

관계형 모델에 대한 XML 계층 구조 사상 기법

안영희⁰ 황부현
전남대학교 전산학과

{goodbye9⁰, bhhwang}@sunny.chonnam.ac.kr

A Mapping Technique of XML hierarchical structure from Relational Model

Young-hee An⁰ Bu-hyun Hawang
Dept. of Computer Science, Chonnam National University

요 약

웹 상에서 다양한 데이터를 표현하고 정보교환을 위한 수단으로 등장하는 XML문서가 급속도로 증가하고 데이터베이스를 이용한 XML 문서 저장기법에 대한 많은 연구가 현재 진행되고 있다. XML 문서의 구조 정보를 활용하기 위해서는 기존의 문서와는 다른 계층적인 트리 방식으로 처리되어야 한다. 본 논문에서는 관계형 데이터베이스에 XML문서를 저장할 때 XML이 지니는 구조정보를 효과적으로 데이터베이스에 표현할 수 있도록 스키마를 생성하는 사상 기법을 제안한다. XML 문서를 엘리먼트 타입에 따라 분류하여 효과적으로 스키마를 생성하고, XML문서의 구조를 나타내기 위해 레코드(record)단위로 ID를 생성한다. 또한 멀티미디어 데이터와 같은 동적인 데이터를 포함하고 있는 XML문서를 효율적으로 저장할 수 있고 빠른 검색이 가능하도록 스키마를 설계한다.

1. 서 론

XML은 W3C에서 제안한 웹 상에서 데이터를 교환하기 위한 표준 언어이다[1]. 웹 상에 많은 정보를 XML 문서로 관리하는데 있어서 문서를 효율적으로 저장하고 관리하는 것은 가장 중요한 문제이다. XML문서는 문서 내부에 별도의 태그를 사용한 문서의 구조 정보를 담고 있고, 복잡한 구조와 다양한 멀티미디어를 표현할 수 있다. 그러나 기존에는 주로 텍스트나 이미지와 같은 정적인 데이터 타입에 대한 연구만이 활발히 진행되고 있을 뿐, 비디오·오디오와 같은 동적인 데이터 타입에 대한 연구는 거의 이루어지지 않고 있다[2][3].

XML문서를 기존의 RDBMS(Relational Database Management System)에 저장하여 관리하는 경우, 두 데이터 모델은 서로 구조가 다르기 때문에 사상(mapping)하는 방법에 있어 매우 복잡하다. 왜냐하면 관계형 데이터 모델은 단조롭고 표준적이며 평면적이고, 반면에 XML 데이터 모델은 연속적인 데이터를 허용하여 유연하고 복잡하며 비 표준적이기 때문이다[4]. XML 문서의 구조 정보를 충분히 활용하기 위해서는 두 모델의 구조적 차이에 대한 보완을 위해 다른 방식으로 처리되어야 한다[5].

본 논문에서는 비디오·오디오 데이터와 같은 동적 데이터 타입을 포함한 XML 문서를 RDBMS에 효율적으로

저장할 수 있고 빠르게 검색할 수 있는 사상 기법을 제안하고 스키마를 설계한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 XML 문서를 데이터베이스에 사상하는 방법에 관한 연구를 살펴보고, 3장에서는 어떻게 비디오에 대한 DTD를 생성하는지 살펴본다. 4장에서는 본 논문에서 제안하는 DTD를 참조하여 관계형 데이터베이스에 XML 문서를 사상할 수 있는 방법을 설명한다. 5장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

2.1 DTD에 의한 스키마 생성방법

2.1.1 DTD에 독립적인 스키마

DTD에 의존적이지 않고 반 구조적 데이터를 분석하여 관계형 모델로 변환하는 방법으로써 DTD의 정보가 인스턴스(instance)로 삽입되기 때문에 어떤 DTD라도 사상되지만, 테이블의 수가 고정적이다[2].

