문서중심 XML 문서를 위한 데이터 모델

김연희*, 김성완**, 신판섭*, 이재호***, 임해철*
*홍익대학교 컴퓨터공학과
**삼육의명대학 전산정보과
***인천교육대학교 컴퓨터교육과
{kyh, swkim, psshin, lim}@cs.hongik.ac.kr, jhlee@mail.inue.ac.kr

Data Model for Document-Centric XML Document

YounHee Kim*, SungWan Kim**, PanSeop Shin*, Jaeho Lee***, HaeChull Lim*
*Dept. of Computer Engineering, Hong Ik University
**Dept. of Computer Information, Sahmyook College
***Dept. of Computer Education, Inchon National University of Education

요 약

웹 상의 데이터 표현 및 교환의 새로운 표준으로 인식되어 점차 그 교류의 양이 증가하고 있는 XML 문서를 효과적으로 저장, 접근 및 검색하기 위한 기법에 대한 연구가 많았으나, 기존의 연구들은 데이터중심 문서의 특성이 두드러지는 XML 문서를 대상으로 하는 것이 대부분이였다. 그러나 효과적인 XML 문서의 저장 및 검색을 위해서는 XML 문서의 실제 사용 목적이나 그 특성에 따라 XML 문서를 분류하여 각 특성에 맞는 저장, 접근 및 검색 기법을 개발하고 이를 통합한 XML 문서 저장 시스템의 개발이 요구된다. 따라서 본 논문에서는 통합 시스템 개발에서, 인간 이해 중심의 문서적 특성을 가지는 문서중심 문서를 위한 데이터 모델을 제안한다. 제안된 데이터 모델은 루트 노드가 존재하는 방향성과 순서가 있는 그래프 형태를 지원하며, XML 문서의 주요 구성 요소를 지원하는 여러 타입의 노드와 다양한 노드 사이의 관계를 표현하는 링크로 구성되어 XML 문서가 가지는 의미와 구조적 특징이 잘 표현되도록 하였다. 또한 모델링 후 손실되는 정보가 거의 없기 때문에, 다시 XML 문서로 변환하면 원래 XML 문서 그대로 복원되는 장점이 있어 문서중심 문서의 저장 및 검색을 위한 전용 XML 저장 시스템에 적합한 데이터 모델이다.

1. 서 론

XML이 웹 상에서 데이터와 구조적 정보를 표현하고 교환하는 새로운 표준으로 인식되면서, 많은 응용에서 XML 기반의 플랫폼을 개발하려는 움직임이 활발하다. 이러한 변화에 따라 웹 상의 많은 XML 문서를 효과적으로 저장, 접근 및 검색할 수 있도록 하는 기법들에 대한 연구의 필요성이 대두되었다[1,2].

기존의 많은 연구들은 XML 문서의 사용 목적이나 특성에 따른 분류를 고려하지 않고, 단순히 일반적인 XML 문서를 대상으로 하는 연구가 대부분이였다. 그러나 보다 효율적인 XML 문서의 저장 및 검색을 위해서는 XML 문서 자체의 특성에 따라 XML 문서를 분류하여 각 특성에 맞는 저장 및 검색 기법을 개발하고 이를 모두 지원할 수 있는 통합 XML 문서 저장, 검색 시스템에 대한 연구가 필요하다[3]. 특히 기존의 연구들은 데이터중심 문서의 특성이 많이 나타나는 XML 문서를 대상으로 하여 그러한 XML 문서를 기존 데이터베이스에 저장하기 위한 모델링 및 매핑 방법 등에 대한 연구가 대부분이고, 문서중심 문서를 대상으로 하는 연구는 드물었다[4].

따라서 본 논문에서는 통합 XML 문서 저장, 검색 시스템 개발에 있어 문서중심의 문서를 그 특성에 맞게 효과적으로 저장, 접근 및 검색할 수 있도록 하는 데이터 모델을 제안한다. 문서중심 문서는 데이터중심 문서와 달리 덜 정형화되어 있고, 혼합 엘리먼트 내 형제 엘리먼트 사이의 순서와 문서 내의 순서 정보가 중요하고, 인간 이해 중심의 문서적 특성 때문에 모델링 후 원문 복원 능력이 요구되는데 제안된 데이터 모델은 이러한 문서중심 문서의 특징을 그대로 모델링할 수 있도록 설계되었다.

