

XML 데이터 지원을 위한 KRISTAL 저장 시스템 연구

정창후^o, 최윤수, 진두석, 서정현
한국과학기술정보연구원
{chjeong^o, armian, dsjin, jerry}@kisti.re.kr

Research for the support of XML in KRISTAL storage system

Chang-Hoo Jeong^o, Yun-Soo Choi, Doo-Seok Jin, Jerry Seo
Korea Institute of Science and Technology Information

요 약

초기의 XML 데이터베이스 시스템은 데이터의 특성과 종류, 사이즈에 대한 충분한 고려 없이 구축되어, XML 문서가 급속하게 축적, 증가함에 따라 예상하지 못했던 성능, 확장성 등의 문제에 직면하게 되었다. 따라서 응용분야별로 적절한 방법론을 선택하여 데이터를 저장하고 관리하는 것이 효과적으로 판단된다. 본 논문에서는 KRISTAL이라는 정보검색관리시스템에서 XML 문서에 대한 저장 및 관리 서비스를 제공하기 위해서, 문서 저장, 문서 복원, 문서 관리, 그리고 문서 색인에 관련된 내용에 대해서 살펴본다. 복잡한 계층 구조의 XML 문서를 서비스의 목적에 따라서 중요하다고 판단되는 단편노드의 단위로 구분하고 단편노드 안에 존재하는 세부 엘리먼트 및 속성에 대해서 별칭을 사용하는 검색 필드를 미리 구성해 놓음으로써, 문서의 중요한 엘리먼트를 중심으로 계층 구조를 간략하게 재구성하여 보다 쉽게 관리 및 검색 서비스를 제공할 수 있다.

1. 서 론

웹상에서의 구조적 데이터 표현 및 문서 교환의 새로운 표준으로 1998년 W3C에서 XML(eXtensible Markup Language)[1]이 채택된 후, 사용이 간편하고 재사용성 및 확장성이 뛰어나다는 장점을 기반으로 전자상거래, 전자책 등 많은 분야에서 활용되고 있다. 또한 XML 관련 소프트웨어나 도구, XML을 지원하기 위한 다른 표준들이 속속 발표됨에 따라 점차 웹상의 모든 문서가 XML로 대체되거나 빠르게 변경되고 있다. 이러한 추세를 고려하면 앞으로도 XML 관련 문서의 양은 더욱 증가할 것이며 그 사용 범위도 확대될 것이기 때문에, XML 문서를 효과적으로 저장하고 관리하는 방법에 대한 연구가 필요하다.

XML 데이터베이스 혹은 어떤 데이터베이스든지 간에 핵심은 정보를 어떻게 저장하고 색인하며 질의를 수행하느냐 하는 것이다. 이러한 디자인 세부사

항은 시스템 외부에서 볼 때는 분명하지 않을 수 있으나, 데이터베이스 시스템의 기능, 성능, 확장성 부분에 중대한 영향을 미친다. XML 데이터베이스는 XML 문서를 저장하고 검색하는 논리 모델을 가지고 있으나, 이것의 내부 저장 모델은 문서와 반드시 일치해야 될 필요는 없다. 여러 가지 측면에서 문서의 저장 포맷은 데이터베이스에서 지원될 수 있는 특성들을 결정하게 된다.

2. 관련 연구

현재 XML은 인터넷 상의 데이터 표현 및 교환의 표준으로 간주되고 있으며 점점 더 많은 데이터들이 XML을 기반으로 표현되고 있는 실정이다. XML 데이터를 검색하기 위한 질의 언어로 XPath, XQuery 등이 제안되었으며, 이러한 질의 언어들은 구조적 특징을 지니는 XML 데이터를 탐색하기 위하여 경로 수식을 기반으로 하고 있다. 따라서, XML 데이

