보이지 않는 조상을 포함하는 안전한 XML 문서의 갱신 질의 처리에 대한 연구*

변창우° 박 석
서강대학교 컴퓨터학과
{chang°, spark}@dblab.sogang.ac.kr

Study on Update Processing for Secure XML documents including Invisible Ancestor

Chang-woo Byun° Seog Park
Department Computer Science , Sogang University

요 약

XML이 웹 정보 시스템의 데이터베이스로 활용되면서 공유 부분에 대한 데이터 처리의 높은 효율성을 제공하고자 최소 단위의 접근제어 모델에 대한 연구가 판독 모드 측면에서 활발히 진행되었다. 질의 처리 연구에서는 XML 데이터베이스에 대한 갱신 질의 표준화 작업이 진행되고 있다. 본 논문은 갱신 질의 연산을 최소 단위 접근제어 모델의 연산 모드로 추가함으로써 발생하는 보이지 않는 조상 문제를 정의한다.

이를 해결하기 위한 고려 사항으로 보이지 않는 조상 노드들을 포함하는 XML 문서에 대한 갱신 연산 시 고려해야 할 특성을 *갱신의 비밀성*, *갱신의 무결성*, 그리고 *갱신의 일관성*으로 정의하고 갱신 연산 수행 시 발생될 수 있는 특성 위배 상황을 정리한다.

1. 서론

XML이 웹 정보 시스템을 위한 데이터베이스로 인식되면서, XML 문서 저장 및 선택 질의에 대한 많은 연구가 진행되었고, 갱신 질의에 대한 표준화 작업이 진행 중이다. 또한, 데이터 보안 측면에서는 접근제어 메커니즘을 적용하여 권한 있는 사용자에게 그에 해당되는 문서의 일부분만을 보여주는 안전한 XML 문서에 대한 연구가 진행되어 왔다.

다음 단계는 이런 안전한 XML 문서에 대한 갱신 질의 처리에 대한 연구일 것이고 이때 반드시 해결해야 할 부분은 보이지 않는 조상 문제(invisible ancestor problem)이다.

[정의 1] 보이지 않는 조상 문제: XML 문서에 대한 레이블 트리 하에서 임의의 노드는 접근 불허, 하위 노드는 접근이 허가된 경우 그 임의의 노드를 보이지 않는 조상 노드(invisible ancestor node)라 하고, 이것에 대한 비밀성 처리 문제를 보이지 않는 조상 문제라 한다.

보이지 않는 조상 문제는 세 가지 접근제어 정책-데이터 가용성 정책[1,2], 데이터의 비밀성 정책[3], 데이터의 의미적 충돌 방지 정책[4]-에 따라 선택되는데 본 논문은 지면 한계상 데이터 가용성 정책에 초점을 둔다.

[정의 2] 데이터의 가용성(availability of data) 하부 노드의 접근을 허용하기 위해서 접근 불허된 노드의 시작 태그와 끝 태그만을 보여주어 문서 구조는 그대로 유지하면서 노드의 데이터를 숨기는 접근제어 정책 방법이다.

본 논문은 데이터 가용성 정책이 적용된 안전한 XML 문서상에 갱신 연산을 수행할 때 고려사항들을 제안한다. 논문은 다음과 같이 구성되어 있다. 2장에서는 접근제어 모델과 데이터 가용성 정책에 대한 예제를 설명하고, 3장에서는 보이지 않는 조상에 대한 갱신 연산에 대한 제약사항 및 갱신 질의 재작성을 설명한다. 4장에서 결론 및 추후 연구를 기술한다.

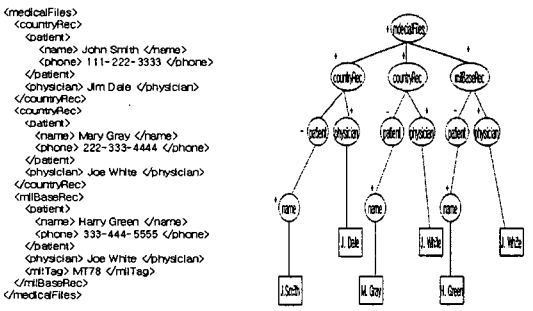
2. 접근제어 모델 및 가정

2.1 접근제어 모델

XML 접근제어 모델[1, 2, 3, 4]에서 제안했던 권한부여 규칙은 다섯-튜플 쌍 {주체, 객체, 접근모드, 허용-불허 기호, 타입}으로 구성된다. XML 문서의 각 엘리먼트는 트리의

*이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음(KR-2004-041-D00569)

노드에 해당하고, 노드들은 XML 문서의 일부분을 표현하기 때문에 XML 문서에서는 엘리먼트에 권한을 부여함으로써, 데이터 일부분에 대한 접근제어가 가능하게 된다. 보이지 않는 조상 부분을 제외한 불허 기호를 갖고 있는 엘리먼트는 삭제되어 사용자에게 보여진다.



