제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제1호 (2004. 9)

XML 문서의 효율적인 검색을 위한 확장된 ETID 구조정보

신주현*, 최준호*, 김판구**
* 조선대학교 전자계산학과
** 조선대학교 컴퓨터공학과
e-mail:jhshinkr@unitel.co.kr

A Extensible the ETID Structured Information for Efficient Retrieval of XML Documents

Ju-Hyun Shin*, Jun-Ho Choi*, Pan-Koo Kim**
*Dept of Computer Science, Chosun University
**Dept of Computer Engineering, Chosun University

요 약

XML문서가 웹상에서의 정보의 표현 및 교환의 표준 포맷으로 선택 되면서 XML데이터에 대한 저장 기법과 검색 효율을 높일 수 있는 방법들이 연구 되고 있다. 본 논문에서는 구조화 정보검색을 위해 기존에 연구되어진 ETID 구조정보 표현의 한계를 개선하여 엘리먼트 및 에트리뷰트의 특정부분에 대한 직접적인 검색을 하기 위해 DTD에 상관없이 형제 엘리먼트간의 확장된 구조정보를 표현하여 효율적인 검색을 수행할 수 있는 방법을 제안한다.

1. 서론

XML(Extensible Markup Language)은 인터넷상에서 데이터 교환을 목적으로 모든 문서 및 응용 프로그램에 대한 범용의 마크업을 정의하는 방법을 표준화한 메타언어이다. XML은 간단하고 일관된 방법으로 인터넷상에서 데이터 전송을 용이하게 하는 문서 정의 형식으로 1996년 W3C(World Wide Web Consortium)에서 제안하였다. HTML은 주로 문서의 정보보다는 외형(appearance)을 표현할 수 있는 기능을 제공하고 있는 반면, 단순하면서도 문서의 구조 및 의미를 손쉽게 표현할 수 있고 새로운 태그와 속성을 새로이 추가할 수 있는 기능을 제공하기도 한다. 이로 인해 다양한 분야에서 XML이 사용되어 지면서 저장기법에 대한 연구와 검색효율을 높이는 방법에 대한 연구가 활발히 진행 되고 있다. 이미 많은 상용데이터베이스가 XML데이터를 저장하고 처리할 수 있는 방법을 채택했음에도 복잡한 구조의 XML데이터가 방대해 지고 변경되어지는 상황에서의 최적화가 되어있지 않은 상황이다. 따라서 검색효율을 높일 수 있는 많은 방법이 연구되어져야 할 실정이다.

본 논문에서는 XML데이터의 검색효율을 높일 수 있는 방안을 제시하기 위해 XML데이터에 대한 구조정보 표현에 대해 분석한 다음 구조검색을 위해 XML문서에서 구조정보 색인을 위하여 ETID(Element Type ID)와 LETID(Leveled Element Type ID)를 보완한 EETID(Extensible Element Type ID)를 제안한다. 제2장에서는 XML데이터에 대한 구조정보를 표현하는 방법에 대해 분석하고, 제3장에서는 효율적 구조검색 시스템을 위한 EETID(Extensible Element Type ID)구조정보와 색인 구조를 제시한다. 제4장에서는 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

XML문서에 대한 구조정보 검색을 위해 기존의 연구들에서는 다양한 방법들을 제시하였다. 구조검색기능으로는 엘리먼트별 분할 검색, 엘리먼트 구조검색, 애트리뷰트 검색등이 지원되어야 한다. 현재 구조정보를 표현하기 위한 모델로는 SCL(Simple Concordance List)모델, K-ary 완전 트리 모델, ETID(Element Type ID) 모델 등이 있다.

1) SCL모델

SCL모델은 호주 RMIT에서 제시하였으며, 정의된 문서 계층과 정의된 마크업 스키마로부터 독립을 제공한다. 이 모델은 텍스트 간격들을 다루며 문서 구조에 대한 질의를 지원하기 위해 계층적인 관계보다 포함 관계를 사용한다. SCL구조는 색인어를 포함하는 엘리먼트에 대해서는 알 수 있으나 엘리먼트들에 대해 트리 내의 깊이를 표현할 수 없다는 단점을 가진다.

