XML 문서에 대한 효율적인 구조 기반 검색을 위한 색인 모델

박종관° 강형일 손충범 유재수
충북대학교 정보통신공학과
{hades00, khi, cbson, yjs}@pretty.chungbuk.ac.kr

An Indexing Model for Efficient Structure-based Retrieval on XML Documents

Chong-Kwan Park° Hyung-Il Kang Chung-Beom Son Jae Soo Yoo
Dept. of Computer & Communication Eng., Chungbuk National University

요 약

XML 문서의 구조검색을 위한 기존 방법들은 특정 엘리먼트의 조상, 자손, 형제에 대한 다양한 구조검색을 효율적으로 지원하지 못한다. 본 논문에서는 XML 문서의 효율적인 관리와 구조검색을 위해 DTD(Document Type Definition)의 논리적 구조를 따르는 XML 문서에 대해 구조정보를 표현하기 위한 방법을 제시한다. 구조정보는 엘리먼트 이름을 식별할 수 있는 EID, 부모와 자식 엘리먼트간의 계층정보를 위한 ETID, 동일한 부모 엘리먼트를 갖는 자식 엘리먼트들의 순서정보를 위한 SORD, 그리고 동일한 부모 엘리먼트를 갖는 자식들 중 동일한 타입의 엘리먼트들에 대한 순서정보를 위한 SSORD로 구성된다. 이런 구조정보를 이용해 빠른 검색을 위한 내용 색인, 구조 색인, 애트리뷰트 색인을 설계한다. 설계된 색인을 통하여 질의를 처리하는 과정을 설명함으로써 다양한 구조적 질의를 효과적으로 처리할 수 있음을 보인다.

1. 서론

최근 인터넷 사용과 정보의 양이 급증하면서 인터넷상의 정보를 보다 효과적으로 사용하고자 하는 연구가 활발히 진행되고 있다. 현재 인터넷상의 대부분 정보는 문서의 구조보다는 표현에 중점을 둔 HTML 문서로 구성되어 있어 특정 응용분야의 구조를 표현하는 문서로는 기능이 부족하다는 단점이 있다[4]. 이에 W3C (World Wide Web Consortium)에서는 차세대 웹 문서의 표준으로 XML (eXtensible Markup Language)이라는 전자문서 메타 언어를 1996년에 제안하였으며 현재까지 그 기능이 계속 확장되고 있는 상태이다[7].

XML은 HTML이 하나의 고정된 DTD를 사용하는 것과는 달리 논리적 구조를 나타내는 여러 DTD를 사용할 수 있다는 점에서 SGML과 같다. 이렇게 DTD의 논리적 구조를 따르는 XML 문서의 구조정보는 XML 문서의 관리나 구조 검색을 효율적으로 수행하는데 이용된다[2].

이러한 논리적 구조를 수용하려는 기존의 구조정보 표현을 위한 연구들은 XML 문서를 가지고 구조정보를 표현하려고 했다. 일부 XML 문서 위주의 구조정보 표현 방법은 문서마다 특정 엘리먼트에 부여된 ID가 다를 수 있다[3,8]. 또한, 특정 엘리먼트에 대한 직접적인 접근이 불가능하거나 조상, 자손, 형제의 관계에 있는 엘리먼트를 접근하기 위해 복잡한 연산을 수행해야 했다[3,4,5,10,11].

이에 본 논문에서는 특정 엘리먼트에 대한 직접적인 접근이 가능하고 엘리먼트 간의 관계를 구하기 위해 복잡한 연산이 필요 없도록 DTD에 나타난 엘리먼트들과 XML 문서의 구조정보를 사용해서 XML 문서를 효율적으로 관리하고 검색할 수 있는 구조정보 표현을 고안하고, 이 정보들을 이용해 효율적인 검색을 위한 색인 구조를 설계한다.

