XML 문서를 위한 효율적인 색인 구조의 설계

안선하°, 박경환
동아대학교 컴퓨터공학과

A Design of an Efficient Index Organization for XML Documents

Seon-Ha Ahn, Kyung-Hwan Park
Dept. of Computer Eng., Univ. of Dong-A

요 약

XML은 메타언어로써 구조적인 문서를 생성할 수 있게 하며 이러한 구조적인 문서는 내용 기반 검색, 질의등의 처리가 가능하다는 것이다. XML 문서의 논리적 구조를 표현하는 방법은 사용자 정의 문서타입(DTD)과 XML Schema의 두 가지 방식이 있으며, XML Schema는 사용자 정의 문서타입(DTD)에 비해 다양한 데이터 타입, namespace, Open-ended 데이터 모델 등을 지원하여 문서의 구조 정의를 훨씬 더 유연하고 명시적이게 하는 장점이 있다.

본 논문에서는 XML 문서 검색을 위해 XML Schema에 기반하여 내용 검색과 구조 검색을 효율적으로 지원하는 인덱스 구조를 제안한다. 요소들의 정의에 따르는 계층 관계를 표현하기 위한 구조 정보와 XML 문서 인스턴스에서 나타나는 각 요소들의 순서 정보를 요소의 ID로 사용함으로써 임의의 요소를 효율적으로 접근할 수 있게 한다.

1. 서론

XML(Extensible Markup Language)[1]은 메타언어로써 구조적인 문서를 생성할 수 있게 하며 이러한 구조적인 문서는 정확한 검색, 질의등의 처리가 가능하고, 자동 생성할 수 있게 한다. XML문서의 구조 정보를 이용하면 문서 전체에 대한 기본적인 단어의 검색뿐만 아니라 문서에 포함된 임의의 요소들이 포함하고 있는 내용에 대한 구조 검색이 가능하다.

XML 문서의 논리적 구조를 표현하는 방법은 사용자 정의 문서타입(Document Type Definition; DTD)과 XML Schema의 두 가지 방식이 있으며, XML Schema[2,3]는 사용자 정의 문서타입(DTD)에 비해 다양한 데이터 타입, namespace, open-ended 데이터 모델 등을 지원하여 문서의 구조 정의를 훨씬 더 유연하고 명시적이게 하는 장점이 있다.

이와 관련한 연구로는 XML 문서의 저장 및 관리를 위해 기존의 데이터베이스를 이용하거나 전용 시스템이 개발이 시작되었다[4-5]. 여기서, 효과적인 색인 구조는 검색 시간을 결정하는 중요한 요소이며, 효율적인 검색과 동시에 동적으로 확장 가능한 색인 구조에 대한 연구가 계속 진행되고 있다[6-8].

본 논문에서는 XML 문서 검색을 위해 XML Schema에 기반하여 내용 검색과 구조 검색을 효율적으로 지원하는 인덱스 구조를 제안한다. 요소들의 정의에 따르는 요소 타입 정보와 계층 관계를 표현하기 위한 요소 문맥 정보, XML 문서 인스턴스에서 나타나는 각 요소들의 순서 정보를 함께 사용하여 각 요소들을 유일하게 식별하는 ID로 정의함으로써 임의의 요소를 효율적으로 접근할 수 있게 한다.

본 논문은 2장에서 XML Schema를 소개하고 3장에서는 XML Schema에 기반한 인덱스 구조를 제안한다. 4장에서 결론이 따른다.

```
<?xml version="1.0"?>
<Schema name="AddressBookSchema"
    xmlns="urn:schemas-microsoft-com:xml-data"
    xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="AddressBook"
               content="eltOnly">
    <element type="entry"
             minOccurs="1" maxOccurs="*"/>
  </ElementType>
  <ElementType name="entry"
               content="eltOnly" order="seq">
    <element type="name"/>
    <element type="phone"
             minOccurs="1" maxOccurs="*"/>
    <element type="address"/>
  </ElementType>
  <ElementType name="name"
               content="textOnly" dt:type="string"/>
  <ElementType name="phone"
               content="eltOnly" order="seq">
    <element type="area"/>
    <element type="phoneNumber"/>
  </ElementType>
  <ElementType name="area"
               content="textOnly" dt:type="string"/>
  <ElementType name="phoneNumber"
               content="textOnly" dt:type="string"/>
  <ElementType name="address"
               content="eltOnly" order="seq">
    <element type="city"/>
    <element type="street"/>
  </ElementType>
  <ElementType name="city"
               content="textOnly" dt:type="string"/>
  <ElementType name="street"
               content="textOnly" dt:type="string"/>
</Schema>
```

