

XML 기반 멀티미디어 데이터의 효율적인 구조 정보 검색

임정윤*, 황부현*

*진남대학교 전산학과

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

bhhwang@chonnam.chonnam.ac.kr

Effective Structure Information Retrieval of Multimedia Data Based on XML

Jeong-Youn Lim*, Bu-Hyun Hwang*

*Dept of Computer Science, Chonnam National University

요약

웹의 대중화와 컴퓨터 기술의 발달로 멀티미디어 데이터의 사용이 크게 증가하고 있지만, 사용자가 원하는 멀티미디어 데이터의 일부분과 해당 데이터의 구조정보를 검색하는 기능은 제공되고 있지 않다. 본 논문에서는 사용자가 원하는 내용을 포함하고 있는 멀티미디어 데이터를 검색하고, 그 데이터의 구조정보를 XML 문서 형태로 보여주기 위하여, 객체지향 데이터베이스 시스템을 이용하여 대용량의 멀티미디어 데이터와 해당 데이터를 표현한 XML 문서를 저장하고, 사용자의 질의를 처리하기 위한 효율적인 구조정보 검색 방법을 제안한다.

1. 서론

급속한 컴퓨터 및 통신기술의 발전과 웹 서비스 기반 검색 시스템의 일반화에 힘입어 단순한 텍스트 정보뿐만 아니라 비디오와 같은 멀티미디어 데이터가 보편화되고 그 교류의 양이 크게 증가하고 있다. 그러나, 멀티미디어 데이터는 서술 정보보다 데이터 내용 자체가 내포하는 의미가 더 중요한 경우가 많다. 따라서 멀티미디어 데이터를 검색 질의의 대상으로 하여 사용자가 원하는 결과를 제공하는 구조 검색이 필요하다.[1,2]

그러나 기존에 개발된 시스템들은 멀티미디어 데이터를 데이터베이스에 이진 파일 형태로 저장하고 텍스트 기반 검색 등의 결과를 사용자에게 제공할 때 부가적 정보로서 멀티미디어 데이터를 이용하는 경우가 대부분이고 멀티미디어 데이터 자체를 검색 질의의 대상으로 처리하는 경우가 드물다.

따라서 본 논문에서는 웹 상에서 멀티미디어 데이터와 해당 멀티미디어 데이터를 표현한 XML (eXtensible Markup Language) 문서를 데이터베이스에 저장하고, 질의에 적합한 멀티미디어 데이터를 검색하여 사용자가 검색 결과인 멀티미디어 데이터를 실행시키고 XML 문서형태로 데이터의 구조 정보를 알 수 있도록 하는 시스템을 제안한다.

2. 관련연구

XML은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 마크업 언어이다. XML은 인터넷에서 기존에 사용하던 HTML (HyperText markup Language)의 한계를 극복하고, SGML (Standard Generalized Markup Language)의 복잡함을 단순화함으로써, SGML과 HTML 양쪽 모두와의 상호운용성 및 용이한 구현 환경을 제공한다.[3]

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

웹에서의 자유로운 데이터 교환을 위해 XML은 DTD(Document Type Definition)를 통하여 문서의 구조를 기술하고 있다. 이와 같이 문서의 구조는 사용자가 원하는 대로 정의할 수 있으며, 이러한 구조적 유연성은 모든 분야의 문서들이 XML로 기술될 수 있도록 해준다. 이것은 웹에서 운용되는 모든 데이터가 동일한 형태로 통합, 저장, 처리될 수 있는 기반을 제공한다.[4]

XML 문서를 구성하는 기본 단위는 엘리먼트이다. 따라서 문서 단위를 위주로 한 검색 이외에 엘리먼트 단위의 검색이 가능해야 한다. 또한 엘리먼트 간의 논리적인 포함관계, 형제 관계 및 엘리먼트의 애트리뷰트 값에 대한 질의도 지원되어야 한다. 이와 같이 구조적 특성을 갖는 XML 문서에 대한 검색은 크게 내용 검색, 구조 검색, 그리고 내용과 구조가 혼합된 혼합 검색으로 나눌 수 있으며 부가적으로 애트리뷰트 검색이 있다.

내용 검색은 사용자에게 의해 주어지는 키워드와 관련된 문서나 엘리먼트를 검색하는 것으로 단순 검색, 불리언 검색, 가중치 검색, 와일드 카드 검색, 제한 검색 등으로 지원될 수 있다. 예를 들어, ["대한민국"을 포함하는 문서를 검색하십시오.]와 같은 질의가 이에 해당된다.

