

# 멀티미디어 데이터에 대한 XML 문서 저장관리 시스템 설계 및 구현

안영희\*, 황부현\*

\*전남대학교 전산학과

e-mail:goodbye9@sunny.chonnam.ac.kr

bhhwang@chonnam.ac.kr

## A Design and Implement of XML Document Storage Management System for Multimedia Data

Young-Hee An\*, Bu-Hyun Hawang\*

\*Dept. of Computer Science, Chonnam National University

### 요 약

인터넷이 정보의 바다가 된 오늘날에 많은 정보를 효과적으로 표현하기 위해 텍스트뿐만 아니라 이미지 동영상 등 멀티미디어 데이터의 표현이 많이 사용되고 있다. 이러한 대용량의 멀티미디어 데이터가 증가함에 따라 대용량의 데이터를 효과적으로 처리하고 저장하기 위한 많은 연구개발이 이루어지고 있다. 본 논문에서는 웹의 표준인 XML을 이용하여 비디오 데이터를 XML 문서로 표현하고, 관계형 데이터베이스에 XML 문서를 저장할 수 있는 스키마를 생성한다. 그리고 생성된 스키마를 이용하여 관계형 데이터베이스를 기반으로 하는 XML 문서 저장관리 시스템을 구현한다.

### 1. 서론

XML(eXtensible Markup Language)은 SGML(Standard Generalized Markup Language)과 HTML(Hypertext Markup Language) 각각의 장점을 수용하고 있는 웹의 새로운 문서 표준으로써 사용되고 있다[1]. 이에 따라 대량의 XML 문서들을 효율적으로 저장하고 관리하는 시스템이 필요하다[2].

기존의 관계형 데이터베이스에 XML 문서를 저장하는 경우 서로 구조가 다르기 때문에 사상(mapping)하는 방법에 있어 매우 복잡하다[3]. XML 문서는 일반 텍스트 문서와는 달리 내용 정보뿐만 아니라 구조정보를 지니고 있기 때문에 구조정보를 활용하기 위해서는 두 모델의 구조적 차이에 대한 보완을 위해 다른 방식으로 처리되어야 한다[4].

XML 문서는 다양한 멀티미디어 데이터를 표현

할 수 있다. 대용량의 멀티미디어 데이터에 대한 사용자의 다양한 요구를 보다 효과적으로 처리하기 위해서는 데이터의 특성을 고려한 효율적인 저장 시스템이 필요하다.

본 논문에서는 멀티미디어 데이터를 표현한 XML 문서를 관계형 데이터베이스에 저장하기 위한 사상 기법으로 스키마를 생성하여 XML 문서를 효율적으로 저장 관리할 수 있는 XML 문서 저장관리 시스템을 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 XML 문서를 데이터베이스에 사상하기 위한 기법에 대해 살펴보고 3장에서는 전체 시스템을 설계하는데 있어 필요한 고려사항을 알아본다. 4장에서 본 논문에서 제안하는 사상 기법을 통해 비디오 DTD와 관계형 데이터베이스를 사상하기 위한 스키마를 설계한다. 5장에서는 이러한 기법을 적용한 XML을 기반으로 한 멀티미디어 저장 관리 시스템 구현하고 6장에서는 결론 및 향후연구에 대해 논의한다.

본 연구는 정보통신부, 디지털 콘텐츠 저작도구 개발 과제 사업에 의해 지원되었음.

## 2. 관련 연구

XML 문서 저장관리 시스템에서 가장 먼저 선행되어야 하는 것이 데이터 모델링이다. XML 문서 모델링에 대해서 기존에 연구되었던 내용들을 살펴보면 DBMS 활용에 따른 데이터 모델링에서 관계형 모델과 객체지향 모델로 나눌 수 있고, XML 문서의 저장 방식에 따른 데이터 모델링으로는 분할 저장 모델, 비분할 저장 모델, 혼합 모델로 나눌 수 있다[4].