본 논문의 구성은 다음과 같다. 2장에서 XML 문서에 대해 구조정보를 표현하고 검색하는 방법에 대한 기존 연구들을 살펴보고 3장에서 효율적인 구조정보 표현을 제안하고 구조정보 추출기를 설계한다. 4장에서는 추출된 구조정보를 이용해 색인 구조를 설계하고 질의 처리의 예를 보인다. 마지막으로 5장에서는 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

XML문서와 같은 구조화된 문서의 구조정보를 표현하기 위한 방법은 기존에 많은 방법이 제시되었다. 이런 방법들은 일반적으로 문서를 구성하는 노드들에 대해 ID를 부여하게 되는데 부여되는 노드의 ID는 부모의 노드 ID에 자신의 순서정보(부모의 몇 번째 자식)를 붙여서 생성되어 진다. 이런 노드 ID에 의해 계층정보를 나타내고 순차정보를 이용해 노드들간의 순서를 나타낸다. K-ary 완전트리를 이용하는 방법[8]은 엘리먼트의 부모, 자식을 빠르게 찾을 수 있는 장점이 있으나, 조상, 자손, 형제 등의 관계를 알기 위해 어렵거나 복잡한 연산을 요구하게 된다. SCL 모델[10,11]은 엘리먼트들에 대해 깊이를 표현할 수 없어 조상, 형제를 알 수 없다. 추상화에 기반한 방법[6]은 추상화에 의해 색인의 크기를 줄일 수 있으나 추상화되지 않은 구조의 검색을 위해서 문서 전체를 읽어야 하는 오버헤드가 있다. 또한 형제 엘리먼트들의 순서를 알 수 없다.

3. 구조정보 표현

본 논문에서 제안하는 구조정보는 엘리먼트 이름을 식별할 수 있는 ID, 부모와 자식 엘리먼트간의 계층정보, 동일한 부모 엘리먼트를 갖는 자식 엘리먼트(이하 형제 엘리먼트)들의 순서정보, 그리고 동일한 부모 엘리먼트를 갖는 자식들 중 동일한 타입의 엘리먼트들에 대한 순서정보로 표현된다. 이들은 부모 엘리먼트의 정보를 유지한다. 그러므로 기준 엘리먼트로부터 특정 엘리먼트에 대한 계층정보와 순서정보를 간단한 문자열 조작만으로 쉽게 구할 수 있다. 이렇게 구한 순서정보는 각 엘리먼트에 유일하게 할당된 값이기 때문에 직접 특정 엘리먼트를 접근할 수 있다.

3.1 엘리먼트 식별 ID

엘리먼트들 간의 계층 정보를 표현하기 위해서는 엘리먼트 이름에 대해 식별할 수 있는 EID(Element ID)를 부여해야 한다. 이러한 EID는 DTD에서 정의된 각 엘리먼트에 대하여 유일한 ID를 의미한다. 각 엘리먼트에 대해 ID를 부여하는 방식을 통해서 DTD의 논리적 구조를 분석할 때 발생하는 엘리먼트간의 순환 문제를 제거할 수 있다.

EID는 2바이트를 사용하여 표현하는데 각 바이트는 '0' → '9' → 'A' → 'Z' → 'a' → 'z' 순으로 된 62개의 문자를 사용하며 ASCII 코드의 순서를 따르고 있다. 이를 통해 EID는 총 3844개의 엘리먼트를 표현할 수 있다. (그림 1)은 DTD에서 정의된 각 엘리먼트에 대해 EID를 부여하고 EID 맵핑 테이블을

보여 주고 있다.