<그림 1. Address Book을 위한 XML Schema>

2. XML Schema

XML 문서내에서 사용되는 각 요소들에 대한 규칙과 서로간의 유기적인 관계를 나타내기 위한 방법으로 사용자 정의 문서타입(DTD)과 XML Schema가 있다. 사용자 정의 문서타입(DTD)는 SGML(Standard Generalized Markup Language)에서 사용되었던 것으로 SGML의 서브셋인 XML에서도 역시 사용 가능하다. 그러나, XML Schema가 사용자 정의 문서타입(DTD)이 가진 몇 가지 제약사항을 해결하므로써 더 유연하고 명시적으로 정의할 수 있다.

우선 사용자 정의 문서타입(DTD)은 새로운 요소들과 속성들을 정의하기 위한 오픈 컨텐트 모델(Open-content Model)을 지원하지 않는다. 또한 문자 데이터 외에 데이터 타입에 대한 지원이 없으며 네임스페이스(namespace)에 대해서도 유연하지 못하다. 마지막으로, 열거되는 값들에 대해서도 허용하지 않는다.

```
<?xml version="1.0"?>
<AddressBook
          xmlns="x-schema:AddressBookSchema.xml">
  <entry>
    <name>Kim Mi-Young</name>
    <phone>
      <area>051</area>
      <phoneNumber>551-3379</phoneNumber>
    </phone>
    <phone>
      <area>0553</area>
      <phoneNumber>279-3380</phoneNumber>
    </phone>
    <address>
      <city>Pusan</city>
      <street>Hadan 840</street>
    </address>
  </entry>
  <entry>
    <name>Kim Young-Ju</name>
    <phone>
      <area>051</area>
      <phoneNumber>206-4153</phoneNumber>
    </phone>
    <phone>
      <area>051</area>
      <phoneNumber>248-4153</phoneNumber>
    </phone>
    <address>
      <city>Pusan</city>
      <street>Dadae 250</street>
    </address>
  </entry>
</AddressBook>
```

<그림 2. 그림 1의 XML Schema에 의한 문서 인스턴스>

이에 반해 XML Schema는 Microsoft사에서 제시한 XML-Data에 기반한 것으로 사용자 정의 문서타입(DTD)이 가진 제약 사항들을 보다 잘 다룰 수 있다. XML Schema는 XML에 기반하여 고안되었기 때문에 저작을 위한 문법이나 처리를 위한 도구들이 따로 필요하지 않다. 또, 정수, 실수, 논리, 날짜, 시간등의 다양한 데이터 타입을 지원한다. 뿐만 아니라, 오픈 컨텐트 모델(Open-content Model)을 지원하여 요소의 정의를 상속·확장할 수 있게한다. 이외에도, 속성 그룹을 제공하여 논리적으로 속성들을 연결할 수 있으며, 네임스페이스(namespce)를 지원하므로 같은 이름의 충돌문제를 해결하고 Schema의 재사용성을 높인다.

이와 같은 점들로 인해 사용자 정의 문서타입(DTD)으로 구조화된 문서보다 XML Schema를 기반으로 정의한 문서가 훨씬 더 효율적임을 알 수 있다.

위의 그림 1은 XML Schema의 예를 보여주고 그림 2는 그에 기반하여 작성된 XML 문서 인스턴스를 보여준다.