이러한 모델중에서 기존의 릴레이션 데이터베이스에서 XML 데이터를 저장하기 위한 스키마를 추출방법에 관한 여러 연구들이 있다. [5]에서는 XML 데이터를 엘리먼트와 에트리뷰트를 구분하지 않고 엘리먼트와 값을 하나의 테이블로 저장하는 방법과 엘리먼트와 값의 특성에 따라 여러 개의 테이블로 나누어 저장하는 방법을 제안했다. [6]에서는 XML 데이터를 엘리먼트, 에트리뷰트, 텍스트, 패스 네 가지 릴레이션으로 나누어 저장하는 방법을 제안했다. 이 방법은 릴레이션 스키마가 XML 문서의 DTD와 독립적으로 DTD의 중요 요소인 IDREFS(요소들 간의 링크를 만들어 주기 위한 속성타입)를 고려하지 않아 구조정보를 알아내기 어렵다. 이러한 문제점을 보완한 계승 엘리먼트 ID를 사용하는 방법이 있는데 이러한 방법은 엘리먼트의 부모/자식관계에 있는 엘리먼트 찾기, 자식 찾기 등에 대한 다양한 구조 정보 질의가 간단한 처리과정을 통해 쉽게 이루어질 수 있다. 그러나 문서의 추가나 변경은 많은 작업을 필요로 하고 각각의 엘리먼트에 ID를 부여한다는 것은 많은 공간 낭비를 초래할 수 있다.

## 3. 시스템 설계 시 고려사항

대용량의 멀티미디어 데이터를 포함하는 XML 저장시스템을 구현하기 위해 아래와 같은 방법을 참조하여 시스템을 구성한다

- XML 문서의 수정과 업데이트를 용이하게 하고 대용량의 멀티미디어 데이터를 지원하기 위해 분할 저장 기법을 사용한다.
- 구조정보를 효율적으로 활용하기 위해서 DTD의 의존적인 스키마를 사용한다.
- 비디오 데이터를 참조 모델링한다.

## 4. 스키마 설계

본 논문에서는 멀티미디어 데이터를 포함하는 XML 문서의 DTD를 설계하여 XML 문서의 유효성을 검사하고, 유효한 XML 문서를 관계형 데이터베이스에 효율적으로 저장하기 위해 본 논문에서 제시하는 사상기법을 통해 스키마를 설계한다.

### 4.1 비디오 DTD 설계 및 구현

#### 4.1.1 비디오 DTD 설계

멀티미디어 데이터 중에서 텍스트나 이미지와는 달리 비디오 데이터가 지니는 중요한 특징은 각각의 프레임 내에서의 객체들간의 공간적인 정보와 일련의 프레임들간의 시간적인 정보가 결합된 시공간 관계성을 통해 표현할 수 있으며, 비디오 데이터에 대한 사용자의 내용 및 개념 기반 검색을 수행하는데 있어 매우 중요한 역할을 한다[7]. 이러한 프레임들은 에피소드/씬 단위의 계층 구조로 구성되고 이 구조를 바탕으로 비디오 DTD를 아래와 같이 구성한다.

```
<ELEMENT meta(title,meta_duration,director,date,producer,production,genre,grade,ost_title,file_name)>
<ELEMENT episode(epi_title,epi_content,epi_file,scene*)>
<ELEMENT scene(sc_title,space,time,ost_sc_file,actor*)>
<!ATTLIST episode id #REQUIRED>
<ELEMENT epi_title(#PCDATA)>
<ELEMENT epi_content(#PCDATA)>
<ELEMENT epi_file(#PCDATA)>
<!ATTLIST scene id #REQUIRED>
<ELEMENT sc_title(#PCDATA)>
...
```

[그림 1] 비디오.dtd

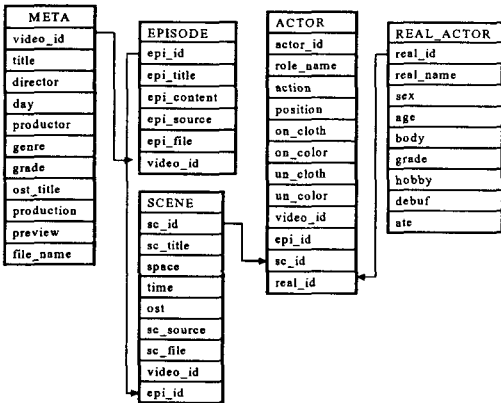
#### 4.1.2 스키마 설계

[그림 1]과 같은 DTD 형식을 따르는 XML 문서를 관계형 데이터베이스에 저장하기 위해 아래와 같은 사상 기법을 통해 [그림 2]와 같이 스키마를 설계한다.