제안된 데이터 모델은 가상의 루트를 가지고 방향성과 순서가 존재하는 그래프로 표현될 수 있고, 자신이 가지는 의미를 설명하는 레이블을 가진 노드와 노드간의 여러 관계를 표현하기 위한 링크로 구성되어 있다. 이 데이터 모델은 엘리먼트 외에 XML 문서를 구성하는 주요 요소를 세분화하여 노드 타입을 구분하고 각 노드간의 관계를 일반적인 엘리먼트 간의 중첩 관계 외에 엘리먼트와 애트리뷰트 사이의 관계, 내부 및 외부 노드로의 참조 관계를 구분하여 XML 문서가 가지는 의미는 물론 구조적인 정보가 모델링 후에 손실되지 않도록 하였다. 또한 정보의 손실없이 모델링되기 때문에 모델링 후 다시 XML 문서로 변환했을 때, 원래의 XML 문서로 그대로 복원될 수 있어서 문서중심 문서의 저장 및 검색에 적합한 전용 XML 저장 시스템이 갖추어야 하는 라운드 트리핑의 조건을 만족한다.

본 논문의 구성은 다음과 같다. 관련 연구에서는 기존 연구에서 제안된 XML 문서를 위한 데이터 모델을 소개하고 그 제한점을 지적한다. 데이터 모델에서는 본 논문에서 제시한 문서중심 문서를 위한 데이터 모델을 설명하고 모델링 예에서는 제안된 데이터 모델을 이용한 실제 모델링 예를 제시하고 결론을 맺는다.

2. 관련 연구

XML 문서는 사용 목적 및 문서 자체의 특성에 따라 데이터중심 문서(data-centric)와 문서중심 문서(document-centric)로 나눌 수 있다[3].

본 연구는 한국과학재단 특정기초연구과제 (과제번호 : 98-0102-09-01-3)의 지원을 받았음

데이터중심 문서는 주로 데이터 교환의 포맷으로 사용되며 어느 정도 정형화된 구조를 가지고 있고, 세분화된 데이터 정보 표현을 위주로 하기 때문에 데이터 외에 형제 엘리먼트의 순서와 같은 XML 문서 내의 다른 정보는 중요하지 않다. 문서중심의 문서는 주로 비정형화된 구조이고, 인간이 이해할 수 있는 문서적 특징을 가지기 때문에 데이터 외에도 혼합요소내의 형제 엘리먼트의 순서와 같이 XML 문서가 가지고 있는 다양한 정보가 중요하다.

기존에 개발된 많은 XML 문서의 저장 및 검색을 위한 연구들은 대부분 데이터중심 문서의 특성을 가지는 XML 문서를 대상으로 하는 경우가 많아 문서가 가지는 모든 의미나 정보를 전부 표현하기보다는 데이터 정보의 모델링을 위주로 하기 때문에, 모델링 후 손실되는 정보가 많고 모델링 후 다시 XML 문서로 변환했을 때 원래의 문서와는 다른 형태의 문서로 복원되는 경우가 많다. 따라서 이러한 모델링 방법은 인간 이해의 문서적 특성을 가지는 문서중심 문서에는 적합하지 않다.

XML과 semi-structured 데이터와의 유사함을 고려하여 기존의 semi-structured 데이터를 위한 OEM 데이터 모델이 XML을 지원하도록 확장한 연구는 XML 문서의 순서 정보와 태그 정보에 대한 표현이 가능하고, 트리 형태 및 그래프 형태를 모두 지원하는 유연성있는 데이터 모델을 제시하였다[5,6,7]. 그러나 XML 문서의 구성 요소 중 엘리먼트에 대한 고려가 대부분이고 애트리뷰트, 주석, PI, 엔터티 등의 의미를 제대로 지원하지 못하고, XLink와 XPointer 개념을 이용한 외부 XML 문서 전체나 문서 부분의 참조 관계를 명확히 표현하지 못하는 단점이 있다.

OEM이 XML을 완전하게 지원하지 못하는 단점을 해결하기 위해 새롭게 제안된 XOM(eXtensible Object Model)의 경우에도 링크 개념을 도입하여, XLink와 XPointer의 개념을 이용한 외부 XML 문서에 대한 참조 관계를 추가하고 애트리뷰트를 엘리먼트와 구분하여 처리하지만 명확하지 않고, OEM 확장의 경우와 마찬가지로 주석, PI, 엔터티 등의 정보에 대한 고려가 없고, XLink를 이용한 외부 참조 관계에서도 확장 링크 타입에 대한 고려가 부족하여 XML의 특성을 모두 반영하지 못하는 단점이 있다[8]. 따라서 이러한 데이터 모델의 경우 문서중심 문서를 모델링 하는데 있어서, 원래의 XML 문서가 가지는 데이터 이외의 정보를 그대로 표현하는데 한계가 있어, 모델링 후 손실되는 정보가 많고 따라서 원문 복원이 제대로 이루어지지 않는다는 단점이 있다.