터의 비정형적 특성을 반영하면서도 경로 수식의 처리를 효과적으로 지원할 수 있는 다양한 저장 기법들이 제안되었다. 일반적으로, XML 질의 수행 성능에 가장 큰 영향을 주는 요소는 XML 데이터를 어떠한 형태로 저장, 관리하느냐 하는 것이다. 따라서, 현재 XML 분야에서는 XML 데이터를 보다 효과적으로 저장함으로써 XML 질의의 성능을 향상시키는 것이 주요 화두이다. XML 저장에 관련된 다양한 논문이 발표되고 있는데, 이것들은 대체적으로 5가지의 큰 흐름으로 분류해 볼 수 있다[2]. 첫째, 하부 저장 시스템에 따른 분류로서 파일 시스템을 이용한 저장 기법, 관계형 데이터베이스를 이용한 저장 기법, 객체지향 데이터베이스를 이용한 저장 기법, 객체관계형 데이터베이스를 이용한 저장 기법, Native XML 데이터베이스를 이용하는 방법이 있다. 둘째, 저장 방식에 따른 분류로서 분할 저장 방식과 비분할 저장 방식이 있다. 셋째, DTD 처리 방식에 따른 분류로서 DTD 종속적 기법과 DTD 독립적 기법이 있다. 넷째, 매핑 정의 방법에 따른 분류로서 시스템이 자동으로 매핑을 정의하는 방법과 사용자가 수동으로 매핑을 정의하는 방법이 있다. 이러한 각각의 XML 저장 기법들은 나름대로의 고유한 장단점을 지니고 있으므로 각 응용분야에 적합한 기법을 채택하여 XML 데이터를 효율적으로 관리하여야 한다. 초기의 XML 데이터베이스 시스템은 데이터의 특성과 종류, 사이즈에 대한 충분한 고려 없이 구축되어, XML 문서가 급속하게 축적, 증가함에 따라 예상하지 못했던 성능, 확장성 등의 문제에 직면하게 되었다. 따라서 응용분야별로 적절한 방법론을 선택하여 데이터를 저장하고 관리하는 것이 효과적으로 판단된다.

3. KRISTAL XML 저장관리 시스템

XML 검색은 일반적으로 XML 문서의 계층적인 구조에 기반을 두고 있다. 이러한 계층적인 구조를 검색하기 위해서 XPath나 XQuery와 같은 XML 질의 표현식을 사용하는데, 이러한 질의는 사용자가 XML 문서상의 계층 구조를 정확하게 알고 있어야만 사용이 가능하다. 또한 XML 질의를 관계형 데이터베이스 시스템에서 처리하기 위해서는 개체와 개체 사이의 계층적인 구조를 검색하기 위한 빠른 조인 알고리즘이 필요하다. 그러나 KRISTAL[3]에서는 관리자가 전처리 과정으로 XML 문서의 계층

관계를 하나의 검색 필드로 정의해 놓을 수 있기 때문에 사용자에게 키워드 방식의 검색 인터페이스를 제공할 수 있을 뿐만 아니라, 구조 검색을 위한 조인을 수행할 필요도 없게 된다. KRISTAL에서의 XML 문서 검색은 일반 문서 검색을 기반으로 이를 확장하여 처리하는 방식을 채택하고 있다. KRISTAL에서 XML 문서를 서비스하기 위한 과정은 그림 1과 같다.



(그림 1) XML 문서 적재 과정

그림 1에서 보여지는 것과 같이 KRISTAL에서 XML 문서를 서비스하기 위해서는 원본 XML 문서와 XML 문서를 변환하기 위한 변환 규칙[4]이 필요하다. 서비스할 XML 문서에 대한 변환 규칙이 정의가 되면 XML 문서 변환 모듈을 이용하여 변환 작업을 수행하고, 스키마를 이용하여 결과로 생성된 적재형 데이터를 저장 시스템에 적재한다. 단위 작업에 대해서 좀 더 세부적으로 살펴보면 다음과 같다.