[그림 2] a) XML 문서 예제 b) 권한부여된 XML 문서의 레이블 트리

[그림 2.b]는 [그림 2.a]의 XML 문서에 대해 간호사에 대한 권한부여 정책에 따라 간호사 Jane이 볼 수 있는 XML 문서의 레이블 트리이다.

- 간호사는 환자의 이름 정보에는 접근 가능하지만, 전화번호 정보는 볼 수 없다.
- 판독 권한이 없으면 갱신 권한도 없다.

2.2 가정

본 논문을 진행하는데 있어 두 가지의 가정이 정의한다.

첫째, 본 논문에서 보이고 있는 예제 갱신 질의에 대한 유효성 검증[6]은 본 논문의 범위를 벗어나 고려하지 않는다.

둘째, 사용자가 알고 있는 문서 구조는 권한부여 정책이 수행된 뷰([그림 2.b])이다.

3. 보이지 않는 조상을 포함한 XML 문서에 대한 갱신 연산 및 질의 재작성

[5]에서는 삽입, 갱신, 삭제, 재명명, 그리고 대체 연산에 대해 소개하고 있다.

3.1 갱신 연산 처리의 세 가지 특성

안전한 XML 문서에 갱신 연산을 수행하기 위해 세 가지 특성을 정의한다.

- **[갱신 연산의 비밀성]** 갱신 연산 후 접근 불허 데이터의 존재가 노출되어서는 안 된다.
- **[갱신 연산의 무결성]** 갱신 연산에 의해 접근 불허 데이터가 변경되어서는 안 된다.
- **[갱신 연산의 일관성]** 사용자에게 보여지고 있는 XML 뷰에 대한 갱신 연산의 결과와 원래 XML 문서에 대한 갱신 연산을 처리하고 접근제어 정책이 수행된 후의 XML 뷰는 같아야 한다.

3.2 삽입 연산

삽입 연산을 수행할 경우 고려해야 할 부분은 접근 불허 노드에 대한 속성을 삽입할 때이다.

```
For $p In /medicalFiles/countryRec
Update $p { Insert
        <patient><name>chang</name></patient>}
```

간호사 Jane이 환자의 이름을 삽입하는 경우로써, countryRec 엘리먼트와 patient 엘리먼트는 '데이터 가용성' 정책에 의해 Jane에게 보여지고 있는 상황이다. 질의 요청자는 자신이 보고 있는 뷰에 존재하기 때문에 이를 보고 삽입 연산을 수행할 수 있게 된다. 그러나 접근 불허 노드에 속성을 삽입하는 경우 나중에 자신이 삽입한 것을 볼 수 없기 때문에 불허한다.

또한, 스키마 상에 속성의 CDATA 형식이 Fixed거나 Required로 지정되어 있다면 사용자의 삽입에 의해 DTD 위배상황이 발생하게 된다. 이를 고려하면 두 가지 방법이 있다.

방법 1 : 삽입을 거절한다.

방법 2 : 대체 방식을 이용하여 삽입을 허용한다.

방법 1은 사용자에게 DTD 위배라는 사실을 알리고 삽입을 거절하는 경우이다. 이럴 경우 결국 사용자에게 비밀 데이터라는 존재를 알리게 됨으로 [갱신 연산의 비밀성] 특성에 위배된다.

방법 2는 Fixed인 경우 속성이름-초기값 쌍을 삽입하고, Required인 경우 속성이름- null 값을 삽입한다. 방법 2가 타당하다. 따라서, 본 논문은 방법 2를 따른다.

3.3 삭제 연산

삭제 연산을 수행할 경우 기본 정책은 다음과 같다.

[첫째, 보이지 않는 조상 노드 삭제]

```
For $p In /medicalFiles/countryRec[1]
    $c In $p/patient
Update $p { Delete $c }
```

질의 요청자가 patient 엘리먼트를 포함한 하부 노드들을 삭제하는 경우 [갱신 연산의 무결성] 및 [갱신 연산의 일관성] 특성을 위해 접근 허가된 하위 노드들(name 엘리먼트)만 삭제하고 접근 불허된 노드(patient 엘리먼트와

phone 엘리먼트)는 그대로 둔다.