3. 데이터 모델 정의

본 논문에서 제안한 데이터 모델은 XML 문서들 가운데 인간 이해 중심의 문서적 특성을 가지는 문서중심 문서를 모델링하기 위한 것으로, XML 문서 자체가 가지는 원래의 의미와 구조적 특징을 반영할 수 있도록 설계되었다. 제안된 데이터 모델은 크게 노드와 링크로 구성되며, 문서 전체를 의미하는 최상위 루트가 존재하고 노드 레이블된 방향성과 순서가 있는 그래프 형태로 표현될 수 있다. 순서 정보는 그래프에서 왼쪽 노드부터 오른쪽 노드 순서로 표현된다. 노드는 한 문서 내의 유일한 값으로 구별되고, 노드 자신의 의미를 설명하는 이름, 즉 레이블을 가지고 있으며 XML 문서를 구성하는 주요 요소에 따라 5개의 타입으로 구분된다. 링크는 노드와 노드간의 다양한 관계를 표현하기 위한 것으로 노드간의 5가지 서로 다른 관계를 표현한다.

3.1 노드

본 논문에서 제안한 데이터 모델에서 노드는 XML 문서의 주요 구성 요소인 엘리먼트, 애트리뷰트, 데이터, 엔터티 등을 표현하기 위한 것으로 엘리먼트 노드, 애트리뷰트 노드, 데이터 노드, 엔터티 노드, 참조 노드 타입으로 구분된다. 노드는 한 문서 내에서 유일하게 구별되는 NID 값과 노드 자체의 의미를 설명하는 이름, 즉 레이블을 갖는다. 레이블은 보통 엘리먼트의 태그 이름, 애트리뷰트의 이름, 엔터티 이름을 이용하므로 노드가 레이블을 가지는 것은 XML 문서를 자연스럽게 모델링하는 방법이며, 이 데이터 모델에 기반한 질의어 작성 시 노드에 대한 질의를 쉽게 정의할 수 있게 한다. 각 타입별 노드에 대한 설명은 다음과 같다.

엘리먼트 노드는 또 다른 엘리먼트, 애트리뷰트, PCDATA 혹은 CDATA 섹션으로 정의된 텍스트를 자식으로 가지는 엘리먼트를 표현하기 위한 것으로 엘리먼트의 태그 정보에 해당하는 레이블, 모든 자식 노드에 대한 리스트를 가지고 있다.

애트리뷰트 노드는 한 엘리먼트의 특징을 설명하는 애트리뷰트를 위한 것으로 애트리뷰트의 이름을 레이블로 가지고, CDATA, ID, IDREF, IDREFS, NMTOKEN 등과 같은 애트리뷰트의 타입, 실제 애트리뷰트 값을 표현한 데이터 노드 리스트를 포함한다.

데이터 노드는 리프 노드에 해당하는 것으로, 실제 XML 문서 내의 데이터를 표현하기 위한 것이다. 데이터 노드는 엘리먼트의 태그로 둘러싸인 PCDATA 혹은 CDATA 섹션의 텍스트, 애트리뷰트의 값, PI와 COMMENT의 내용, 엔터티의 실제 대체값을 모두 표현할 수 있고, 이들을 PCDATA, CDATA, ATTRIBUTE, PI, COMMENT, ENTITY 데이터 타입으로 구별한다. 이러한 데이터 노드는 실제 데이터 값과 정수, 스트링, 실수 등의 기본 타입 혹은 집합과 리스트에 대한 타입 정보를 포함한다. 데이터 값의 타입으로 기본적인 타입 외에 리스트 타입과 집합 타입을 지정함으로써 다양한 타입의 애트리뷰트에 대한 지원이 가능하다.

엔터티 노드는 XML 문서 내에서 사용되고 있는 엔터티 정보를 표현하기 위한 것으로, 엔터티의 이름, 내부 엔터티와 외부 엔터티를 구별하는 타입, 실제 대체될 데이터에 대한 정보를 포함하고 있다.