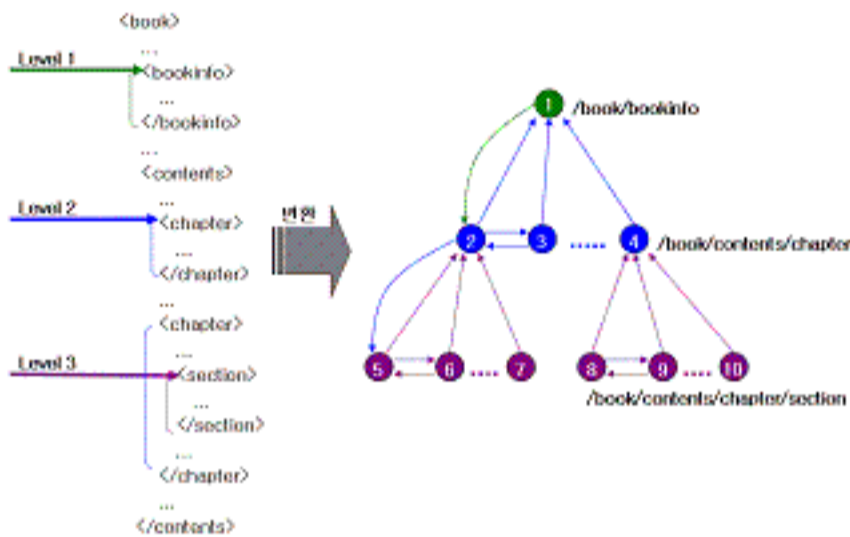
3.1 문서 저장

XML 데이터베이스의 핵심은 저장 포맷과 저장단위를 설정하여 문서를 저장하는 방식이다. 이러한 저장 방식은 어떤 모델로 문서를 저장하느냐에 따라 다양한 특성을 갖게 되는데, 잘 정제된 문서 저장은 질의 처리 혹은 질의 언어의 지원과 밀접하게 연관되어 있기 때문에 저장될 데이터 모델의 선택에 신중을 기해야 한다.

KRISTAL에서의 데이터 모델은 아주 간단하다. KRISTAL에서는 XML 문서를 의미 있는 엘리먼트를 중심으로 Document Fragment로 나눈다. 나누어진 Document Fragment는 하나의 노드 객체를 나타내고 각각의 노드 객체는 DOM과 같은 트리 구조로 재구성되어 저장된다. KRISTAL에서의 데이터 모델이 DOM과 다른 점은 DOM에서는 트리 구조가 엘리먼트를 기반으로 형성되는 데 반해 KRISTAL에서는 트리 구조가 Document Fragment를 기반으로

형성된다는 것이다. DOM과 마찬가지로 조상, 부모, 자식, 후손, 형제 노드를 얻어오는 API가 존재한다.

XML 문서를 Document Fragment로 나누는 문서 분할은 복잡한 계층의 XML 문서를 의미 있는 엘리먼트를 중심으로 계층을 단순화시켜서 검색 및 계층적 결과 표현에 사용하도록 하는 작업으로, XML 문서의 부모-자식 관계, 형제 관계 등의 계층 정보를 유지하면서 문서를 단편화한다. 이때 계층 관계 및 병합 정보, 그리고 검색에 관련된 각종 정보를 정의하고 있는 변환 규칙을 이용하여 XML 문서를 구조적 특징을 포함한 단편화된 Document Fragment로 변형시키는데, 이 각각을 단편노드라고 부른다. 이러한 단편노드는 원본 XML 문서의 구조 정보를 유지하면서 다루기 편한 트리 형태로 재구성된다.



(그림 2) XML 문서의 단편화된 트리 표현

그림 2를 통해 알 수 있듯이 XML 문서를 단편노드로 분할하면 원본 XML 문서가 가지고 있던 복잡한 엘리먼트 계층 구조를 의미 있는 엘리먼트를 중심으로 이루어진 몇 레벨의 간략한 계층의 트리 구조로 재구성할 수 있다.

3.2 문서 복원

XML 문서 복원은 데이터베이스에 저장되어 있는 XML 문서의 단편들을 조합하여 하나의 완성된 XML 문서를 생성하는 것을 의미한다. XML 문서 분할 모듈을 통해서 단편화된 노드 상태로 데이터베이스에 저장된 문서 조각들을 역으로 재조합하여 처음의 상태로 되돌아가도록 하는 작업으로 라운드 트리핑(Round Tripping) 조건을 만족시켜야 한다.