#### <사상기법>

- ① DTD를 참조하여 자식 노드를 가진 복합(complex) 엘리먼트와 자식이 없는 원자(atomic) 엘리먼트로 분류한다.
- ② 같은 형제 노드의 원자 엘리먼트는 테이블의 속성이 된다.
- ③ 같은 형제 노드의 부모 복합 엘리먼트는 그 엘리먼트 테이블을 생성한다.

- ④ 각 테이블마다 테이블명 ID 속성을 생성하여 기본 키로 지정한다.(DTD에선 애틀리뷰트 ID가 이해속함)
- ⑤ 상위 테이블의 ID를 하위 테이블의 속성으로 포함하여 외래 키로 지정한다.

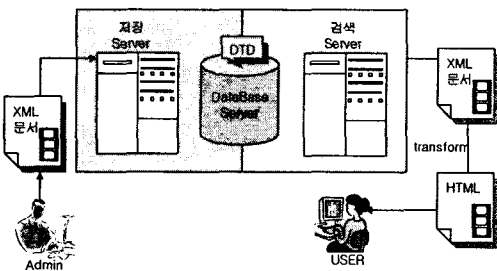


[그림 2] 비디오 스키마

5. XML을 기반으로 한 멀티미디어 저장시스템 구현

비디오 데이터를 단위별(Episode/Scene)로 저장 및 검색하는 Web Engine으로써 비디오를 보면서 사용자가 비디오에 대한 메타정보를 계층별로 입력할 수 있고, 원하는 비디오를 검색하여 해당 비디오를 관람할 수도 있으며, XML문서 video.xml를 얻을 수 있다.

5.1 시스템 구성도



[그림 3] 시스템 구성도

멀티미디어 데이터는 대용량인 경우가 많아서 시스템의 과부하와 오버헤드를 초래할 수 있다. 이러

한 문제점을 해결하기 위해 데이터베이스 서버, 저장 서버, 검색 서버를 각각 구성함으로써 시스템의 과부하를 최소화할 수 있다.

관리자는 비디오 데이터를 입력하고 검색할 수 있으며 원하는 정보만을 추출하여 데이터 교환 수단으로 비디오 XML 문서를 생성할 수 있다. 사용자는 웹 브라우저를 통해 원하는 비디오 일부분 에피소드/씬 단위로 검색하여 관람할 수 있고, 전체적인 비디오의 메타 정보를 구조적으로 알 수 있으므로 구조적인 검색이 가능하다.

5.2 구현

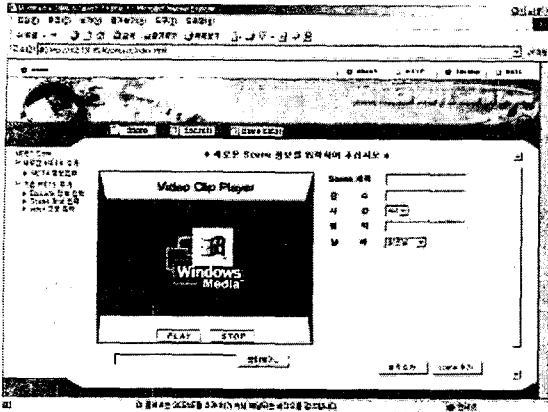
전체적으로 시스템을 XML과 JAVA로 구현하였고 비디오 자료를 분할하기 위해 windows XP/ME에 있는 Movie maker를 사용하였다.