참조 노드는 외부 참조 관계에서 참조되는 외부의 XML 문서 전체 혹은 부분을 표현하기 위한 노드이다. XLink/XPointer의 개념을 이용하여 외부 XML 문서나 문서의 한 부분을 참조할 때 참조되는 외부 XML 문서나 문서의 한 부분을 한 문서의 데이터 모델 내에 직접 표현하기 어렵기 때문에, 이들의 참조 정보를 표현하는 노드를 따로 생성한다. 참조 노드는 참조될 외부 문서의 이름이나 위치 정보와 참조 관계의 의미를 설명하는 레이블을 포함한다

노드 사이의 순서 정보, 특히 혼합 엘리먼트 내에서 형제 엘리먼트 사이의 순서 정보는 데이터중심 XML 문서와는 달리 문서중심 XML 문서에서는 중요하다. 하나의 부모 엘리먼트를 가지는 형제 엘리먼트 사이의 순서는 그래프에서 왼쪽 엘리먼트 노드부터 오른쪽 엘리먼트 노드 순서로 표현된다. 필요에 따라 하나의 엘리먼트의 애트리뷰트들 간의 순서 또한 이와 같은 방법을 적용할 수 있다.

3.2 링크

노드간의 다양한 관계를 표현하는 링크는 링크 타입과 타겟 노드에 대한 정보를 포함한다. 링크 타입은 엘리먼트와 자식 엘리먼트 간의 관계를 나타내는 링크, 엘리먼트와 태그 사이의 텍스트 정보의 관계를 나타내는 링크, 엘리먼트와 그 특징을 설명하는 애트리뷰트와의 관계를 나타내는 링크, 엘리먼트가 문서 내부적으로 참조하는 다른 엘리먼트와의 관계를 나타내는 링크, 엘리먼트와 외부 문서 전체 혹은 문서의 특정 부분 사이의 외부 참조 관계를 나타내는 링크로 나뉜다. 엘리먼트와 자식 엘리먼트, 엘리먼트와 참조 엘리먼트, 그리고 엘리먼트와 애트리뷰트 간의 모든 관계를 명확히 분류함으로써 모델링 후 발생할 수 있는 XML 문서의 구조적 정보 손실이 생기지 않는다.

엘리먼트의 내부 참조 관계는 IDREF/IDREFS 타입의 애트리뷰트를 가지는 엘리먼트가 이 애트리뷰트 값과 일치하는 ID 타입의 애트리뷰트 값을 가지는 엘리먼트를 참조하는 것을 의미한다. 내부 참조 관계는 엘리먼트와 자식 엘리먼트의 관계와 유사하지만, 링크 타입으로 그 의미를 구별한다.

엘리먼트의 외부 참조 관계는 XLink/XPointer의 개념을 고려한 것으로 외부 문서 전체 및 문서 부분을 참조할 때, 참조될 문서를 위한 참조 노드를 생성하여 그 참조 관계를 표현한다. 여러 로케이터 엘리먼트 간의 참조, 양방향 참조 등을 지원함으로써 일반적인 단순 링크뿐만 아니라 확장 링크의 표현도 가능하다.

4. 모델링 예제

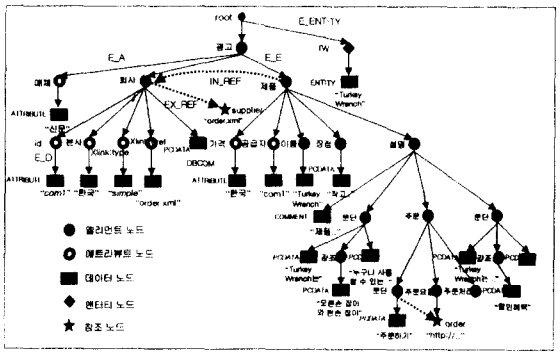
본 논문에서 제안한 데이터 모델은 문서중심의 XML 문서를 모델링

하기 위한 것이다. 다양한 타입의 노드와 링크로 구성된 데이터 모델은 문서 전체를 의미하는 가상의 노드를 루트로 하고, 노드 자체에 레이블된 방향성과 순서가 존재하는 그래프로 표현 가능하다.

<그림 1>는 일반적인 문서중심 문서 특성을 가지는 XML문서의 예를 보여준다. 이 XML 문서는 "Turkey Wrench"라는 공구 제품을 광고하기 위해 제품에 대해 설명하는 문서이다. 문서에는 엘리먼트, 애트리뷰트, 엔터티, 주석 등의 다양한 XML 구성 요소가 사용되고 내부 참조 관계와 외부 참조 관계가 표현되었다.

```
<광고 매체="신문">
    <회사 id="com1" 본사="한국" xlink:type="simple" xlink:href="order.xml" >DBCOM</회사>
    <제품 가격="13.03" 공급자="com1">
        <이름>&TW;</이름>
        <장점>작고 강력한 멍키스내퍼</장점>
        <설명>
        <!--제품 자체에 특성을 설명하는 부분-->
            <문단>&TW;는 <강조>오른손 집이와 왼손 잡이</강조> 누구나 사용할 수 있는 강철
                스내퍼입니다. 사용법이 간단하여 손쉽게 사용할 수 있습니다.</문단>
            <주문 xlink:type="extended">
                <문단 xlink:type="resource" xlink:role="para">온라인 주문하기</문단>
                <주문요청 xlink:type="locator" xlink:href="                "
                        xlink:role="order"/>
                <주문처리 xlink:type="arc" xlink:from="para" xlink:to="order"
                        xlink:show="replace" xlink:role="GetOrder"/>
            </주문>
            <문단>&TW;은 손쉽게 휴대가 가능합니다. 지금 주문하시면
                <강조>할인혜택</강조>이 있습니다</문단>
        </설명>
    </제품>
</광고>
```