KRISTAL에서의 문서 복원은 검색 결과 생성을 위한 부분 문서 복원과 원본 문서 재현을 위한 전체

문서 복원이 있다. 부분 문서 복원은 검색된 단편노드들이 변환 규칙에서 지정된 레벨에 맞게 계층화된 트리 구조로 구성될 때 발생하는데, 이때 단편노드의 내용은 이미 XML로 구성되어서 저장되었기 때문에 엘리먼트들을 재조합해주는 작업은 필요하지 않다. 따라서 검색 결과 생성에 대한 오버헤드를 제거할 수 있다. 그리고 각각의 검색된 단편노드를 보다가 XML 문서 전문 보기를 원할 경우에 전체 문서 복원 기능을 이용한다. KRISTAL에서 문서를 복원하였을 경우에 라운드 트리핑의 정도는 변환 규칙을 어떻게 정의하느냐에 따라서 달라진다. 따라서 역으로 재생된 문서가 원본과 반드시 일치하지 않을 수 있다. 하지만, 재생된 문서를 XML 문서 변환 모듈을 통하여 다시 분할시켜서 적재하였을 경우에는 원래 데이터베이스에 저장되어 있던 상태 그대로 데이터베이스를 복원할 수 있다. 즉, $A \rightarrow B$ (A : XML 문서, B : 단편노드 집합)로 변환된 이후에 $B \rightarrow A$ 가 될 수도 있고, $B \rightarrow A'$ (A' : A 와 유사한 문서)이 될 수도 있지만, XML 문서 변환 모듈을 통하여 다시 문서를 분할하였을 경우에는 $A \rightarrow B$ 가 되고, $A' \rightarrow B$ 가 된다는 것을 보장하고 있다. 따라서 원본 문서로의 복원은 100% 라운드 트리핑이 이루어지지 않을 수 있으나 복원된 문서를 가지고 다시 변환했을 경우에는 원래의 데이터베이스 구조를 그대로 유지하기 때문에 문서의 백업이나 브라우징을 위한 목적으로 사용될 수 있다.

3.3 문서 관리

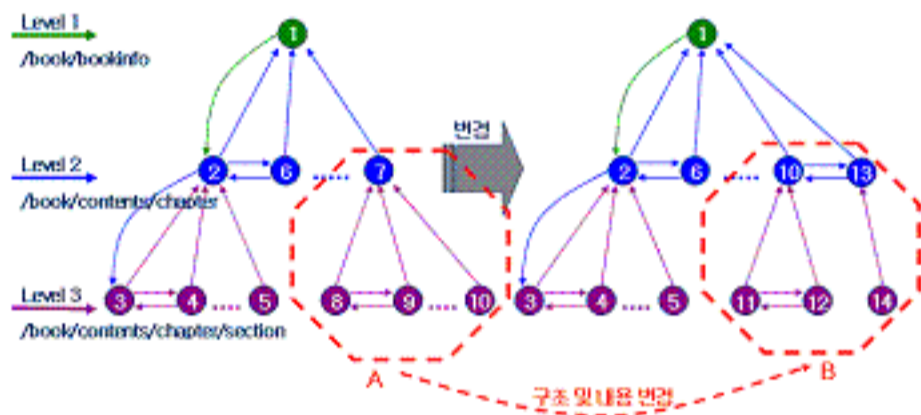
문서 관리란 데이터베이스에서 새로운 문서를 생성하고, 기존의 문서를 삭제하거나 수정하는 작업을 의미한다. 가상 분할 저장 방식에서는 문서의 어느 일부가 수정된다면 문서 내에서 영향을 받는 모든 엘리먼트들의 위치 정보도 수정해야 한다. 이것은 잠재적으로 비용이 많이 들지만, 단순하다는 장점이 있다. 분할 저장 방식은 문서 전체를 수정하지 않고, 문서의 일부 조각만을 분리해서 수정할 수 있다. 이러한 방법은 잠재적인 성능과 확장성에 이점이 있지만, 정확하고 효율적으로 수행하기가 매우 어렵기 때문에 시스템 개발자들은 많은 고민을 해야 한다.