시스템 사양	
저장 Server	· jdk1.3.1 · JDBC (classes12.zip) · Win2000 professional · Tomcat 3.3 / Apache (Java Servlet) · Movie maker (win XP/ME)
검색 Server	· jdk1.3.1 · JDBC · Unix · Tomcat 4/Apache(Java Servlet) · Xml parser ( xmlParserAPIs.jar ,xercesS.jar ,xercesimpl.jar)
DataBase Server	· Oracle 8.i(Win2000 Server)

[표 1] 시스템 사양

전체적인 웹 페이지 구성은 비디오 분할/ 저장/ 검색으로 크게 구분되어 있고, 사용법은 도움말을 보면 초보자라도 쉽게 사용할 수 있다. 저장 부분은 새로운 비디오 추가와 기존 비디오 추가로 구분되어 있는데 새로운 비디오 추가 탭을 사용하면 새로운 비디오에 대한 메타 정보부터 순서적으로 저장할 수 있다. 그리고 기존 비디오에 대한 추가 탭을 사용하면 기존에 저장되어 있는 비디오를 검색하여 에피소드 및 씬 단위로 각각 추가할 수 있고, 배우정보도 추가하여 배우정보로도 검색할 수 있다.

[그림 4]는 전체 시스템의 일부분으로써 씬별 장면을 저장하는 화면인데 비디오를 직접 보면서 내용 정보를 데이터베이스에 저장하는 화면이다.



[그림 4] SCENE 장면 저장하는 폼

## 6. 결론 및 향후과제

본 논문은 멀티미디어 데이터가 포함된 XML 문서를 관계형 데이터베이스에 저장 관리하는 시스템을 구현하였다. 이러한 시스템의 스키마는 XML 문서의 특징인 구조정보를 수용하기 위해 XML 문서의 DTD를 활용하여 엘리먼트를 복합/원자 엘리먼트로 분류하여 테이블과 애트리뷰트를 생성하고, r 각 테이블에 ID를 생성하는 방법을 사용하였다. 복합/원자로 분류하여 생성된 테이블에 각각 테이블단위로 ID를 생성하는 방법은 구조정보를 표현하기 위한 색인테이블을 별도로 생성하지 않아 저장공간을 줄일 수 있으며, 검색하기 위한 많은 조인연산 수를 줄일 수 있어 검색속도를 향상 시킬 수 있다.

또한 분할 저장 기법을 사용함으로써 대용량의 멀티미디어 데이터를 저장할 수 있는 장점이 있다. 본 시스템은 학습비디오나 방송 및 영상 콘텐츠 등에 다양하게 활용될 수 있다. 그러나 데이터베이스 관리자의 설계에 크게 의존하는 단점이 있어 향후에는 이러한 문제점을 해결해야 할 것이다.

## 참고문헌

- [1] J.Shanmugasundaram, K.Tufte, C.Zhang, G.He, D.J.Dewitand, J.F.Naughon, "Relational Database for Querying XML Document : Limitation and Opportunities", Proc. of 25th Int'l conf. on VLDB, Edinburgh, Scotland, UK, 1999
- [2] Simon, H., Startegic Analysis of XML for Web Application Development, Computer Research Corp., 2000
- [3] Mary Fernandez, Atsuyuki Morishima, Dan

Suciu, WangChiew Tan, "Publishing Relational Data in XML: the SilkRoute Approach", Published in IEEE Data Engineering Bulletin, no. 24(2), 2001

[4] 유재수, XML 저장관리시스템 동향, "http://www.dpc.or.kr/dbworld/document/9909/spec-4.html", 1999

[5] D. Florescu and D. Kossmann, "Storing and Querying XML Data Using an RDBMS", IEEE Data Eng.Bulletin, 1999

[6] T.Shimura, M.Yoshikawa, S.Uemura, "Sorage and Retrieval of XML Documents Using Object-Relational Databases", DEXA, 1999

[7] 심춘보, 장재우, "비디오 데이터에서 움직임 경로의 효율적인 검색을 위한 시그니처-기반 접근 기법", 추계학술발표회 논문집(a), 정보과학회, 2001