XML 문서의 접근 권한 관리 시스템 개발

김동신, 이용규
동국대학교 컴퓨터공학과

Development of an Authorization System for XML Documents

Dong Shin Kim, Yong Kyu Lee
Dept. of Computer Engineering, Dongguk University

요약

지금까지 XML 문서는 사용자에게 문서내의 모든 내용을 공개하였지만, 전자상거래와 같은 특정 분야의 경우에 사용자에 따라 문서의 일부만을 공개하는 것이 필요하다. 따라서, 본 논문에서는 사용자에게 XML 문서의 권한을 부여하고 접근 권한을 근간으로 XML 문서에 대해서 접근을 관리하는 접근 권한 관리 도구를 설계하고 구현한다. 권한 주체의 기본 단위로 사용자 그룹을 설정하고, XML 문서의 엘리먼트를 권한 객체의 기본 단위로 설정한다. 문서의 생성자는 문서를 생성할 때 일반 사용자 그룹에게 문서 내의 엘리먼트 권한을 부여한다. XML 문서를 접근할 경우, 문서의 접근에 대한 특정 사용자 그룹의 접근 권한을 검사하고 접근 권한에 맞는 문서의 특정 부분을 보여준다. 그 결과 XML 문서에 대한 접근 권한 관리가 가능하다.

1. 서론

XML[1] 문서는 자료의 공유 측면에서 문서의 모든 내용에 대하여 문서에 접근하는 사용자에게 제한 없이 공개하였다. 그렇지만, 예를 들어 전자상거래와 같은 특정 분야의 경우, 타인에게 공개하지 않아야 하는 개인정보에 대한 보안이 필요하고 이에 따라 문서에 대한 접근 권한 관리가 필요하다. 따라서, 문서를 생성할 때, 문서의 구조와 내용에 대해서 사용자들에게 접근 권한을 부여하고, 접근 권한을 소유하는 사용자만이 문서의 특정 내용에 접근 가능하도록 하기 위한 접근 권한 관리가 필요하다.

접근 권한에 대한 관련 연구로는 기존 객체 지향 데이터베이스에서의 권한 관리 기법[4][5]이 있다. 객체 지향 데이터베이스[4][5]에서의 권한 관리 형태에는 명시적(Explicit) 암시적(Implicit) 권한 부여, 긍정적(Positive) 부정적(Negative) 권한 부여, 강한(Strong) 약한(Weak) 권한 부여의 3가지 형태가 있다. 이러한 권한 부여 정보를 관리하기 위해 권한 부여 목록(Catalog)이 필요하며, 권한 부여를 수행하기 위해 기본적인 연산(Operation)들이 지원되어야 한다. 즉, 기본적인 권한 부여 연산에는 권한 부여가 명시적으로 목록에 저장되어 있는지, 또는 목록에 저장된 권한들로부터 권한 부여가 유추될 수 있는지를 결정하는 검사(Checking)연산과 권한 부여의 정보가 권한 부여 목록에 저장되도록 하는 부여(Granting)연산이 있다, 그리고 해당 정보를 목록으로부터 삭제하는 취소(Revoking)연산이 있다.

본 논문에서는 XML 문서의 접근 권한 및 접근 관리 시스템을 구현한다. 먼저 사용자의 접근 권한을 관리하기 위해서 사용자 그룹을 권한 주체의 기본 단위로 처리하고, 이를 근간으로 권한 주체에 대해서 검사, 부여, 취소의 권한 연산을 구현한다. XML 문서에 대한 접근 관리는 사용자가 XML 문서에 접근할 때, 사용자의 주체 계층과 해당 문서에 대한 권한을 근간으로 하여 필요한 내용만으로 구성된 문서를 생성하여 사용자에게 보여주도록 한다. 이를 위해서 접근 권한 관리 도구는 권한 주체에 따른 접근 권한들의 정보를 저장한 접근 권한 데이터베이스를 이용한다.

2. XML 문서의 접근 권한

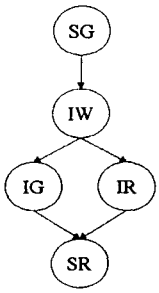
접근 권한을 정의하고 관리하기 위해서 권한 유형과 계층 구조, 권한의 주체 계층에 대해 서술한다.

2.1 권한 유형과 계층 구조

XML 문서의 권한 유형은 크게 두 가지로 XML 문서에 대한 권한 유형과 XML 문서 내의 엘리먼트에 대한 권한 유형이 있다.