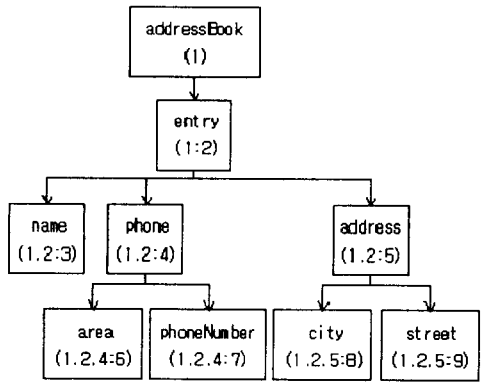
3. XML 문서를 위한 색인 구조

기존의 텍스트 기반 검색 시스템은 문서의 내용으로부터 단어 단위로 추출하여 색인으로 사용하고, 사용자 질의어에 나타난 단어와 비교 검색하여 찾은 문서들이 결과로 제시된다. 그러나, XML은 메타언어로써 구조적인 문서를 생성할 수 있게 하며 이러한 구조 정보를 이용하면 문서 전체에 대한 기본적인 단어의 검색뿐만 아니라 문서에 포함된 임의의 요소들이 포함하고 있는 내용에 대한 구조 검색이 가능하다.

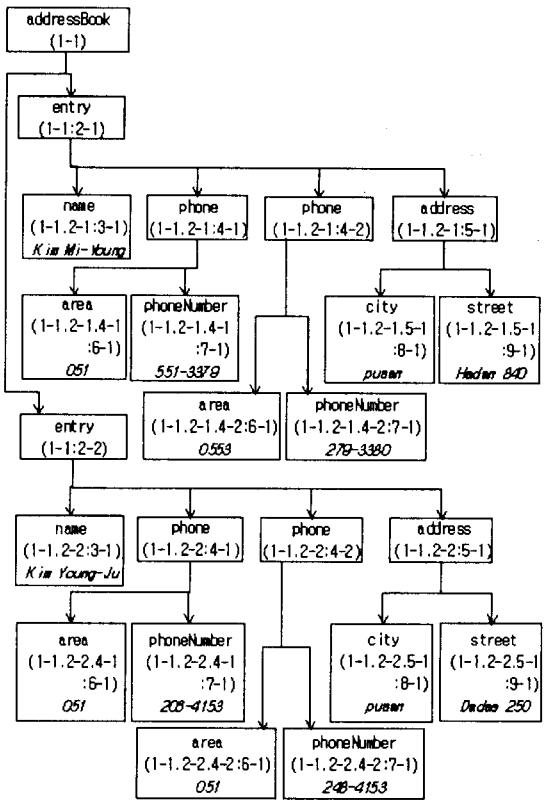
본 논문에서는 XML 문서의 검색을 위해 XML Schema에 기반하여 내용 검색과 구조 검색을 효율적으로 지원하는 인덱스 구조를 제안한다. 요소들의 정의에 따르는 요소 타입 정보와 계층 관계를 표현하기 위한 요소 문맥 정보, XML 문서 인스턴스에서 나타나는 각 요소들의 순서 정보를 함께 사용하여 각 요소들을 유일하게 식별하는 ID로 정의함으로써 임의의 요소를 효율적으로 접근할 수 있게 한다.

XML Schema로부터 정의되는 색인에서는 요소 타입 식별자로서 ETID와 요소 문맥 정보로서의 ECID가 있다. 요소 타입 식별자 ETID는 새로운 요소가 정의될 때마다 새로 부여한다. 요소 문맥 식별자 ECID는 요소간의 계층 관계를 명시하고, 같은 이름의 요소라 할지라도 다른 요소의 부 요소로 사용된 경우를 구별한다. 이와 같이 부여된 ECID와 ETID를 함께 명시하여 XML Schema에서 정의된 모든 요소들을 구별하는 EID를 구성한다. 그림 1에서 예시한 addressBookSchema의 phone의 ETID는 (4)이다. ECID는 상위 모든 계층의 요소들의 ETID를 나열하여 (1.2)가 된다. 따라서, addressBook/entry/phone을 구별하기 위한 EID는 (1.2:4)가 된다. 그림 1에서 보인 XML Schema에 대한 색인 구조를 그림 3에서 보이고 있다.