2.1.2 DTD에 의존적인 스키마

DTD에 의존적인 스키마는 DTD에 정의된 요소의 구조에 따라 다양한 형태의 테이블로 생성할 수 있는 방법으로 기본 인라인닝방법(Basic Inlining Technique), 공유 인라인닝 방법(Shared Inlining Technique), 혼합 인라인닝 방법(Hybrid Inlining Technique)이 있다[6].

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

2.2 XML 문서의 저장 구조

2.2.1 분할 저장 모델

XML 인스턴스를 엘리먼트별로 나누어서 저장하는 방식을 말하며, 문서의 실제 내용을 가지고 있는 각각의 단말 엘리먼트안에 문서의 내용이 분할되어 저장된다. 따라서, 문서의 구조적 정보나 일부 내용들이 수정되었을 때 관계되는 노드들만 수정하면 되므로 문서의 편집 및 관리가 쉽고, 동일한 내용을 갖는 노드들을 공유할 수 있다는 장점이 있지만, 문서의 내용을 추출하고자 할 때 각 단말 노드들을 순회하며 통합하는 과정에서 시스템의 성능을 저하시키는 문제가 발생한다[4].

2.2.2 비분할 저장 모델

분할 저장 모델과 같이 문서의 내용을 나누어서 저장하지 않고 문서 전체를 BLOB 형태로 저장한 다음, 각각의 단말 노드는 오프셋 정보를 가지고 접근하는 방식이다. 이는 문서를 한꺼번에 저장하였기 때문에 통합 과정이 필요 없어 문서 참조를 빨리 할 수 있지만, 내용의 일부만이 수정되었을 때도 문서 전체를 재구성해야 한다는 큰 단점이 있다[4].

이런 기법들을 사용하여 실제 연구된 방법으로는 계층 엘리먼트 ID를 사용하는 방법과 LETID 방법이 있다[7]. 계층 엘리먼트 ID를 사용하는 방법은 특정 엘리먼트의 부모/자식관계에 있는 엘리먼트 찾기, 자식 중 2번째 자식 찾기 등에 대한 다양한 구조 정보 질의가 간단한 처리과정을 거쳐 쉽게 설계될 수 있다. 그러나 문서의 추가나 변경시 많은 작업을 필요로 한다는 단점이 있다. LETID방법은 ID만 식별하면 해당 엘리먼트의 위치 및 구조정보까지 알 수 있으나 각각 엘리먼트 하나 하나에 ID를 부여한다는 것은 많은 공간 낭비를 초래할 수 있고, 내용검색을 지원하는 내용 색인, 구조 검색을 지원하는 구조 색인, 애트리뷰트 검색을 지원하는 애트리뷰트 색인으로 구성되어 있는데 색인 구조가 너무 복잡하다.

3. 비디오 DTD 설계

본 논문에서는 멀티미디어 데이터를 XML 문서로 표현한다. 비디오 데이터의 시공간적인 특징으로 인해 분류하고 그 분류에 따라 비디오의 DTD를 설계한다. 비디오 데이터를 계층 모델 링하여 XML 문서의 구조적인 표현을 설계한다.

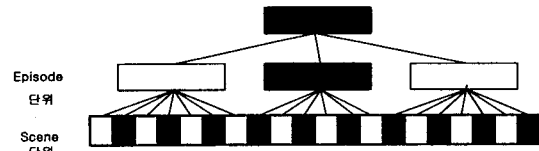
3.1 엘리먼트의 분류

엘리먼트의 성격으로 구분하면 엘리먼트는 complex 엘리먼트와 atomic 엘리먼트로 나뉜다. complex 엘리먼트는 자식 엘리먼트를 가지는 엘리먼트를 의미하며, atomic 엘리먼트는 자식으로 #PCDATA나 EMPTY 속성을 가지는 엘리먼트를 의미한다.

3.2 계층 비디오 모델 링

비디오 모델 링의 기본 단위는 세그먼트(segment)이

다. 세그먼트는 연속된 프레임의 집합으로, 그 각각 세그먼트 및 그것의 하위 부 세그먼트들의 종속 관계를 표현하면 트리 형태가 되고, 특정 비디오에 대해 이런 트리 구조를 만드는 과정을 계층 비디오 모델 링(hierarchical video modeling)이라 한다[8]. 본 논문에서는 비디오 데이터를 [그림 1]과 같이 episode/scene으로 구성한다.