DTD

```
<!ELEMENT paper (head, section+, reference+)>
<!ELEMENT head (title, author+, abstract)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (name, department)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT department (#PCDATA)>
<!ELEMENT abstract (#PCDATA)>
<!ELEMENT section (num, title, para, subsection*)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT subsection (num, title?, para)>
<!ELEMENT para (#PCDATA | img)*>
<!ELEMENT img EMPTY>
<!ELEMENT reference (#PCDATA)>
```

EID mapping table

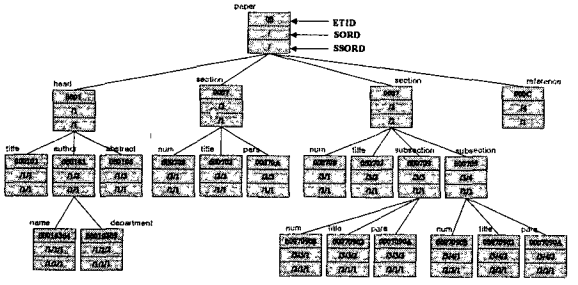
Element Type	EID
paper	00
head	01
title	02
author	03
name	04
department	05
abstract	06
section	07
num	08
subsection	09
para	0A
img	0B
reference	0C

(그림 1) EID 부여 예

3.2 구조정보 표현 방법

DTD의 논리적 구조를 따르는 XML 문서의 구조정보는 엘리먼트를 기반으로 한다. 이러한 XML 문서상의 엘리먼트를 유일하게 구별하면서 엘리먼트들 간의 계층정보를 표현하기 위해서 일반적으로 ID를 부여한다. 본 논문에서는 특정 엘리먼트를 구별하면서 엘리먼트들 간의 계층정보를 표현하기 위해 EID를 사용하여 문서상의 엘리먼트들에 ETID(Element Type ID)를 부여한다. 이렇게 부여된 ETID는 문서의 논리적 구조를 나타내게 된다. 또한 XML 문서에서는 DTD에 나타난 발생지시자에 의한 반복적인 엘리먼트의 사용이 가능하다. 이렇게 반복적으로 사용된 엘리먼트들은 ETID만으로는 구별이 불가능하다. 따라서 본 논문에서는 다양한 구조검색을 지원하기 위해서 두 종류의 순서정보를 제안한다. 형제 엘리먼트들의 발생순서를 나타내는 SORD(Sibling ORDer)와 동일한 타입의 형제 엘리먼트들 간의 순서정보를 나타내는 SSORD(Same Sibling ORDer)가 그것이다.

(그림 2)는 XML 문서의 구조정보를 ETID, SORD, SSORD를 사용해서 트리의 형태로 표현한 예를 보여 준다.



(그림 2) XML 문서의 구조정보 표현

3.2.1 계층 정보

엘리먼트들 간의 계층정보를 표현하기 위해 ETID를 사용하는데 이 ETID는 앞에서 생성된 EID를 이용하여 만들게 된다. 즉, 부모의 ETID에 자신의 EID를 붙여서 자신의 ETID를 생성하게 되는데 부모가 없는 루트 엘리먼트인 경우는 자신의 EID를 ETID로 사용한다. 이 ETID는 문서의 논리적 구조상의 각 엘리먼트 타입에 부여되는 유일한 값이다.

예를 들어 (그림 2)의 구조정보 표현에서 보면 루트 엘리먼트인 paper의 ETID는 "00"이고, section의 ETID는 부모 노드의 ETID인 "00"에 자신의 EID인 "07"을 붙여 "0007"이 된다. 이렇게 생성된 ETID와 해당 엘리먼트 이름과의 사상을 위해 ETID 매핑 테이블이 존재하게 되는데 이 매핑 테이블을 참조하면 효과적인 구조 검색이 가능해 진다. 즉, 엘리먼트간의 조상, 자식이 어떤 형의 엘리먼트인지 쉽게 알 수 있다. 다음의 (그림 3)는 엘리먼트 이름과 ETID의 매핑 테이블을 보여 준다.

엘리먼트들 간의 계층정보를 알아내는 한 예로 (그림 3)의 매핑 테이블에서 subsection 엘리먼트의 부모 엘리먼트는 우선 subsection 엘리먼트의 ETID "000709"에서 뒤의 두 바이트를 잘라내어 부모 엘리먼트의 ETID인 "0007"을 구하고 이 ETID에서 뒤 두 바이트 "07"에 해당하는 엘리먼트를 EID 매핑 테이블에서 찾으면 section이라는 것을 알 수 있다. 결과적으로 XML 문서의 접근이 필요 없이 단지 ETID만을 가지고 엘리먼트들 간의 계층정보를 검색할 수 있다.