KRISTAL에서 XML 문서를 변경하기 위한 사용자 작업은 단편노드 단위로 문서를 변경하는 방법과 단편노드의 범위를 벗어난 노드그룹 단위로 문서를 변경하는 방법으로 구분해서 살펴볼 필요가 있다.

먼저 단편노드 단위로 문서를 변경하는 작업에 대

해서 살펴보도록 하겠다. 단편노드 삽입의 경우에는 단편노드가 삽입될 위치와 기존 노드를 지정하여 단편노드를 새롭게 추가한다. 단편노드 삭제의 경우에는 사용자가 지정한 단편노드의 아이디와 단편노드가 속한 테이블 아이디를 이용하여 단편노드를 제거한다. 단편노드 수정의 경우에는 단순히 단편노드의 XML 일부 내용만 수정하면 된다. 이와 같은 작업은 레코드 기반의 데이터 중심 XML 문서의 변경에서 유용하게 사용될 수 있다.

다음으로 단편노드의 범위를 벗어난 노드그룹 단위로 문서를 변경하는 작업에 대해서 살펴보도록 하겠다. 노드그룹 삽입이나 노드그룹 제거는 그룹의 가장 상위 단편노드에 대해서만 변경해주면 되기 때문에, 단편노드 단위로 이루어지는 방법과 동일하다. 그러나 노드그룹 변경의 경우에는 단순히 가장 상위 단편노드 하나만 변경하는 것이 아니라 그룹에 포함되어있는 모든 단편노드간의 계층 정보까지도 함께 변경해야 한다. 이와 같은 작업은 계층 관계가 중요한 문서 중심 XML 문서의 변경에서 유용하게 사용될 수 있다.



(그림 3) XML 문서의 변경

그림 3에서 보여지는 것과 같이 여러 개의 단편노드를 한꺼번에 수정하려고 하는 경우 XML 문서는 단편노드의 XML 일부 내용뿐만 아니라 단편노드간의 계층 관계도 함께 변경이 일어나기 때문에 단편노드 단위의 수정이 아니라 노드그룹 단위의 수정이 일어나야 한다. 그림 3의 경우에 관리기를 통하여 노드그룹 A를 수정했기 때문에 새롭게 생성된 노드그룹 B를 기존에 수정하고자 했던 노드그룹의 가장 상위 단편노드 위치에 연결시켜 준다. 이 작업은 기존 노드그룹을 삭제하고 새로운 노드그룹을 삽입하는 작업으로 구현된다.

대용량 데이터베이스 상에서 단편노드 혹은 노드

그룹을 삭제하거나 삽입할 경우 색인을 갱신하는 시간이 많이 걸릴 수 있기 때문에, 삭제는 삭제 플래그를 사용하고 삽입은 보조 색인 테이블을 이용하여 빠르게 작업을 수행한다.