<그림 2>는 <그림 1>의 XML 문서를 본 논문에서 제안한 데이터 모델로 모델링한 결과를 표현한 것이다. 모델 표현에서 내부 참조 관계, 외부 참조 관계 링크는 점선으로 표현하여 다른 링크와 구별하였고, 다른 링크 관계는 모델 내에서 추측이 가능하므로 따로 구분하지 않았다. 모델 내에서 노드간의 순서는 하나의 부모 노드를 갖는 왼쪽 자식 노드부터 오른쪽 자식 노드 순서로 표현된다. 특히 "문단" 엘리먼트와 같이 여러 개의 PCDATA 텍스트와 다른 엘리먼트를 동시에 자식으로 가지는 경우 그들 사이의 순서 정보는 문서 중심 XML 문서에서 중요하므로 모델링 후 그 순서가 그대로 유지되어야 한다. <그림 2>와 같이 본 논문에서 제안한 데이터 모델로 XML문서를 모델링하면 XML 문서의 의미와 구조적 특징을 손실없이 그대로 표현할 수 있다.



5. 결 론

본 논문에서는 XML 특성에 맞는 효과적인 저장 및 검색을 제공하기 위한 통합 XML 문서 저장 시스템의 개발에서, 데이터뿐만 아니라 XML 문서가 가지는 다양한 의미와 구조적 정보를 그대로 반영해야 하는 문서중심 문서의 특성에 적합한 데이터 모델을 제안하였다. 제안한 데이터 모델은 문서 자체를 나타내는 가상의 루트를 가지고 방향성과 순서가 존재하는 그래프 형태로 표현이 가능하고, 자신이 가지는 의미를 설명하는 레이블을 가지는 노드와 노드 간의 여러 관계를 표현하기 위한 링크로 구성되어 있다.

제안된 데이터 모델은 XML 문서의 주요 구성 요소를 세분화하여 노드 타입으로 구분하고, 일반적인 엘리먼트 중첩 관계는 물론 다양한 노드 간의 관계를 표현하기 위한 링크 타입을 정의하여 모델링 후 문서가 가지는 의미와 구조적 정보에 대한 손실을 줄이고, 모델링 후 다시 XML 문서로 변환하는 경우 라운드 트리핑, 즉 원래 XML 문서로 복원될 수 있다.

참고 문헌

[1] Jennifer Widom, "Data Management for XML", IEEE Data Engineering Bulletin, Special issue on XML, Vol.22, No. 3, pp. 44-52, September 1999

[2] Stefano Ceri, Piero fraternali and Stefano Paraboschi, "XML: Current Developments and Future Challenges for the Database Community", Proc. of the 7 th Int. Conf. on Extending Database Technology (EDBT), pp. 3-17, March, 2000

[3] Ronald Bourret, "XML and Databases"

[4] JeongEun Kim, PanSeop Shin, Jaeho Lee, HaeChull Lim, "A Database Approach for Modeling and querying XML Documents", ITC-CSCC 2000 Proceedings Vol. 2, pp. 703-706, 2000

[5] Yannis Papakonstantinou, Hector Garcia-Molina and Jennifer Widom, "Object Exchange Across Heterogeneous Information Sources", Proc. of the 11th IEEE International Conference on Data Engineering, March, 1995

[6] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom, "Lore: A Database Management System for Semistructured Data", ACM SIGMOD Record, Vol. 26, No. 3, pp. 54-66, September, 1997

[7] Roy Goldman, Jason McHugh, and Jennifer Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language", Proc. of the 2nd International Workshop on the Web and Databases (WebDB '99), pp. 25-30, Philadelphia, Pennsylvania, June 1999.

[8] Dell Zhang, Yisheng Dong, "A Data Model and Algebra for the Web", Proc. of the 10th International Workshop on Database & Expert Systems Applications, pp. 711-714, September, 1999

[9] W3C, "XML Information Set", http://www.w3c.org/TR/xml-infoset/