추가적으로 고려해야 할 사항은 갱신 연산에 포함된 경로 표현은 사용자가 보고 있는 뷰에 한정된 것이다. 따라서, 실제 원래 XML 문서에 갱신 연산을 적용하기 위해서는 원래 XML 문서에 맞는 경로 표현으로 갱신 질의가 재작성이 되어야 한다. 즉, 보이지 않는 조상 노드를 기준으로 접근 가능한 하위 노드들이 있으면 각 노드에 대한 바인딩 처리를 하고 삭제 연산은 그 바인딩된 경로를 삭제하는 것으로 변환된다.

```
For $p In /medicalFiles/countryRec[1]
    $temp1 In $p/patient/name
    $temp2 In $p/physician
Update $p{ Delete $temp1
           Delete $temp2 }
```

[둘째, 보이지 않는 조상 노드의 부모 노드 삭제]

```
For $p In /medicalFiles/countryRec[1]
Update $p {
  Delete $p }
```

이런 경우 세 가지의 방법이 있을 수 있다.

방법 1 : 보이지 않는 조상 노드를 삭제한다.

방법 2 : 접근 허가 노드들만 삭제하고 접근 불허 노드는 그대로 둔다.

방법 3 : 삭제 대신 권한부여 규칙을 변경한다.

방법 1은 접근 불허 노드인 비밀 데이터를 삭제하게 되므로 [갱신 연산의 무결성]에 위배된다. 방법 2는 접근 허가 노드들만 삭제하기 때문에 [갱신 연산의 무결성]에는 위배되지 않지만, 부모-자식 구조 정보가 상실되어 [갱신 연산의 일관성]에 위배된다. 따라서, 적합한 방법이라 볼 수 없다. 방법 3은 접근 허가 노드인 보이지 않는 조상 노드의 하위 노드들은 삭제하고, 부모 노드는 놔두는 대신 그 노드에 대한 접근 불허 권한부여 규칙을 추가하는 방법이다. 방법 3은 권한부여 규칙의 생성 및 추가에 대해 추가 작업은 있지만, 갱신 연산의 세 가지 규칙들에 위배되지 않기 때문에 타당하다고 볼 수 있다. 이 경우도 갱신 질의 재작성이 되야 한다.

3.4 대체 연산 및 재명명 연산

대체 연산은 삽입 연산 후 삭제 연산을 수행하는 것과 같은 결과를 나타내기 때문에 삽입 연산 및 삭제 연산에서 요구하고 있는 고려사항들을 그대로 적용하면 된다.

한편, 접근 불허 노드에 대한 재명명 연산을 처리할 때 문제가 발생한다. 재명명 연산을 거절하면 [갱신 연산의 비밀성]에 위배되고, 그렇다고 수행하면 [갱신 연산의 무결성]에 위배된다. 따라서, 이때는 커버 스토리 메커니즘을 적용한다.

4. 결론 및 추후 연구

XML이 대중화되면서 개별적으로 연구되어 온 XML 데이타베이스에 대한 질의 처리 연구와 접근제어 연구를 혼합한 갱신 모드를 확장한 접근제어 정책 기반의 안전한 XML 문서 처리에 초점을 두었다. 경신 연산 시 고려해야 할 사항을 보이지 않는 조상 문제로 정의하고 경신 연산 수행 시 지켜야 할 특성을 갱신 연산의 비밀성, 갱신 연산의 무결성, 그리고 갱신 연산의 일관성으로 분류하여 각 갱신 연산 수행 시 발생될 수 있는 특성 위배 상황을 기술하였다.

추후 연구로는 유사한 방법으로 ' 데이터 의미적 충돌 방지 정책' 시 갱신 연산의 고려사항을 살펴 보고, 위배 상황을 자동으로 감지할 수 있는 감지기 개발 및 질의 재작성을 자동으로 수행할 수 있는 알고리즘 개발이 필요하다.

5. 참고문헌

[1] E. Damiani, S. Vimercati, S. Paraboachk and P.Samarati, " A Fine-grained Access Control System for XML Documents" , ACM Trans. Information and System Sec., Vol.5, No.2, May 2002.

[2] E. Bertino, S. Castano, E. Ferrari, M. Mesiti, "Specifying and Enforcing Access Control Policies for XML Document Sources", WWW Journal, Baltzer Science Publishers, Vol.3, N.3, 2000.

[3] A. Gabillon and E. Bruno, " Regulating Access to XML Documents" , In Proc. IFIP WG11.3 Working Conference on Database Security, 2001.

[4] A. Stoica and C. Frakas, " Secure XML Views" , In Proc. IFIP WG11.3 Working Conference on Database and Application Security, 2002.

[5] Igor Tatarinov, Zachary G. Ives, Alon Y.Halevy, Daniel S.Weld. " Updating XML" , ACM SIGMOD, Santa Barbara, California, USA, May, 2001, pp .413-424.

[6] Sang-Kyun Kim, Myungcheol Lee, Kyu-Chul Lee. Validation of XML Document Updates based on XML Schema in XML Databases. DEXA 2003, Lecture Notes in Computer Science, Vol.2736, pp.98-108, Sep. 2003.