3.4 문서 색인

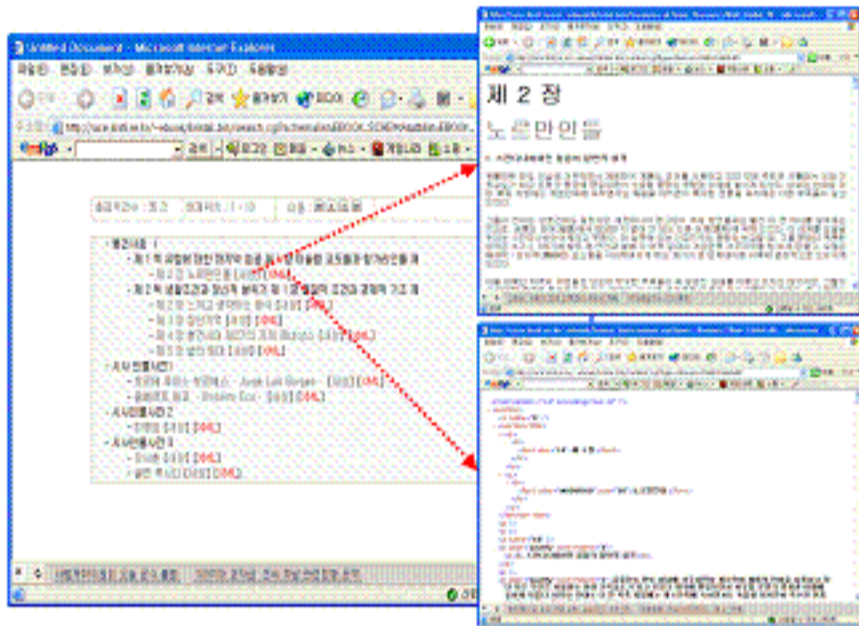
색인의 적절한 명세와 사용은 많은 양의 데이터에서 검색 속도를 증가시킨다. 그러나 색인된 데이터가 자주 업데이트될 때 재 색인을 위해서 사용되는 시간은 색인된 접근의 이점을 제거시킬 수 있기 때문에 실시간으로 색인을 변경시키는 작업은 많은 주의를 요한다. 문서 생성, 문서 제거, 문서 수정과 같은 문서 관리 작업은 색인에 영향을 주기 때문에 계산량의 관점에서 비용이 많이 들뿐만 아니라, 잠재적인 동시성 제어 문제를 발생시킨다. 따라서 색인 갱신을 실시간으로 수행하기 위한 여러 가지 방법들이 연구되어 왔는데, KRISTAL에서는 보조 테이블을 이용하는 방법으로 위의 문제를 해결한다. 초기 대량으로 문서를 적재할 때에는 주 테이블에 색인을 생성하고, 이후에 나타나는 관리 작업으로 인해 색인이 갱신될 때에는 보조 테이블을 사용한다. 보조 테이블을 사용함으로써 색인 갱신 비용을 획기적으로 제거할 수 있다. 다만 이 경우에 주 테이블과 보조 테이블을 주기적으로 통합시켜주는 색인 최적화 작업이 추가적으로 요구된다.

4. 실험 및 결과 표현

실험은 문서 중심 XML 데이터베이스와 데이터 중심 XML 데이터베이스의 두 가지 예를 대상으로 수행하였다.

먼저 문서 중심 XML 데이터베이스에 사용된 실험은 전자책 데이터베이스를 이용하여 수행하였다. 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 4) 문서 중심 XML 검색 예제

XML 문서 변환 모듈을 이용해서 구현한 시스템의 검색 결과는 그림 4와 같다. 검색된 단편 노드들은 사용자가 변환 규칙에서 지정한 레벨에 맞게 계층화된 트리 구조로 재구성되어서 보여진다. 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 chapter가 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.

다음으로 데이터 중심 XML 데이터베이스에 사용된 실험은 한의약 데이터베이스를 이용하여 수행하였다. 실험에 사용된 데이터는 처방정보, 천연약물, 화학정보, 병증사전, 그리고 수치사전에 관련된 5개의 XML 문서 테이블과 약재형상, 분자구조식을 나타내는 2개의 이미지 문서 테이블로 구성되어 있다. 실험에 사용된 데이터베이스의 특징으로는 각각의 XML 테이블 간에 복잡한 연결 관계가 존재한다는 것이고, 검색 결과 생성 시에 XML 테이블 사이의 네비게이션을 지원해야 한다는 것이다.