Element Type	ETID	Element Type	ETID
paper	00	section	0007
head	0001	num	000708
title	000102, 000702, 00070902	subsection	000709
author	000103	para	00070A, 0007090A
name	00010305	img	NULL
department	00010305	reference	000C
abstract	000106		

(그림 3) ETID Mapping Table

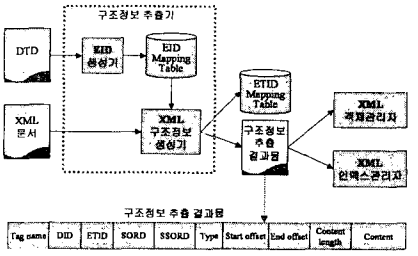
3.2.2 순서정보

XML 문서에서는 동일한 엘리먼트들이 반복적으로 나타날 수 있는데 반복적인 엘리먼트들과 엘리먼트들 간의 순서정보를 표현하기 위해 SORD와 SSORD를 사용한다. 이들의 표현 방법은 UNIX 파일 시스템에서 디렉토리를 표현하는 방법을 사용하며 부모의 순서정보를 상속받는다. 예를 들어, 부모의 SORD가 "/2"(루트 엘리먼트의 2번째 자식)이고 자신의 형제들 중 세 번째로 나타난 엘리먼트라면 이 엘리먼트의 SORD는 "/2/3"이 된다. SSORD 또한 같은 원리로 부여된다.

SORD는 XML 문서에서 나타나는 엘리먼트들을 유일하게 구분할 수 있는 값으로 특정 엘리먼트를 검색하는데 유용하게 이용된다. 또한 형제 엘리먼트들의 순서를 나타내기 때문에 "몇 번째 자식 엘리먼트"라는 질의에 사용될 수 있다. SSORD는 동일한 타입의 형제 엘리먼트들의 순서를 나타내는 것으로 "자식 엘리먼트들 중 몇 번째 특정 엘리먼트"라는 질의 처리에 효과적이다. 따라서 문서의 특정 엘리먼트는 ETID, SORD, SSORD를 이용하면 쉽게 검색할 수 있다.

3.3 구조정보 추출

XML 문서의 구조정보는 EID 생성기와 XML 구조정보 생성기로 이루어지는 XML 구조정보 추출기에 의해서 얻어진다. (그림 4)는 구조정보 추출기의 시스템 구성도를 나타낸다. EID 생성기는 DTD로부터 각각의 엘리먼트들에 대해 엘리먼트 타입을 추출하고 엘리먼트 이름과 EID를 사상시켜주는 매핑 테이블을 구성한다. 그리고 XML 구조정보 생성기는 XML 문서를 분석하면서 EID 매핑 테이블을 참조하여 ETID를 생성해내고 SORD와 SSORD를 생성함으로써 문서의 구조정보를 추출한다.



(그림 4) 구조정보 추출기 시스템 구성도

구조정보 추출기에 의해 추출되는 정보는 위의 그림에 나타난 바와 같이 여러 필드들을 갖는 텍스트 파일이다. 각 필드들은 각각의 의미를 갖는데 Tag name은 엘리먼트의 이름 혹은 애트리뷰트의 이름이며 DID는 문서를 구분하기 위해 부여된 고유번호이다. ETID, SORD, SSORD는 이미 설명한 바와 같고 Type은 추출된 결과물이 엘리먼트인지 애트리뷰트인지를 나타낸다. Start offset와 End offset은 각각 XML 문서 화일에서 해당 엘리먼트의 시작위치와 끝위치를 나타낸다. Content는 해당 엘리먼트에 속하는 실제 내용이거나 애트리뷰트의 값이며 Content length는 Content 필드에 있는 값의 길이를 나타낸다. 이렇게 문서에서 추출된 구조정보는 XML 문서 관리 모듈과 색인을 구성하기 위해 XML 색인 관리 모듈에 의해 사용되어진다.