(1) XML 스키마와 문서에 대한 권한 유형

- Schema Read(SR) : XML 문서의 정의 부분을 읽을 수 있다.
- Schema Generate(SG) : XML 문서의 정의 부분인 스키마 문서를 읽을 수 있고, 스키마 문서를 생성할 수 있다.

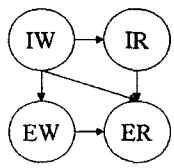


[그림 1] XML 문서에 대한 권한 유형의 계층 구조

• Instance Read(IR) : 스키마 문서와 인스턴스 문서를 읽을 수 있다.
• Instance Generate(IG) : 스키마 문서를 읽을 수 있고, 이를 근간으로 인스턴스 문서를 생성할 수 있다.
• Instance Write(IW) : 스키마 문서와 인스턴스 문서를 읽고 생성할 수 있을 뿐만 아니라, 생성한 인스턴스 문서에 대한 수정도 가능하다.

스키마 문서의 권한 유형과 인스턴스 문서에 대한 권한 유형은 상호 결합된 형태로 존재하고 이를 계층 구조로 표현할 수 있으며, 상위 권한 유형은 하위 권한 유형에 대해서 암시적 권한이 있다. [그림 1]은 스키마 문서와 인스턴스 문서의 권한 유형을 계층 구조로 나타낸 것이다.

(2) XML 문서 내의 엘리먼트에 대한 권한 유형



[그림 2] XML 문서 내의 엘리먼트 권한 유형의 계층 구조

• Element Read(ER) : 문서 내의 데이터를 읽을 수 있다.
• Element Write(EW) : 문서 내의 데이터를 읽고 수정할 수 있다.

문서 내의 엘리먼트에 대한 권한의 유형은 인스턴스 권한 유형과 상호 결합된 형태로 존재하고 이를 계층 구조로 표현할 수 있다. [그림 2]는 인스턴스와 엘리먼트 권한 유형의 계층 구조를 나타낸 것이다.

2.2 권한 주체와 계층 구조

동일한 접근 권한을 공유하는 하나 이상의 사용자를 권한 주체라고 한다. 권한 주체는 자신의 특정 기능이나 역할에 따라 세부적으로 구분하여 계층적으로 표현할 수 있으며, 상위 사용자는 하위 사용자들에 대한 모든 권한을 행사할 수 있다.

3. 사용자의 권한 관리

사용자의 권한 관리는 권한의 주체에 대해서 권한을 부여하거나 회수하는 것을 의미한다. 다음은 권한 주체와 생성한 문서를 관리하기 위한 데이터베이스의 설계와 권한 목록이 저장된 데이터베이스에 대한 연산을 설명한다.

3.1 권한 주체의 관리

권한의 관리를 위해서는 권한 연산의 대상인 권한 주체가 설정되어야 하는데 사용자 그룹을 기본으로 하며, 사용자는 하나 이상의 그룹에 속할 수 있다. 권한 주체와 권한 유형에 대한 모호성을 배제하고 권한의 상속을 명확하게 하기 위해서 트리 형태의 권한 주체 계층 구조를 사용한다. 다음의 [표 1]과 [표 2]는 권한 주체에 대한 데이터베이스 테이블들의 내용을 보여주고 있다.

[표 1] UserTable

필드	역 할
UID*	사용자 ID (기본키)
GID	그룹 ID
PWD	패스워드

[표 2] GroupAuthTable

필드	역 할
GID*	그룹 ID (기본키)
PGID	상위 부모 주체의 그룹 ID
Auth	권한 유형(SG, IW, IG, IR, SR)

3.2 접근 권한의 관리

접근 권한의 관리는 사용자 그룹의 XML 문서에 대한 접근 권한을 유지하고 사용자가 특정 XML 문서에 접근할 때 권한을 검사하기 위한 것이다. 다음은 [표 3], [표 4], [표 5], [표 6]은 이를 위해 필요한 데이터베이스 테이블의 내용을 설명하고 있다.

[표 3] SchemaTable

필 드	역 할
SID*	스키마 ID (기본키)
SDN	스키마 문서 이름
SGUID	스키마 생성자 ID
IE	인스턴스 문서 존재 유무(True, False)

[표 4] InstanceTable

필 드	역 할
IID*	인스턴스 ID (기본키)
IDN	인스턴스 문서 이름
SID	참조된 스키마 ID
IGUID	인스턴스 생성자 ID

[표 5] InstanceAuthTable

필 드	역 할
IID*	인스턴스 ID (기본키)
PerID	허가 그룹 ID
Auth	허가된 그룹의 권한

[표 6] ElementTable

필 드	역 할
IID*	인스턴스 ID (기본키)
EID*	엘리먼트 ID
PerID	허가 그룹 ID
Auth	허가된 그룹의 권한

3.3 권한 연산

권한의 연산은 접근 권한 데이터베이스를 이용하여 권한의 목록을 연산하는 것으로 검사, 부여, 취소가 있다.