2) K-ary 완전트리 모델

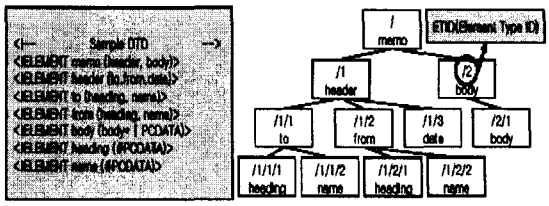
K-ary 완전트리 방식은 문서를 K-ary 완전 트리에 매핑 하여 구조검색을 지원하는 방법이다. K-ary 완전 트리의 특성상 간단한 연산식에 의하여 부모노드와 자식노드를 찾을 수 있다. i번째 노드의 부모노드를 구하기 위하여 Parent(i) = (i-2)/k+1을 이용하고, i번째 노드의 j번째 자식노드를 찾기 위해 Child(i,j)=K(i-1)+j+1을 이용하여 간단하게 찾을 수 있다. 단점으로는 트리가 깊어질수록 노드의 변화가 커지고 노드의 데이터가 많아지며, 실제로 존재하지 않는 노드가 증가하여 검색 시 실제로 존재하는 노드인지 아닌지를 판단하는 부분이 필요하다.

3) ETID(Element Type ID) 모델

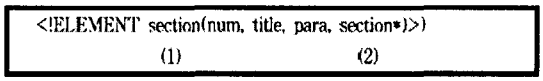
특정 엘리먼트를 구별하면서 엘리먼트간의 계층정보를 표현할 수 있는 ETID는 DTD의 논리적 구조를 분석할 수 있는 각 엘리먼트 타입에 부여되는 유일한 값이다. [그림 1]에서와 같이 DTD의 구조를 분석하여 ETID를 모두 부여한다. "from"의 ETID는 부모 노드 "header"의 "/1"을 상속받아 "1/2"가 된다. 그러나, DTD는 단순히 XML문서 내부에서 엘리먼트 들의 나열방식에 대하여 기술해 놓은 것이다. 도서와 같은 XML문서에서는 "chapter" 노드가 1번 이상 나타날 수 있으며, "*" 형태로 표시된 노드의 경우 ETID는 가지지만 XML문서에서는 한번도 출현하지 않을 수도 있다.

[그림 1]에서는 실제 엘리먼트가 가상 엘리먼트를 이용하여 DTD 트리를 구성하고 ETID를 부여한다. 엘리먼트 트리에 따라 ETID를 부여하면서 가장 문제가 되는 순환(recursion)이 발생하는데, [그림 2]와 같이 자기 자신을 포함하는 형태가 되었을 경우 순환이 일어난다.

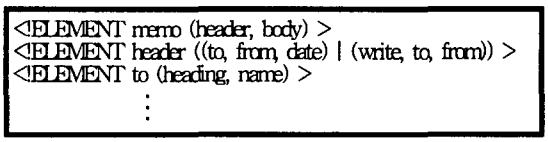
실제적으로 XML문서에서는 순환이 일어나고 있지 않지만 DTD를 이용해 트리를 만들어 ETID를 부여하는 방식에서는 순환이 문제가 된다.



[그림 1] DTD를 이용한 ETID부여



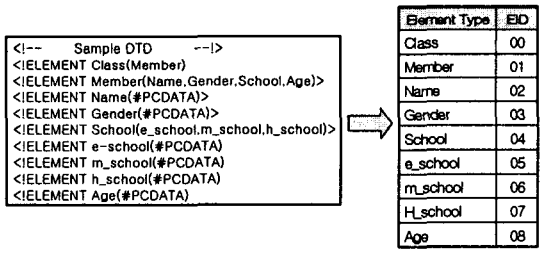
[그림 2] DTD 순환의 예



[그림 3] 노드의 선택이 있는 DTD

[그림 3]에서 header 내부의 노드 표현이 선택적으로 되어있을 때 트리를 구성하기에 많은 문제점이 생긴다. 이러한 문제점을 해결하기 위하여 ETID를 추출하기 위한 방법을 변형시킨 방법이 있다.