4. 색인 구성 및 질의 처리

XML 문서에 대한 검색은 내용에 대한 검색 이외에 구조 검색이 지원되어야 하며 애트리뷰트 검색 또한 지원되어야 한다. 구조 검색은 문서의 논리적인 구조에 기반한 질의로서, 엘리먼트들의 계층간의 관계, 같은 계층내에서의 관계, 계층 전체 관

계 등을 고려해야 한다. 계층간의 관계는 부모, 자식, 조상, 자손 관계가 있으며, 같은 계층내의 간계는 형제 일리먼트간의 순서가 있다. 그리고 계층 전체 관계는 선후관계가 있다. 애트리뷰트 검색은 엘리먼트에 나타날 수 있는 속성에 대한 질의로 애트리뷰트 이름과 값을 주고 해당 문서나 엘리먼트를 찾는 질의이다. 본 논문에서 제안하는 구조정보 표현 방법을 이용하여 모든 질의들을 처리할 수 있으며, 이런 질의들을 효율적으로 처리하기 위한 색인 구조를 설계한다. 이 색인 구조는 다음의 내용 색인, 구조 색인, 애트리뷰트 색인의 3개로 구성된다.

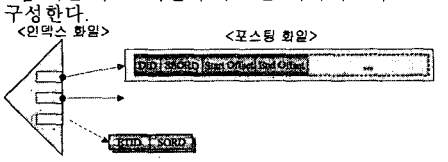
4.1 내용 색인

내용 색인은 내용 검색을 지원하는 색인로서 XML 문서로부터 추출된 색인어로 구성된 색인 화일과 색인어가 출현한 문서와 엘리먼트의 정보를 나타내는 포스팅 화일로 구성된다. 포스팅 화일은 DID, ETID, SORD로 구성된다. DID는 문서 단위 검색의 경우 사용되며, ETID, SORD는 엘리먼트 단위의 검색을 지원하기 위해 사용된다. (그림 5)는 내용 색인의 구조를 나타낸다.

(그림 5) 내용 색인의 구조

4.2 구조 색인

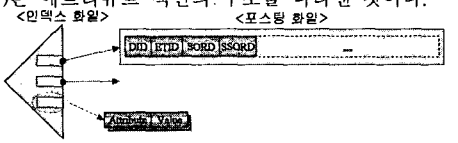
구조 색인은 구조 검색을 지원하는 색인로서 문서의 논리적인 구조 정보를 손실없이 표현한다. 이를 통해 엘리먼트 간의 계층관계 검색, 엘리먼트 간의 순서 관계인 형제 검색을 지원한다. (그림 6)은 구조 색인의 구조를 나타내는데 ETID로 색인 화일을 구성한다.



(그림 6) 구조 색인의 구조

4.3 애트리뷰트 색인

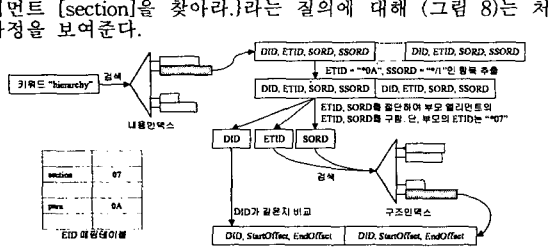
애트리뷰트 색인은 애트리뷰트 검색을 지원하는 색인로서 추출된 구조 정보 중 애트리뷰트와 값을 색인으로 구성하고 이와 관련된 DID, ETID, SORD들을 포스팅 형 화일로 구성한다. 다음 (그림 7)은 애트리뷰트 색인의 구조를 나타낸 것이다.



(그림 7) 애트리뷰트 색인의 구조

4.4 질의 처리

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. 한 예로 {"hierarchy"를 포함하는 첫 번째 [para]의 부모 엘리먼트 [section]을 찾아라.}라는 질의에 대해 (그림 8)는 처리 과정을 보여준다.