구조 검색은 문서의 논리적 구조에 대한 검색으로 부모, 조상, 자식, 자손, 형제에 대한 검색이 될 수 있으며, 또한 이들이 혼합된 검색을 의미한다. 이런 구조 검색은 [<character>가 4명인 영화를 검색하십시오.]와 같은 질의가 될 수 있다.

혼합 검색은 내용과 구조 정보를 이용한 검색, 내용과 애트리뷰트를 이용한 검색, 애트리뷰트와 구조 정보를 이용한 검색 등으로 나눌 수 있다. [<title>에 우산을 포함하는 영화를 검색하십시오.]와 같은 질의가 이에 해당한다.

애트리뷰트 검색은 엘리먼트에 나타날 수 있는 애트리뷰트에 대한 검색으로 [<actor>의 애트리뷰트 sex가 "female"인 영화를 검색하십시오.]와 같은 질의처럼 애트리뷰트 이름과 값을 주고 그에 해당하는 문서나 엘리먼트를 찾는 질의이다.

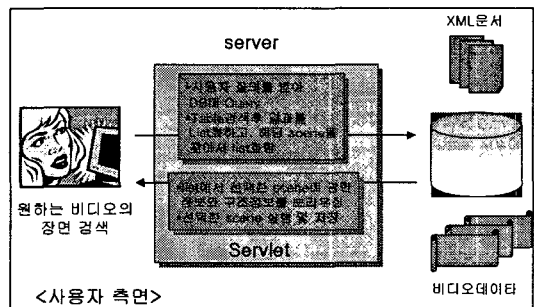
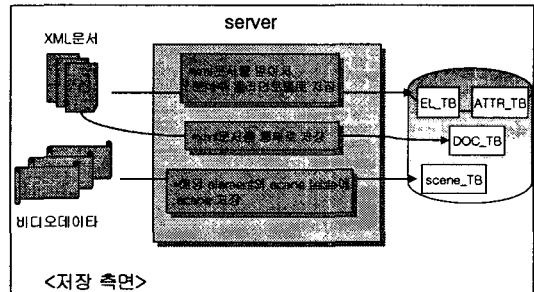
내용 검색, 구조 검색, 애트리뷰트 검색은 각각의 색인 즉, 내용 색인, 구조 색인, 애트리뷰트 색인을 통하여 직접 지원이 되며, 혼합 검색의 경우는 이들 색인의 결합으로 지원 가능하다.[5]

3. 시스템 설계

멀티미디어 자료를 XML로 표현하기 위해서는 XML 문서의 유효성을 검증할 수 있는 DTD가 필요하다. 본 논문에서는 모든 요소를 포함하는 원본 DTD에서 사용자가 원하는 요소를 선택해서 새로운 DTD를 만들어 사용하는 방법을 사용한다.[6] 이미지나 동영상과 같은 멀티미디어 데이터를 나타낼 때는 일반적으로 계층적인 구조를 사용하여 표현한다.

멀티미디어 데이터를 표현함에 있어서 주안점은 XML 데이터를 다른 응용에서도 사용할 수 있도록 쉽고 간결하게 표현하는가와 그 표현이 검색에 효율적으로 사용될 수 있는가에 중점을 두고 표현하였다. 시스템의 구성은 다음과 같다.

멀티미디어 데이터의 구조정보가 XML 문서로 표현되어 데이터베이스에 저장되며, 일반 사용자는 원하는 부분의 멀티미디어 데이터를 웹 상에서 검색하여 실행시키고 해당 데이터의 구조정보를 알아볼 수 있는 시스템이다.



[그림 1]. 시스템 구성도

시스템은 크게 저장모듈과 검색모듈로 구성되어있다. 저장모듈은 멀티미디어 데이터와 그에 대한 XML 문서를 데이터로 입력하여 저장하는 부분으로, 서버는 XML 문서를 DOM(Document Object

Model) 트리 형태로 분석하여 엘리먼트 테이블(EL_TB)과 애트리뷰트 테이블(ATTR_TB)에 저장하고, 멀티미디어 데이터는 문서 테이블에 전체 XML 문서와 함께 저장된다. 검색모듈은 멀티미디어 데이터에서 사용자가 원하는 부분만을 찾기 위해 질의를 하면 서버가 엘리먼트 테이블에서 원하는 질의에 대한 정보를 찾아내고, 장면 테이블(scene_TB)에서 적합한 결과를 찾아 사용자에게 제공한다. 또한 원하는 일부분뿐만 아니라 전체 멀티미디어 데이터와 해당 데이터의 구조정보까지 표현한다.