XML Schema에 기반한 문서 인스턴스의 색인 구조에서는 위에 정의한 EID에 일련 번호(SN)를 삽입하여 반복적으로 나타나는 요소들의 순서 정보를 유지한다. ETID가 동일한 요소들이 나타나는 순으로 SN을 부여한다. ETID에 SN을 삽입하면 (ETID-SN)가 되고, 요소의 타입과 순서를 알 수 있다. 이에따라, 계층 구조를 위한 ECID에도 SN이 삽입되어지므로, (ECID-SN:ETID-SN)의 형태로 요소 인스턴스 식별자(EIID)를 구성한다. 그림 2의 XML 문서에서 2번째 entry 요소의 첫 번째 phone 요소의



<그림 3. 그림 1의 XML Schema에 대한 색인 구조>



<그림 4. 그림 2의 XML 문서에 대한 색인 구조>

area 요소의 EIID는 (1-1.2-2.4-1:6-1)이 된다. 만일, phone을 하위 요소로 포함하는 모든 요소를 찾는다면, phone의 ECID (1:2)와 임의의 SN이 결합된 EIID를 찾으면 (1-1:2-1), (1-1:2-2)가 된다. 그림 2의 XML 문서에 대한 색인 구조를 그림 4에서 보인다.

본 논문에서 제안한 XML Schema를 기반 색인 구조는 XML Schema로부터 구한 요소들의 타입과 문맥 정보와 문서 인스턴스에 나타난 순서 정보를 사용하여 임의의 요소를 검색하고 직접 접근할 수 있도록 한다.

5. 결론

XML은 메타언어로써 구조적인 문서를 생성할 수 있게 하며 이러한 구조적인 문서는 정확한 검색, 질의등의 처리가 가능하고, 자동 생성할 수 있게 한다. XML문서의 구조 정보를 이용하면 문서 전체에 대한 기본적인 단어의 검색뿐만 아니라 문서에 포함된 임의의 요소들이 포함하고 있는 내용에 대한 구조 검색이 가능하다.

XML 문서의 논리적 구조를 표현하는 방법은 사용자 정의 문서타입(Document Type Definition; DTD)과 XML Schema의 두 가지 방식이 있으며, XML Schema는 사용자 정의 문서타입(DTD)에 비해 다양한 데이터 타입, namespace, open-ended 데이터 모델 등을 지원하여 문서의 구조 정의를 훨씬 더 유연하고 명시적이게 하는 장점이 있다.

본 논문에서는 XML 문서 검색을 위해 XML Schema에 기반하여 내용 검색과 구조 검색을 효율적으로 지원하는 인덱스 구조를 제안하였다. 요소들의 정의에 따르는 요소 타입 식별자(ETID)와 계층 관계를 표현하기 위한 요소 문맥 식별자(ECID)를 사용하여 요소 식별자(EID)를 구성하였고, XML 문서 인스턴스에서 나타나는 각 요소들의 순서 정보를 함께 사용하여 모든 요소들을 유일하게 식별하는 요소 인스턴스 식별자(EIID)로 정의함으로써 임의의 요소를 효율적으로 검색하고 접근할 수 있게 한다.

앞으로, 제안한 색인 구조의 구현및 성능평가와 이러한 구조에 잘 적용될 수 있는 질의 언어에 대한 연구가 수반되어야 할 것이다.

[참고문헌]

[1] W3C, Extensible Markup Language(XML) 1.0, W3C Recommendation, http://www.w3.org/TR/REC-xml, 1998.

[2] W3C, XML Schema Part 0: Primer W3C Working Draft, 2000. http://www.w3.org/TR/xmlschema-0/ 1999.

[3] Microsoft, Introduction to Schemas, http://msdn.microsoft.com/xml/c-frame.htm?/xml/xmlguide/schema-example-dtd.asp

[4]Daniela Florescu, Donald Kossmann, "A Performance Evaluation of Alternative Mapping Schemes for Strong XML Data in a Relational Database", INRIA, manuscript, may, 1999.

[5] Troy D. Milner and Jason Lowder, "The SCCOBS Context Based Search Engine", Proceedings of XML\SGML Asia Pacific 1999.

[6] 장재우, 이희주, 손정한, 심부성, 주종철, "SGML 정보검색을 위한 인덱스 관리자의 설계 및 구현", 정보과학회논문지(C), Vol. 5, No. 2, 1999.

[7] Dongwook Shin, Hyuncheol Jang, and Honglan Jin, "BUS: An Effective Indexing and Retrieval S초등 in Structured Documents," Proceedings of the third ACM Conference, 1998.

[8] 이계준, 신동욱, 권택근, "XML 문서의 검색을 위한 효율적인 색인 기법과 질의 언어(TQL) 설계, 26th Kiss Conference, 1999.