(그림 5) 데이터 중심 XML 검색 예제

그림 5는 병증사전 테이블에서 “가슴&답답”으로 검색한 결과의 한 예이다. 그림 5에서 보여지는 것과 같이 XML 검색 결과의 단위인 단편노드는 각 노드를 기준으로 해서 자신을 포함하고 있는 부모 단편노드로의 이동(①), 자신과 형제 관계인 이전 단편노드와 다음 단편노드로의 이동(②, ③), 자신의 자식들 중 첫 번째 단편노드로의 이동(④)을 지원한다. 또한 ⑤에서 보여지는 것과 같이 하나의 필드에 여러 개의 값을 가지고 있는 집합 값의 형태도 효과적으로 표현할 수 있다. 만일 분할 모델에서 ⑤와 같은 집합 값을 표현하려고 하면 저장 시에 집합 값을 모두 분리했다가 추출 시에 다시 재조합해야 하는 번거로움이 발생한다. 또한 비분할 모델에서도 ⑤와 같은 집합 값을 표현하려고 하면 집합 값 중의 하나가 추가, 삭제 및 수정이 발생하였을 경우 위치 값을 모두 재조정해줘야 하는 번거로움이 있다. 그러나 문서의 의미 있는 엘리먼트를 기준으로 문서를 분할하고, 그 분할된 단편노드들 간의 계층 관계를 유지하는 저장 구조에서는 과학 데이터나 카탈로그와 같은 레코드 기반의 데이터 중심 XML 문서를

서비스할 때 하나의 단편노드에 존재하는 XML의 일부 내용만 수정해주면 되기 때문에 훨씬 효율적인 시스템을 구성할 수 있다.

5. 결론 및 향후 연구

XML 문서에 대한 저장 및 관리 서비스를 제공하기 위해서 문서 저장, 문서 복원, 문서 관리, 그리고 문서 색인에 관련된 내용에 대해서 살펴보았다. KRISTAL은 분할 저장 방식과 비분할 저장 방식의 중간자적인 형태를 취함으로써 문서의 구조적인 정보를 효과적으로 표현하고, 검색 결과 재구성시에 노드 취합에 발생하는 오버헤드를 제거할 수 있다. 이것은 실제적으로 XML 문서 검색 서비스를 실용화 시키는데 있어서 가장 중요한 두 가지 요소라고 생각된다. 또한 갱신 연산의 경우에는 갱신할 노드의 범위가 제한되어 내용 갱신 및 색인 갱신의 오버헤드를 줄일 수 있다.

KRISTAL XML 저장관리 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, 복잡한 계층 구조의 XML 문서를 서비스의 목적에 따라서 중요하다고 판단되는 단편노드의 단위로 구분하고 단편노드 안에 존재하는 세부 엘리먼트 및 속성에 대해서 별칭(alias)을 사용하는 검색 필드를 미리 구성해 놓음으로써 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. 다른 시스템들이 질의 시에 XML 질의 표현식을 사용하는 반면에 KRISTAL은 사용자 변환 규칙을 통하여 전처리 과정으로 질의에 관련된 모든 작업을 처리하기 때문에, 변환 규칙에 좀 더 융통성을 부여할 필요가 있다. 그리고 단편노드에 대한 검색을 위해서 XPath나 XQuery에 대한 지원을 연구해볼 필요가 있다. 현재는 전처리 과정을 통해서 XML 문서 안의 중요 엘리먼트나 속성에 대해서 검색을 지원하고 있지만,

고급 사용자들은 이러한 제약을 불편하게 생각할 수 있다. 따라서 분할된 단편 노드 안의 세부 사항에 대해서 사용자 임의대로 검색을 지원할 수 있도록 XML 관련 질의 언어를 추가적으로 지원해 보는 방향에 대한 연구가 필요하다.

참고 문헌

- [1] Extensible Markup Language(XML) 1.0, "http://www.w3.org/XML".
- [2] 김명호 외 13인, "XML 및 IMQL을 지원하기 위한 저장엔진 개발 연구", 『한국과학기술원 최종 연구보고서』, 2005.
- [3] 최운수, 서정현, 진두석, 김광영, 이민호, 정창후, 주원균, 최성필, 김진숙, "정보검색 관리 시스템 KRISTAL-2002", 『한국인터넷정보학회 추계학술발표논문집』 제5권 제2호, 2004, pp.193-196.
- [4] 정창후, 최운수, 주원균, 진두석, 김광영, 이민호, 서정현, "문서 단편화 기법을 이용한 XML 변환기의 설계 및 구현", 『한국정보과학회 추계학술발표논문집』 제2권, 2004, pp.214-216.