권한의 검사는 사용자의 권한 유무를 판단하는 것으로 [그림 3]은 인스턴스 문서에 대한 권한 유형의 검사를 위한 알고리즘이다. 권한의 부여는 권한 주체의 객체에 대한 권한을 데이터베이스에 등록하는 것으로 [그림 4]는 인스턴스 문서에 대한 권한 부여 알고리즘이다. 마지막으로 권한의 취소는 권한 주체의 권한을 데이터베이스로부터 삭제하는 것이다. [그림 5]는 인스턴스 문서에 대한 권한 취소 알고리즘이다.

```
function auth_check(사용자 ID, 인스턴스 ID){
 UserTable에서 사용자 ID를 조사한다.
 if(사용자 ID가 존재할 경우){
   UserTable에서 사용자 ID에 대한 그룹 ID를 조사한다.
   InstanceTable에서 입력 받은 인스턴스 ID가 존재하는지
   조사한다.
   if(인스턴스 ID가 존재할 경우){
      InstanceTable과 InstanceAuthTable에서 그룹 ID에
      해당하는 인스턴스 ID의 권한 유무를 조사한다.
      return 그룹 ID와 권한 유형.
   }else{ 경고를 출력한다. }
 }else{ 경고를 출력한다. }
}
```

[그림 3] 인스턴스 문서의 권한 검사 알고리즘

```
function auth_grant(사용자 그룹 ID, 인스턴스 ID, 부여할
                    권한 유형){
  GroupAuthTable에서 입력받은 사용자 그룹 ID의 존재
  여부를 조사한다.
  if(그룹 ID가 존재할 경우){
     InstanceAuthTable과 ElementTable에서 허가 그룹 ID에
     사용자 그룹ID의 존재 여부를 조사한다.
     if(사용자 그룹 ID가 존재하지 않을 경우){
        InstanceAuthTable과 ElementTable에서 허가 그룹
        ID에 입력받은 권한 유형을 저장한다.
     }else{ 경고를 출력한다. }
  }else{ 경고를 출력한다. }
}
```

[그림 4] 인스턴스 문서의 권한 부여 알고리즘

```
function auth_revoke(사용자 그룹 ID, 인스턴스 ID){
  GroupAuthTable에서 입력받은 사용자 그룹 ID의 존재
  여부를 조사한다.
  if(그룹 ID가 존재할 경우){
     InstanceAuthTable에서 허가 그룹 ID에 사용자 그룹ID의
     존재 여부를 조사한다.
     if(사용자 그룹 ID가 존재할 경우){
        InstanceAuthTable과 ElementTable에서 허가 그룹
        ID에 입력받은 권한 유형을 삭제한다.
     }else{ 경고를 출력한다. }
  }else{ 경고를 출력한다. }
}
```

[그림 5] 인스턴스 문서의 권한 취소 알고리즘

3.4 스키마와 인스턴스 문서의 등록

스키마 문서와 인스턴스 문서의 생성자는 문서를 생성한 후 앞에서 설명한 방법을 이용하여 사용자들에 대한 접근 권한을 부여한다. 또한, 문서를 삭제할 때는 사용자들의 권한을 취소한다.

4. XML 문서의 접근 관리

XML 문서의 접근 관리는 사용자인 권한 주체가 XML 문서에 접근했을 때 이에 대한 접근을 관리하는 것으로 스키마 문서와 인스턴스 문서에 대한 접근 관리가 있다.

