문서 단편화 기법을 이용한 XML 변환기의 설계 및 구현

정창후° 최윤수 주원균 진두석 김광영 이민호 서정현
한국과학기술정보연구원 정보시스템개발실
{chjeong°, armian, joo, dsjin, kykim, cokeman, jerry}@kisti.re.kr

Development of an XML Converter using Document Fragmentation Method

Chang-Hoo Jeong°, Yun-Soo Choi, Won-Kyun Joo, Du-Seok Jin,
Kwang-Young Kim, Min-Ho Lee, Jeong-Hyeon Seo
Korea Institute of Science and Technology Information

요 약

최근 다양한 응용 분야에서 점차 증가하고 있는 XML 문서에 대한 효과적 검색을 위해서 많은 검색 시스템들이 제안되고 있다. 그러나 이러한 검색 시스템은 XML 문서의 구조적 특성을 명확하게 알지 못하거나 질의어 작성에 익숙하지 못한 사용자에게 XML 문서를 검색하는데 많은 어려움을 주고 있다. 이러한 문제를 해결하기 위해 본 논문은 복잡한 계층의 XML 문서를 의미 있는 엘리먼트를 중심으로 계층을 단순화시켜서 검색에 이용할 수 있도록 도와주는 XML 문서 변환기를 제안한다. XML 문서 변환기는 XML 문서의 부모-자식 관계, 형제 관계 등의 계층 정보를 유지하면서 문서를 단편화 시켜주는 도구이다. XML 문서 변환기와 더불어 이것을 이용하여 구현된 XML 문서 검색 시스템의 계층적 출력 인터페이스에 대하여 함께 설명하도록 한다.

1. 서 론

XML 검색은 일반적으로 XML 문서의 계층적인 구조에 기반을 두고 있다. 이러한 계층적인 구조를 검색하기 위해서 XPath[1]와 같은 패스 형식을 사용하는데 이러한 질의는 사용자가 XML 문서상의 계층 구조를 정확하게 알고 있어야만 사용이 가능하다. 또한 XPath로 표현된 XML 질의를 관계 형 데이터베이스 시스템에서 처리하기 위해서는 개체와 개체 사이의 계층적인 구조를 검색하기 위한 빠른 조인 알고리즘이 필요하다. 그러나 본 논문에서 제시하는 변환 규칙의 색인 엘리먼트 지정 방법을 사용할 경우 관리자가 전 처리 과정으로 XML 문서의 계층 관계를 하나의 검색 필드로 정의해 놓을 수 있다. 따라서 사용자에게 키워드 방식의 검색 인터페이스를 제공할 수 있을 뿐만 아니라, 구조 검색을 위한 조인을 수행할 필요도 없게 된다. 결국 위에서 제시된 두 가지 단점을 효과적으로 극복할 수 있다. 이러한 XML 정보 검색 시스템을 구현하기 위해서는 XML 문서 변환기, 구조 문서 저장소, 구조 문서 검색기, 그리고 검색 결과의 계층적 출력 인터페이스가 필요하다. 본 논문에서는 XML 문서 변환기와 이 변환기를 이용하여 검색 결과를 어떻게 계층적으로 출력할 수 있는지에 대해 설명하기로 한다.

2. XML 문서 검색 시스템

본 논문에서 제시하는 XML 문서 변환기는 KRISTAL-2002[2]라는 정보검색관리시스템에서 이용할 수 있도록 설계되었다. KRISTAL-2002에서의 XML 문서 검색은 일반 문서 검색을 기반으로 이를 확장하여 처리하는 방식을 채택하고 있다. KRISTAL-2002에 XML 문서를 적재하기 위한 과정은 그림 1과 같다.

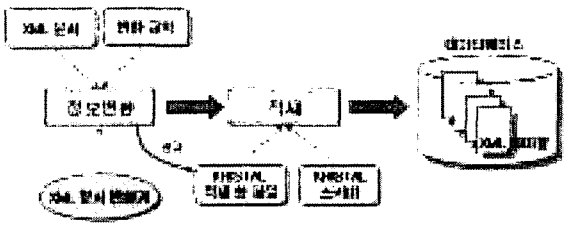

그림 1 XML 문서 적재 과정

그림 1에서 보여 지는 것과 같이 KRISTAL-2002에 XML 문서를 적재하기 위해서는 원본 XML 문서와 XML 문서를 변환하기 위한 변환 규칙이 필요하다. 서비스할 XML 문서에 대한 변환 규칙이 정의가 되면 1)XML 문서 변환기를 이용하여 변환 작업을 수행하고, 2)스키마를 이용하여 결과로 생성된 적재 형 데이터를 저장 시스템에 적재한다.