먼저 DTD에 EID(Element ID)를 추출하고 XML 문서를 순회하면서 ETID를 부여하는 방식이다.



[그림 4] DTD에서 ETID 추출

[그림 4]와 같이 DTD에서 모든 엘리먼트에 EID를 할당한 후 XML 문서를 순회하면서 부모노드의 EID를 상속받아 ETID를 할당하는 방식으로 계층정보를 나타낸다. class의 EID가 "00" 이며, class 노드의 하위노드인 member는 "01"이다. XML문서에서 member의 ETID는 부모노드의 "00"과 자신의 EID "01"을 합하여 "/00/01"이 된다.[2,3] 이 방법은 DTD에서 계층정보는 만들어 내지 않으며 단순히 EID 만을 추출한다. 따라서 순환이 발생하지 않으며, DTD를 분석하여 계층구조로 표현하지 않아도 된다. 단점으로는 ETID 정보에 의하여 엘리먼트간의 부모-자식 관계는 파악할 수 있으나 형제 노드에

대한 정보는 알지 못한다. 이를 위하여 형제 노드들 간의 순서정보(SORD : sibling order)와 동일 타입의 형제 엘리먼트들 간의 순서정보(SSORD : same sibling order)를 추출하여 사용한다[3]. 형제 노드들 간의 순서정보(SORD)는 동일 부모를 갖는 엘리먼트들의 출현 순서이며, 동일 부모를 갖는 엘리먼트들 중 동일한 형(TYPE)간의 순서는 SSORD로 표현된다.

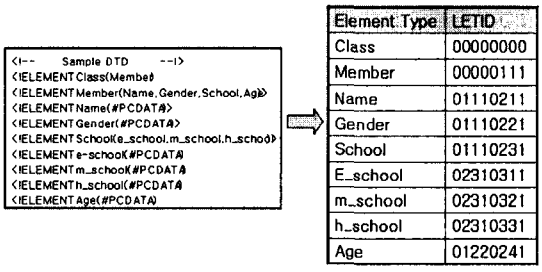
4) LETID(Element Type ID) 모델

[4]에서 구조정보를 추출하기 위하여 LETID(Leveled Element Type ID)를 제안하였다. ETID와는 다르게 LETID의 구조정보는 8바이트 고정크기로 모든 정보를 표현하며, 상위 4바이트는 부모노드의 정보이며, 하위 4바이트는 자신의 정보이다. <표 1>은 각 바이트들이 의미하는 정보를 나타낸 것이다.

자리	의미	자리	의미
1,2	부모노드의 깊이	5,6	현재 노드의 깊이
3,4	3: 부모 노드의 SORD 4: 부모 노드의 SSORD	7,8	7: 현재 노드의 SORD 8: 현재 노드의 SSORD

<표 1> LETID 정보

[그림 5]는 DTD에서 각각의 엘리먼트 순서에 의해 LETID값을 표현했다. 하지만, ETID의 경우처럼 DTD에서는 순환이 일어나 계층정보를 추출하기 어려우며, 형제노드 순서의 경우 DTD에서는 알아낼 수가 없다.



[그림 5] DTD에서 LETID 부여 방법

[그림 6]에서 XML1과 XML2 모두 DTD에 유효한(Valid) 문서이다. XML1에서는 Sub section노드가 존재하지 않으므로 SORD와 SSORD의 값을 지정할 수 없다. XML2에서는 첫 번째 section의 SORD는 "4"이며, SSORD는 "1"이다. 두 번째 section의 SORD는 "5", SSORD는 "2"이다. 이와 같은 정보는 DTD에서 추출이 불가능하다.