3.1 멀티미디어 데이터의 구조정보표현 방법

멀티미디어 자료의 XML 표현을 위하여, 멀티미디어의 계층적인 특징을 표현하며, 메타 컨텐츠의 기능을 가지는 DTD를 작성하였다.

```

<!ELEMENT video(meta,sequence*)>
<!ATTLIST video src CDATA #REQUIRED>
<!ATTLIST video start_offset CDATA #REQUIRED>
<!ATTLIST video end_offset CDATA #REQUIRED>
<!ELEMENT meta(title, date, director, casting, producer, production,
file_format, genre)>
<!ATTLIST meta duration CDATA #REQUIRED>
<!ELEMENT title(#PCDATA)>
<!ELEMENT data(#PCDATA)>
<!ELEMENT director(#PCDATA)>
<!ELEMENT casting(casting_name*)>
<!ELEMENT casting_name(#PCDATA)>
<!ELEMENT producer(#PCDATA)>
<!ELEMENT production(#PCDATA)>
<!ELEMENT file_format(#PCDATA)>
<!ELEMENT genre(#PCDATA)>
<!--
<!ELEMENT sequence(se_title,scene*)>
<!ATTLIST sequence id CDATA #REQUIRED>
<!ATTLIST sequence start_offset CDATA #REQUIRED>
<!ATTLIST sequence end_offset CDATA #REQUIRED>
<!ELEMENT se_title(#PCDATA)>
<!ELEMENT scene(sc_title,character, l_data?, i_place?, incident?, music*)>
<!ATTLIST scene id CDATA #REQUIRED>
<!ATTLIST scene start_offset CDATA #REQUIRED>
<!ATTLIST scene end_offset CDATA #REQUIRED>
<!ELEMENT sc_title(#PCDATA)>
<!ELEMENT character(cast_name, character_name*, clothes?, property,
action*)>
<!ELEMENT real_name(#PCDATA)>
<!ELEMENT role_name(#PCDATA)>
<!ELEMENT clothes(clothes_type*, clothes_color*)>
<!ELEMENT clothes_type(#PCDATA)>
<!ELEMENT clothes_color(#PCDATA)>
<!ELEMENT property(#PCDATA)>
<!ELEMENT action(#PCDATA)>
-->
<!ELEMENT l_data(#PCDATA)>
<!ELEMENT i_place(#PCDATA)>
<!ELEMENT incident(#PCDATA)>
<!--
<!ELEMENT music(m_title, m_composer, m_singer)*>
<!ELEMENT m_title(#PCDATA)>
<!ELEMENT m_composer(#PCDATA)>
<!ELEMENT m_singer(#PCDATA)>
-->
    
```

[그림2]. 멀티미디어 데이터 DTD

3.2 저장

기존의 관계형 데이터베이스를 이용하여 저장구조로 4개의 테이블을 구성하였다. 입력 모듈에서 멀티미디어 데이터와 그에 대한 내용정보를 표현하고 있는 XML문서를 서버로 넘기면, 서버의 입력 처리모듈에서는 이 XML문서를 분석하여 엘리먼트나 속성별로 설계되어 있는 4개의 테이블에 저장을 하게 된다. 전체 멀티미디어 파일과 XML문서는 문서 테이블(DOC_TB)에 저장되며, 각 엘리먼트와 엘리먼트의 내용과 내용정보는 엘리먼트 테이블(EL_TB)에 저장된다. 그리고, 각 엘리먼트의 애트리뷰트들은 애트리뷰트테이블(ATTR_TB)에 값과 함께 저장되며, 장면별 멀티미디어 자료와 장면 정보들이 장면 테이블(scene_TB)에 저장된다.

3.3 검색

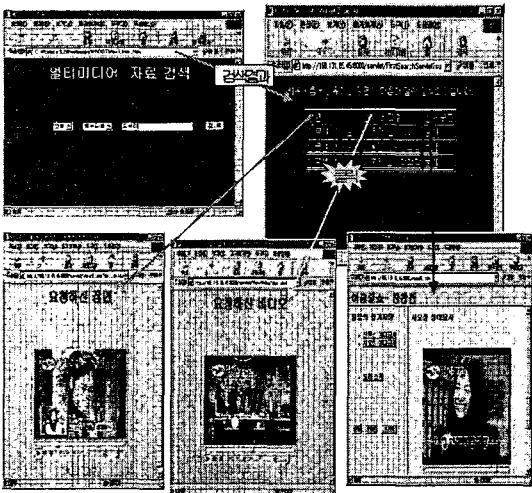
본 논문에서의 검색 모듈은 저장구조가 DOM 기반으로 되어 있기 때문에 기본 연산은 부모와 자식 노드를 찾는 연산과 이전 형제 노드와 다음 형제 노드를 찾는 연산으로 검색을 한다. 그리고 DOM 기반의 저장 구조에서 각 노드의 정보에 노드의 ID를 두어 구조 정보를 검색할 때에 효율을 높였다.