3. XML 문서 변환기

XML 문서 변환기는 XML 문서의 부모-자식 관계, 형제 관계 등의 계층 정보를 유지하면서 문서를 단편화 한다. 이때 사용자가 입력한 변환 규칙을 이용하여 XML 문서를 구조적 특징을 포함한 단편화된 XML 문서 노드들(Document Fragments)로 변형시킨다.

원본 XML 문서에 변환 규칙을 적용하여 생성된 최종 문서는 원본 XML 문서의 구조 정보를 유지하면서 보다 다루기 편한 트리 형태로 재구성된다.

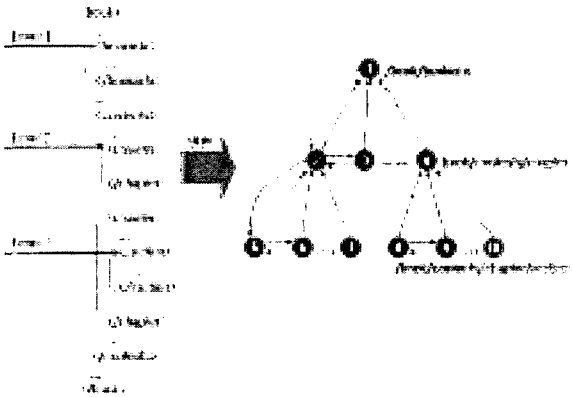

그림 2 XML 문서의 단편화된 트리 표현

그림 2를 통해 알 수 있듯이 XML 문서 변환기를 이용하면 원본 XML 문서가 가지고 있던 복잡한 엘리먼트 계층 구조를 의미 있는 엘리먼트를 중심으로 한, 몇 레벨의 간략한 계층의 트리 구조로 재구성할 수 있다. 또한, 재구성된 트리 역시 원래의 XML 문서가 가지고 있던 노드 간의 부모-자식 관계 및 형제 관계를 그대로 수용하고 있어 원본 문서 구조 형태로의 복원도 지원할 수 있다.

예를 들어, 관리자가 전자책 XML 문서의 /book/contents/chapter를 하나의 의미 있는 검색 단위로 설정하고 chapter 엘리먼트의 하위에 존재하는 chapter-title 엘리먼트를 CHAPTER_TITLE이란 색인 필드로 정의해 놓는다면, 사용자는 단순히 CHAPTER_TITLE이란 필드를 대상으로 키워드 검색을 수행해서 chapter 단위의 상세 검색을 수행할 수 있고 검색 결과를 생성하기 위한 book, contents, chapter간의 조인도 필요가 없게 된다. 또한 단편 노드의 텍스트, 이 예제에서는 chapter 및 그 하위 엘리먼트들을 대상으로 다양한 색인 방식을 취할 수 있어 단편 노드들에 대한 정확한 랭킹 결과도 지원할 수 있다. 최종 검색 결과 화면은 사용자가 정의한 계층 관계에 의해서 구조 정보를 갖는 트리 형태로 구성되어 보여진다.

3.1 문서 변환 규칙(Rule DTD)