DTD
<!ELEMENT section(num, title, para, section*)>)

XML 1

```
<section>
<num> </num>
<title>.....</title>
<para>.....</para>
</section>
```

XML 2

```
<section>
<num> ..... </num>
<title> ..... </title>
<para> ..... </para>
<section>.. </section>
<section>.. </section>
</section>
```

[그림 6] DTD에 의해 표현 될 수 있는 XML문서의 예

LETID방식은 ETID에서처럼 EID가 필요 없으므로 DTD에서 반드시 추출하지 않아도 된다. LETID의 경우 타 연구에서는 DTD를 이용하나, 본 연구에서는 DTD에서 정보를 뽑아낼 수 없음과 DTD가 불필요함을 발견하였다. 이를 효과적으로 개선할 수 있는 방법으로 본 논문에서는 ETID와 LETID 방식을 확장한 EETID(Extensible Element Type ID)를 사용하여 XML문서에서 구조정보를 추출할 수 있는 방안을 제안한다.

3. 확장된 ETID 구조정보 표현과 색인구조

3.1 구조정보 표현

본 논문에서 사용되는 구조정보는 <표 2>과 같다. 각 필드는 기본적으로 1바이트를 차지한다. LETID의 경우 형제노드가 1바이트, 동일 형제노드가 1바이트로 구성되어 형제노드의 개수 제한이 일어난다. '0~9', 'A~Z', 'a~z' 와 같이 표현 할 경우 62개의 노드 표현에 대한 한계가 있다. XML 문서의 경우 분할된 엘리먼트가 기하급수적으로 증가하게 된다. 따라서, 본 논문에서 제한하는 EETID의 경우 62개를 초과하는 노드를 표현할 경우에 해당되는 노드를 2바이트로 증가할 수 있고, 그 경우에는 62*62개의 노드까지 처리가 가능하다.

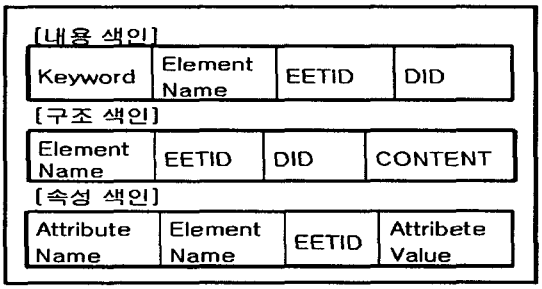
자식노드수 Count(CN)	속성 수 ACount(AN)	부모노드 SORD(PS)	부모노드 SSORD(PSS)	현재노드깊이(CD)	현재노드 SORD(CS)	현재노드 SSORD(CSS)

<표 2> EETID 구조정보

[그림 5]의 2번째 Member노드를 <표 2>과 같이 나타내면 "/4/0/0/0/1/2/2"로 표현된다. EETID 정보에서는 부모노드의 깊이(PD)는 'CD(현재노드 깊이) -1 = PD'와 같은 방법으로 쉽게 추출할 수 있기 때문에 저장하지 않는다. 확장된 조건 검색을 위해 자식노드의 개수와 속성개수를 가진다. 이와같은 구조정보는 내용색인, 구조색인, 속성색인에서 기본정보로 사용된다.

3.2 색인 구조

효율적 검색을 위한 색인구조로 내용 색인, 구조 색인, 속성색인으로 나누어 인덱스 데이터베이스 내에 각각 독립적인 테이블로 관리한다.



[그림 7] 색인 정보

1) 내용 색인

내용 색인 정보는 내용검색을 수행하기 위한 것으로 키워드를 중심으로 구성된다. 내용 검색은 논문과 같은 자료인 경우 "구조정보 를 포함하는 엘리먼트를 찾아라"와 같은 질의를 처리하기 위하여 사용된다.