[그림 1] 비디오 편집 단위 구조

3.3 DTD 설계

생성된 계층 구조를 참조로 비디오 DTD를 설계하기 위해 비디오의 단위 정보(episode/scene) 및 메타 정보, 배우정보를 포함한 비디오 DTD[그림 2]를 생성한다. 본 논문의 참조 모델은 멀티미디어 데이터를 포함하기 때문에 관계형 데이터베이스와 분할 저장 방식을 채택하여 대용량의 멀티미디어 데이터를 지원 가능하게 하고 수정을 용이하게 한다. DTD에 의존적인 스키마로 구성하여 주어진 DTD 구조에 따라서 특정 응용에 알맞은 저장 구조를 직접 설계할 수 있으므로 더 효율적인 처리가 가능하다.

```
<!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 2] 비디오.dtd

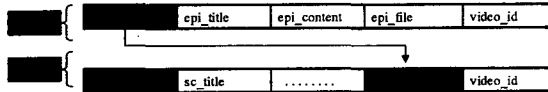
4. 스키마 사상 기법

XML은 문서의 내용에 의한 검색이외에 문서의 구조에 의한 검색도 지원한다. 구조화 된 문서를 효율적으로 검색하기 위해서는 키워드 인덱스 이외에 구조 정보에 대한 인덱스 구조가 필요하다[7]. 기존의 구조 정보를 표현하기 위하여 많은 색인 테이블을 별도로 생성하거나, 엘리먼트 각각에 ID를 부여하는 등 많은 테이블 생성 및 복잡성이 발생된다. 이를 해결하기 위하여 형제 노드의 atomic 엘리먼트는 속성으로 지정되고 형제 노드의 부모 complex 엘리먼트는 테이블 명으로 생성된다. DTD의 속성인 ID는 해당 도메인의 기본 키로 생성되고, 하위 구조의 테이블에 상위 구조의 ID(기본키)를 외래키 참조함으로써 XML 문서의 구조적인 표현을 한다. 이렇

게 함으로써 레코드 단위로 ID를 생성하게 되는데 엘리먼트 각각에 ID를 생성하는 것 보다 군집 화하여 ID를 생성함으로 보다 빠른 검색이 가능하고 저장공간도 줄일 수 있다. 그리고 엘리먼트와 테이블을 1:1로 매핑 하는 방법과 비교해 볼 때 많이 발생하는 테이블의 수를 줄일 수 있고 값을 가지지 않는 엘리먼트에 의해서 생겨난 테이블로 인해서 생길 수 있는 대량의 null값을 방지할 수 있다. 색인 테이블이나 엘리먼트 테이블, 애트리뷰트 테이블 등 따로 생성하지 하지 않아 보다 더 빠른 검색과 효율적인 스키마 생성이 가능하다. 아래의 [그림 3]의 DTD를 참조로 [그림 5]의 테이블을 생성한다.