질의 인터페이스는 사용자가 쉽게 XML 문서와 멀티미디어 데이터에 대한 통합 검색을 하고, 검색 결과를 볼 수 있도록 한다.

검색 화면에서 내용 검색 형태로 사용자가 검색하고자 하는 분류와 검색대상 항목을 선택한 후 질의를 입력하면 서버의 검색 모듈이 이 질의를 받아서 내용 검색과 구조 검색을 이용한 혼합 검색 형태로 해당 엘리먼트의 구조를 알아낸 다음 그 구조 정보를 가지고 적합한 멀티미디어 데이터의 제목을 알아낸다. 이 과정에서 애트리뷰트 검색 방법이 사용되기도 한다. 그리고 검색 결과 리스트 출력 화면에 질의에 해당하는 부분 데이터 제목, 전체 데이터 제목, 구조를 출력하고, 각 결과의 제목을 선택하면 결과 모듈이 해당하는 부분 데이터나 전체 멀티미디어 데이터, 또는 데이터의 구조를 보여주게 된다.

[그림3]은 예를 들어, 인물 분류에서 배우이름을 선택한 후 검색어로 "핑클"을 입력했을 경우의 검색된 결과를 보여주는 화면이다. 검색된 결과 리스트에서 장면의 제목을 클릭하면 해당 장면을 보여주고, 전체 비디오의 제목을 클릭하면 해당장면이 포함된 전

체 비디오 화면을 보여주며 전체구조 보기를 클릭 하면 해당 비디오에 대한 전체적인 구조를 볼 수 있다. 맨 윗 프레임에 비디오 파일의 이름이 출력되고, 왼쪽 창은 데이터 파일의 계층적 구조를 보여준다. 왼쪽 창에서 원하는 장면을 선택했을 경우 오른쪽 창에 해당 장면이 보여지고, 그 장면을 저장할 수도 있고, 구조를 XML문서 형태로 볼 수 있다. 또한 해당 장면의 이전 장면이나 다음 장면을 볼 수도 있다.



[그림3]. 검색 결과

4. 결론

본 논문에서는 멀티미디어 데이터와 XML 문서가 저장되어 있는 데이터베이스에서, 사용자가 원하는 부분의 멀티미디어 데이터를 웹 상에서 검색, 실행시키고 해당 데이터의 구조정보를 알아볼 수 있도록 하기 위해 구조 검색, 내용 검색, 혼합 검색, 애트리뷰트 검색 방법을 사용하는 시스템을 제안하였다.

이 방법은 XML을 정보의 표현, 교환 및 중재의 표준으로 사용함으로써 풍부한 구조 정보의 표현과 효과적 검색의 이점을 얻을 수 있고, 이질 정보 환경에의 통합이 용이하다. 또한 사용자가 질의 대상에 대한 의미정보나 부가정보에 대해 사전 지식을 가지고 있는 경우뿐만 아니라 사용자가 멀티미디어 데이터 자체에 대한 정보만을 가지고 있을 경우에도 이용할 수 있다.

향후 보다 다양하고 많은 XML 문서와 멀티미디어 데이터에 대해 각 멀티미디어 데이터의 계층적 구조별로도 검색이 가능하도록 추가적인 연구와 구

현이 필요하다.

참고문헌

[1] Alin Deutsch, M.F. Erandes, D.Suciu, "Storing Semi-structured Data with STORED," Proceedings of ACM SIGMOD Conference, May 1999.
 [2] D.Florescu and D.Kossmann, "Storing and Querying XML Data using as RDBMS," Bulletin of the Technical Committee on Data Engineering Vol. 22, No.3, 1999.
 [3] 이강찬, 손홍, 박기식, "XML 표준화 동향," 정보과학회지, pp.6, 2001.
 [4] 박상원, 정재목, 정태신, 김형주, "XML과 데이터베이스," 정보과학회지, pp.24, 2001.
 [5] 박종관, "XML 문서의 효율적인 구조 검색을 위한 색인 모델," 충북대학교, 2000
 [6] 서승현, 김유성, "멀티미디어 정보의 효율적인 교환을 위한 파생 DTD 관리 시스템의 개발," 한국정보과학회 '2000년 가을 학술발표논문집, 제27권 2호, pp. 15, 2000.