2) 구조 색인

구조 색인 정보는 엘리먼트 중심으로 구성되며, DID, EETID, Content로 구성되어진다. 예를 들어 "이름이 title인 엘리먼트의 2번째 자식노드"와 같은 질의를 처리하기 위한 정보이다. 본 연구에서 제시하는 색인 구조는 상위노드 검색, 하위노드 검색, 형제 노드검색, 같은 타입의 형제 노드 검색, 특정 개수의 자식노드를 가지는 노드 검색, 특정 개수의 속성을 가지는 노드 검색, 노드 value 검색등이 가능해진다.

3) 속성 색인

속성색인은 속성 이름이 중심이 되며, 속성이 속한 엘리먼트와의 매핑을 위해 DID, EETID를 가지고 있다. 속성 질의의 경우 엘리먼트의 이름과 함께 질의하는 것이 대부분이며 내용질의와 같이 단순하여 처리가 간편하다. "이름이 Jonson인 엘리먼트의 속성이름이 name인 것"과 같은 질의를 처리하기 위한 정보이다.

3.3 구조정보 확장성과 검색 효율

구조정보 표현을 위해 EETID만으로 가능하며, 노드의 개수가 62개를 넘지 않을 경우 7바이트로 표현이 가능하다. 분할 엘리먼트의 수가 62개를 넘을 경우 2바이트로 확장하여 표현이 가능하다. 각각의 색인정보는 테이블에 저장되고, ETID와 같은 방식으로 각각 다른 컬럼에 정보가 저장되어, AND와 같은 다중 연산자를 사용할 수 있다. 따라서 특정 자식 노드 검색을 위한 횟수가 적어짐에 따라 검색효율을 높일 수 있다.

5. 결론 및 향후 연구 방향

XML 문서의 우수성으로 인하여 많은 인터넷 자료들이 XML문서로 표현되고, 많은 분야에서 활용되어지고 있다. 본 논문에서는 특정 엘리먼트나 속성에 대한 검색을 위해 기존 연구보다 더 확장된 방법이 제시 되었다. 특히, 기존의 방식들이 요구하는 DTD정보 없이 XML문서의 구조검색이 가능한 예를 보였다.

향후 연구로는 설계내용을 중심으로 타 검색시스템에 비해 성능이 우수함에 대한 실험과 평가를 한다. 또한 최근 많은 연구가 이루어지는 온톨로지 부분에 대해서 데이터베이스와의 연계방법에 대한 연구를 수행할 계획이다.

참고문헌

[1] T.Bray, J. Paoli, and C. M. SperbergMcQueen, "Extensible Markup Language(XML) 1.0", W3C Working Draft, 1998.

[2] 조윤기, 조정길, 이병렬, 구연설, "XML 문서에 포함된 구조정보의 표현과 검색", 정보처리학회논문지 D, 제8-D권, 제 4호, 2001.

[3] 박종관, 손충범, 강형일, 유재수, 이병엽, "XML 문서의 효율적인 구조 검색을 위한 색인 모델", 정보처리학회 논문지 D, 제8-D권, 제 5호, 2001.

[4] 김성완, 정헌석, 이재호, 임해철, "XML문서에서의 엘리먼트 타입을 이용한 구조적 검색 기법의 설계", 한국정보과학회, 2003.

[5] B. Chang, J. Kesselman, and R. Rahman, "Document Object Model(DOM) Level 3",W3C Working Draft, 2003.

[6] Philip J harding, and Quanzhong Li, Bongki Moon "XISS/R:XML Indexing and Storage System Using RDBMS", In Proceedings of the 29th VLDB Conf.,, 2003.

[7] 객체-관계형 데이터베이스를 이용한 XML 문서 저장 기법, "이월영, 용환승", 정보처리학회논문지 D, 제2호, 2004

[8] 천윤우, 홍동권, "관계형 모델에서 XML 변경과 전문 검색을 지원하기 위한 역 인덱스 구축 기법, 정보처리학회논문지 D, 제11-D권, 제 3호, 2004

[9] R..Bourret, XML-DBMS : Middleware for Transferring Data betwwneen XML Documents and Relational Databases, available at http:L// www.rpbourret.com / xmldbms/Incidents on the Internet," Ph.D. Thesis Carnegie Mellon University, 1998.