XML 문서를 KRISTAL-2002에 적재하기 위해서는 먼저 변환 규칙을 정의해야 한다. 변환 규칙은 XML 문서에 대한 검색 및 브라우징 서비스를 위해 정의하는 것으로서, 계층 관계 및 병합 정보, 그리고 색인 필드에 관련된 각종 정보를 담고 있는 XML 문서이다. XML 문서를 단편화 시키는데 필요한 변환 규칙 DTD는 다음과 같다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!ELEMENT Rule (LevelInfo+)>
<!ATTLIST Rule
    nodeRelation     (YES | NO) #IMPLIED
    nodeInclusion    (YES | NO) #IMPLIED>
<!ELEMENT LevelInfo (MergeFieldList?, IndexFieldList?)>
<!ATTLIST LevelInfo
    no           CDATA    #REQUIRED
    path         CDATA    #REQUIRED
    constraint   CDATA    #IMPLIED>
<!ELEMENT MergeFieldList (MergeField+)>
<!ATTLIST MergeFieldList
    delimiter    CDATA    #IMPLIED>
<!ELEMENT MergeField EMPTY>
<!ATTLIST MergeField
    path         CDATA    #IMPLIED
    attr         CDATA    #IMPLIED
    type         (SELF-TEXT | SINGLE-TEXT | MULTI-TEXT |
ALL) #IMPLIED
    delimiter    CDATA    #IMPLIED
    length       CDATA    #IMPLIED>
<!ELEMENT IndexFieldList (IndexField+)>
<!ELEMENT IndexField EMPTY>
<!ATTLIST IndexField
    name         CDATA    #REQUIRED
    path         CDATA    #IMPLIED
    attr         CDATA    #IMPLIED
    type         (SELF-TEXT | SINGLE-TEXT | MULTI-TEXT |
ALL) #IMPLIED
    delimiter    CDATA    #IMPLIED>
```

규칙 정보(Rule element)에서는 레벨, 병합, 색인에 관련된 변환 규칙에 대해 정의한다. 레벨 정보(LevelInfo element)는 XML 문서의 계층 관계를 명시하거나 사용자가 별도로 레벨을 지정함으로써 계층 관계를 새롭게 생성하는데 사용된다. 병합 정보(MergeField element)는 레벨 정보로 설정된 노드를 식별하기 위한 필드로서, 적재 과정에서 노드 병합 시 서로 동일한 노드인지를 식별하기 위한 용도로 사용된다. 색인 정보(IndexField element)는 레벨 정보로 설정된 노드의 사용자 정의 색인 필드로서, 노드 안의 세부 엘리먼트 및 속성에 대해서 색인과 검색을 지원하기 위해서 사용된다. 문장 추출 방식(type attribute)은 XML 문서에서 텍스트를 어떠한 형태로 뽑아 올 것인지를 지정하는 속성으로서, 병합 정보 생성 혹은 색인을 위한 텍스트 추출 시에 사용된다.

3.2 문서 단편화 알고리즘

XML 문서를 단편화시키기 위해서 DFS(Depth First Search) 방법을 사용하는데 알고리즘은 다음과 같다.

