다른 포맷을 가진 의료문서를 표준화 시켜 어느 병원에서든 표준을 따르면 문서의 교환과 공유가 가능하게 한다. CDA는 진료와 문서의 정보를 포함하는 header와 문서의 내용을 포함하는 body로 이루어진 XML로 기술된 의료 문서이다. 이것은 의료 문서라는 특수한 것으로 문서를 데이터베이스에 저장할 때 쪼개서 넣지 않고 원본을 그대로 저장하게 되고 문서에 대한 모든 element에 대한 검색이 필요 없다. 본 논문에서는 XML을 위한 XDB를 사용하지 않고 관계형 데이터베이스에 인덱스를 추출하는 방식으로 저장하여 성능을 향상시킬 수 있었다.

현재는 CDA release1를 바탕으로 구현하였지만 앞으로 release2가 완성되면 이를 제공하는 시스템도 필요하게 되고 관계형 데이터베이스뿐만 아니라 다른 데이터베이스에 대해서도 적용할 수 있는 테이블 구성과 시스템 개발이 필요하다.

6. 참고 문헌

[1] Health Level 7, www.hl7.org
 [2] S. Amer-Yahia, M.Fernandez, Techniques for Storing XML(Tutorial), ICDE, 2002
 [3] D.Florescu, D.Kossman, A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database, IEEE Data Eng. Bulletin, 1999.
 [4] A. Sihem, S. Divesh, A Mapping Schema and Interface for XML Stores, WIDM'02, 2002
 [5] L. Alschul, R. H. Dolin, Clinical Document Architecture Framework release 1.0, hl7, 2000