```
<!ELEMENT episode(episode_title,episode_content,episode_file,scene*)>
<!ELEMENT scene(scene_title,space,time,ost,scene_file,actor*)>
<!ATTLIST episode id #REQUIRED>
```

[그림 3] DTD의 예



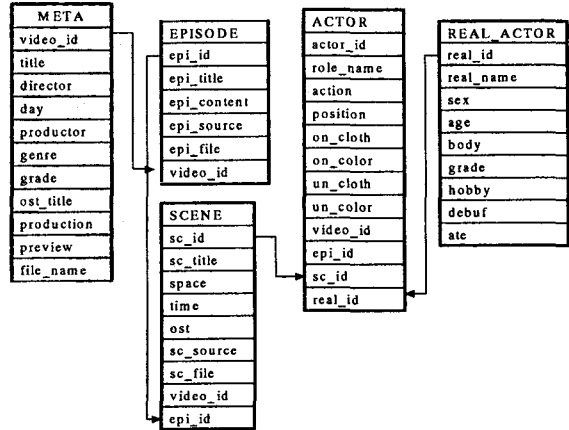
[그림 4] 스키마 매핑 방법

본 논문에서 제시하는 사상 기법을 참조하여 비디오.dtd를 [그림 5]와 같이 관계형 데이터 구조로 매핑하였다. 즉 이러한 사상 기법으로 인해 관계형 데이터 베이스에 계층 구조를 나타내기 위해 ID를 생성 할 때 엘리먼트 단위가 아닌 군집 화하여 데이터 레코드(record)별로 생성함으로써 많은 색인 테이블을 생성하는 것을 방지하고, 분할 기법의 단점인 XML 문서 생성 시 많은 순회를 해야 하는 단점을 보완한 것이라 할 수 있다.

- <사상 기법>**
- DTD를 참조하여 complex : atomic 엘리먼트로 분류
 - 형태 노드의 atomic 엘리먼트: 테이블의 속성으로 생성
 - 형태 노드의 부모 complex 엘리먼트: 테이블 명으로 생성
 - 테이블별로 ID 생성하여 기본키로 생성(DTD에서 애트리뷰트가 이해 속함)
 - 상위 테이블의 ID를 하위 테이블의 속성으로 포함하여 외래키로 지정한다

5. 결론 및 향후 연구

본 논문에서는 관계형 데이터베이스에 XML의 특성들을 손실 없이 표현할 수 있도록, 멀티미디어 데이터인 비디오·오디오데이터를 표현하는 XML 문서를 구성하여 관계형 데이터베이스로 매핑 시키는 사상 기법을 제안하고 설계하였다. 이러한 사상 기법을 통해 테이블이 많아지는 것을 방지하고 테이블의 구성을 최적화하고 군집화한다. 즉, 레코드 단위별로 ID를 생성하기 때문에 많은 색인 테이블을 생성하지 않아도 되고, 구조 정보 또한 데이터 손실 없이 표현할 수 있다. 따라서 XML문서를 간략한 스키마로 변환하여 관계형 데이터 베이스에 표현할 수 있어 검색 속도를 더욱 향상시킬 수 있다. 그러나



[그림 5] 비디오 스키마

전체문서의 삽입이 어렵다는 단점이 있다. 향후에는 이러한 단점을 극복하여 전체문서의 삽입이 가능하도록 스키마를 설계할 수 있는 방법이 연구되어야 할 것이다.

6. 참고문헌

- [1] T. Bray, J. Paoli, and C.M. Sperber-McQueen, "Extensible Markup Language(XML) 1.0", [http:// www. w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml). W3C Recommendation 6. October. 2000.
- [2] Alin Deutsch, M.F. ernaldex, D.Suciu, "Storeing Semi-structured data with STORED", Proceedings of ACM SIGMOD Conference. Philadelphia, Pennsylvania, May 1998
- [3] G.E. Bake, M.P. Consens, P. Kilpelainen, P.A. Larson, T. Snider, and F.W. Tompa, "Text/Relational Database Management System: Harmonizing SQL and SGML", Proc. Application of Database (ADB 94), Vadstena, Sweden (June 1994), Lecture Notes in Computer Science 819, Springer-Vadstena, pp.267-280
- [4] 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), pp. 12-19, 2001
- [5] 유재수, XML 저장관리시스템 동향 "<http://www.dpc.or.kr/dbworld/document/9909/spec-4.html>", 1999
- [6] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. Dewit and J. F. Naughton, "Relational Database for Querying XML Document : Limitation and Opportunities", Proc. of 25th Int'l conf. on VLDB, Edinburgh, Scotland, UK, 1999
- [7] 박은경, 정채영, 김현주, 배종민, "XML DTD로 부터 관계형 테이블로의 사상 구조 설계", 춘계학술발표회 논문집(b) 28권 1호, 정보과학회, pp 133-135, 2001
- [8] D. Zhong, et al., "Clustering methods for video browsing and annotation", Proc. Storage and retrieval for image and video database IV, pp.239-246, Jan. 1996