```
XML_Fragment_Algorithm(Node node) {
1. 파라메터로 전달된 노드의 자식 노드들을 하나씩 가져온다.
2. 자식 노드의 타입이 엘리먼트 노드인지를 비교한다.
3. 엘리먼트 노드이면 사용자가 단편화를 요구한 노드인지를 검사한다.
4. 단편화를 요구한 노드라면 검색 서비스의 기본 단위가 되는 노드이기 때문에 아이디를 새롭게 부여하여 단편화 시킬 준비를 한다.
(단편화를 요구한 노드가 아니면 검색 서비스의 기본 단위가 되는
```

노드가 아니기 때문에 해당 노드를 매개변수로 하여 "XML_Fragment_Algorithm"을 재 호출한다. 즉 해당 노드의 자식 노드들 중에서 단편화를 요구한 노드가 있는지를 계속해서 검사한다.)
5. 현재의 노드를 단편화시키기 이전에 하위 노드에서 단편화를 요구한 노드가 존재하면 그것을 먼저 처리하도록 한다. 따라서 이미 단편화 시킬 노드가 발견되었을 지라도 노드에 대한 중요 정보를 추출하기 이전에 해당 노드를 매개변수로 하여 "XML_Fragment_Algorithm"을 재 호출한다.
6. 하위 노드에 대한 단편화 처리가 끝났으면 현재의 노드에서 병합 정보, 색인 정보, 단편화된 문서, 그리고 노드의 부모, 자식, 형제간의 구조 정보를 추출한다. 여기서 추출된 정보는 검색 및 결과의 계층적 출력 인터페이스에 사용된다.
7. XML 문서의 루트 노드로부터 시작해서 위의 과정이 모두 끝나면 여러 개의 단편화된 문서(Document Fragment)들이 존재하게 된다. 이렇게 단편화된 문서는 저장 시스템에 저장되어 효과적인 검색을 위해서 사용되어진다.
}

4. 계층적 결과 표현

실험에 사용된 데이터는 인물, 역사, 문학서적에 관련된 전자책으로 XML 문서로 작성되어 있다. 사용한 전자책의 문서 계층도는 매우 복잡한데 이것을 사용자가 쉽게 이해하고 접근할 수 있는 bookinfo, chapter, section 엘리먼트를 중심으로 한 3 레벨의 문서 계층으로 변환하였다. 각 레벨을 구성하는 단편 노드에 대해서는 형태소 분석을 이용한 색인 작업을 수행하였다. 변환 규칙을 이용하면 각각의 단편화된 노드 안에서 특별히 중요한 엘리먼트에 대한 상세 검색도 가능하다. 사용자가 특정 엘리먼트에 대한 검색을 원하는 경우, 검색인터페이스에서 특정 엘리먼트와 연결된 검색 대상 필드를 선택하고 질의를 입력하면 단편 노드 안의 특정 엘리먼트에 대한 상세 검색을 수행할 수 있다.

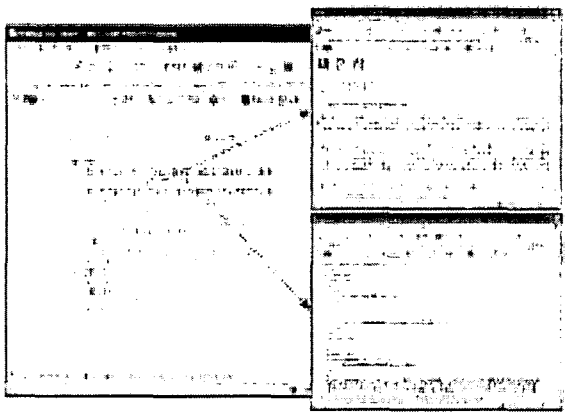

그림 3 XML 검색 예제 화면

XML 문서 변환기를 이용해서 구현한 시스템의 검색 결과는 그림 3과 같다. 검색된 단편 노드들은 사용자가 변환 규칙에서 지정한 레벨에 맞게 계층화된 트리 구조로 재구성되어서 보여 진다. 화살표는 각각 XML 문서 검색 결과를 스타일 시트를 적용하여 브라우징한 것과 스타일 시트 없이 구조 정보를 그대로 출력한 것을 나타낸다.

검색 결과를 보면 실제로 검색된 단편 노드 이외에도 해당 노드의 조상 노드들(section의 경우 chapter와 bookinfo, chapter의 경우 bookinfo)이 함께 검색되는 것을 확인할 수 있다. 검색된 단편 노드들은 문서 변환 규칙에 정의되어 있는 자신의 조상 노드로 지정된 노드를 함께 출력하도록 되어 있기 때문이다. 이렇게 함으로써 검색 키워드가 들어가 있는 section만을 브라우징해서 볼 수도 있고, 해당 section이 어떤 chapter에서 검색된 것인지, 그리고 해당 chapter 가 어떤 book에서 검색된 것인지를 더불어 알 수가 있다. 반대로 book에 어떤 chapter가 있고, 해당 chapter에 어떤 section이 있는지는 하위 노드 브라우징을 통하여 알 수가 있다. 결국 복잡한 계층 구조의 XML 문서를 서비스의 목적에 따라서 중요하다고 판단되는 단편 노드의 단위를 구분하고 단편 노드 안에 존재하는 세부 엘리먼트에 대해서 색인 필드를 미리 구성해 놓음으로써, 사용자에게 보다 효율적인 XML 문서 검색 서비스를 제공할 수 있다.

5. 결론 및 향후 연구

XML 문서 변환기를 사용한 XML 문서 검색 시스템의 장점은 서비스 공급자인 관리자와 서비스 수요자인 사용자 입장에서 살펴볼 수 있다. 먼저 관리자 입장에서는 원본 XML 문서의 복잡한 계층 구조로 인해 사용자에게 관리 및 검색 서비스를 제공하기 어려운 경우에 XML 문서 변환기를 사용함으로써 문서의 중요한 엘리먼트를 중심으로 계층 구조를 간략하게 재구성하여 보다 쉽게 관리 및 검색 서비스를 제공할 수 있다. 사용자 입장에서는 복잡한 XML 관련 문법을 알지 못하더라도, 키워드 방식의 인터페이스를 사용해 쉽게 XML 문서 검색 서비스를 이용할 수 있다.

향후 연구로는 사용자의 다양하고 복잡한 요구 사항을 충분히 반영할 수 있도록 변환 규칙에 대한 보다 상세하고 다양한 명세 작업이 필요하다.

6. 참고 문헌

[1] XPath: XML Path Language, Available at: http://www.w3.org/TR/xpath.
[2] KRISTAL-2002: Information Retrieval & Management System, Available at: http://giis.kisti.re.kr.