(그림 8) 질의 처리 과정

5. 결론

본 논문에서는 XML 문서를 위한 질의를 분석하고, XML 문서의 효과적인 관리와 검색을 위해 문서의 구조정보를 표현하는 방법을 제시하고 이에 따라 검색을 위한 색인을 설계하였다. 구조정보를 표현하기 위한 요소는 DTD에 나타나는 각각의 엘리먼트에 대해 유일한 값인 EID, 엘리먼트들 간의 계층정보를 나타내는 ETID, XML 문서에서 형제 엘리먼트의 순서정보인 SORD, 그리고 같은 엘리먼트 형에 대한 순서정보인 SSORD로 구성된다. 이들 정보들은 효율적인 검색을 위한 색인 구조를 설계하는데 사용된다. 제안된 색인 구조는 내용 검색을 지원하는 내용 색인, 구조 검색을 지원하기 위한 구조 색인, 애트리뷰트 검색을 지원하는 애트리뷰트 색인로 구성되며, 혼합 검색은 이들 색인의 결합으로 차리하게 된다. 이와 같은 구조정보 표현과 색인을 통해 특정 엘리먼트에 대한 직접적인 접근이 가능하며, 다양한 구조적 질의를 효과적으로 처리할 수 있어 보다 효율적이고 빠른 검색을 지원할 수 있게 되었고 구조화된 문서관리 시스템 개발을 위한 기틀을 마련하였다.

향후 연구로써 다량의 XML 문서에 대해 본 논문에서 제안한 구조 정보 표현 방법을 이용한 색인 구조와 기존의 구조 검색을 위한 방법들과의 성능평가를 수행하는 것이다.

참고 문헌

[1] 김용훈, "다양한 구조 검색을 지원하는 XML 문서 검색기의 설계 및 구현", 충남대학교 석사학위논문, 1998

[2] 민영수외 5인, "XML 문서를 위한 구조정보 추출기의 설계 및 구현", 한국정보과학회 '99 가을 학술발표논문집(I), 한국정보과학회, pp, 81~83, 1999

[3] 손정한, 이희주, 장재우, 심부성, 주종철, "구조화된 문서를 위한 정보검색시스템의 설계 및 구현", '98 동계 데이터베이스 학술대회 논문집 제 14권 1호, pp. 102-106, 1998

[4] 연제원, 조정수, 이강찬, 이규철, "XML 문서 구조검색을 위한 저장 시스템 설계", 한국정보과학회 학술 발표 논문집 (B), 제 26 권 1호, pp. 3-5 1999

[5] Brian Lowe, Justin Zobel, Ron Sacks-Davis "A Formal Model for Databases of Structured Text", Proceedings of the Fourth International Conference on Database Systems for Advanced Applications (DASFAA '95), pp. 449-456,1995

[6] Chow, J.H., Cheng, J., Chang, D., Xu, J., "Index Design for Structured Documents Based on Abstraction", Proceedings of the 6th International Conference on Database Systems for Advanced Applications, pp. 89 -96, 1999

[7] Extensible Markup Language(XML) 1.0, "http://www.w3.org/TR/1998/REC-xml-19980210"

[8] Lee, Y.K., Yoo, S.J., Yoon, K.R and Berra, P.B., "Index Structures for Structured Documents", Proc. Digital Library 96, pp. 91-99, 1996

[9] R. Sacks-Davis, T. Arnold-Moore, and J. Zobel, "Database systems for structured documents", Proc. The International Symposium on Advanced Database Technologies and Their Integration (ADTI '94), Nara, Japan, pp. 277-283, 1994

[10] Tuong Dao, Ron Sacks-Davis, James A. Thom, "An Indexing Scheme for Structured Documents and its Implementation", Proceedings of the Fifth International Conference on Database Systems for Advanced Applications (DASFAA '97), pp. 125-134, 1997

[11] Tuong Dao, "An Indexing Model for Structured Documents to Support Queries on Content, Structure and Attributes", Proceedings of ADL'98, pp. 88-97, 1998