4.1 스키마 문서의 접근 관리

스키마 문서의 접근 관리는 스키마 문서의 읽기 역할을 관리한다. [그림 6]은 스키마 문서의 접근 관리에 대한 알고리즘을 보여주고 있다.

```
function schema_manage(사용자 ID, 스키마 ID){
 UserTable에서 사용자 ID를 조사한다.
 if(사용자 ID가 존재할 경우){
   사용자 ID에 해당하는 그룹 ID를 조사한다.
   if(그룹 ID가 존재할 경우){
      SchemaTable에서 입력받은 스키마 ID를 조사한다.
      if(스키마 ID가 존재할 경우){
         스키마 ID에 대해서 사용자가 속한 그룹 ID의
         권한을 검사한다.
         if(권한이 존재할 경우){
            스키마 문서를 출력한다.
         }else{ 경고를 출력한다. }
      }else{ 경고를 출력한다. }
   }else{ 경고를 출력한다. }
 }else{ 경고를 출력한다. }
}
```

[그림 6] 스키마 문서의 접근 관리 알고리즘

4.2 인스턴스 문서의 접근 관리

인스턴스 문서의 접근 관리는 스키마 문서를 참조하여 인스턴스 문서의 읽기와 문서 내의 엘리먼트에 대한 수정을 처리한다. [그림 7]은 인스턴스 문서의 접근 관리에 대한 알고리즘을 보여 주고 있다.

```
function instance_manage(사용자 ID, 인스턴스 ID){
  auth_check(사용자 ID, 인스턴스 ID).
  읽기 또는 수정의 역할을 입력받는다.
  if(읽기 역할일 경우){
    InstanceTable, InstanceAuthTable, ElementTable에서
    사용자 ID가 속한 그룹 ID에 해당하는 인스턴스 문서를
    출력한다.
  }else{ // 수정 역할일 경우
    기존 엘리먼트 ID와 수정할 엘리먼트 ID를 입력받는다.
    ElementTable에서 입력받은 기존 엘리먼트 ID의
    존재여부를 검사한다.
    if(기존 엘리먼트 ID가 존재할 경우){
      ElementTable에서 엘리먼트 ID를 수정한다.
    }else{ 경고를 출력한다. }
  }
}
```

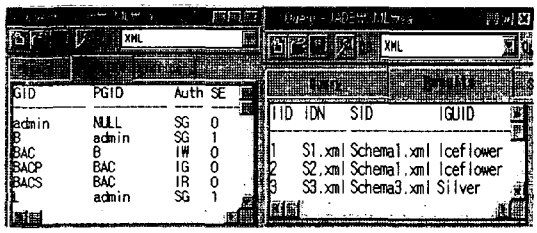
[그림 7] 인스턴스 문서의 접근 관리 알고리즘

5. 구현

본 연구에서는 XML-Data[3]를 이용하여 XML 문서의 접근 권한 관리 시스템과 XML문서의 접근 관리 시스템을 구현하였다. 시스템 구현 도구로는 Windows/NT 4.0 운영체제에서 마이크로소프트사의 XML 파서, ASP(Active Server Pages), DOM(Document Object Model)[4], 인터넷 익스플러러 5.0, Visual C++ 5.0, 그리고 MS-SQL 6.5를 이용하였다. 그리고, 개발된 시스템을 대학의 성적 관리에 적용하였다.

5.1 사용자 권한 관리의 구현

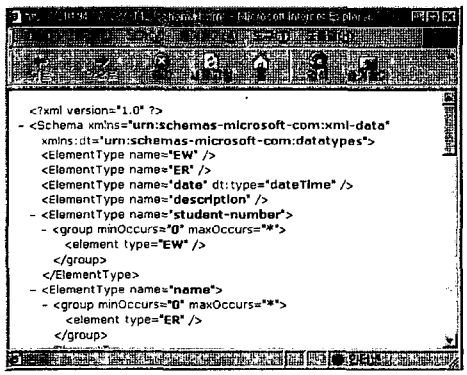
권한 주체로 대학의 구성원들을 트리 형태로 설정하였다. [그림 8]은 GroupAuthTable과 InstanceTable의 일부분이다.



[그림 8] GroupAuthTable과 InstanceTable의 예

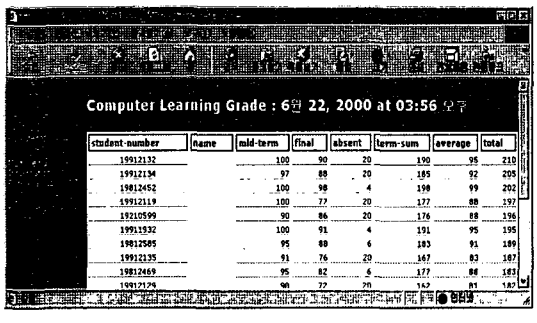
5.2 XML 문서 접근 관리의 구현

XML 문서의 접근 관리는 사용자인 주체 계층에 대한 인증 단계부터 시작하여 스키마 문서에 대한 처리와 인스턴스에 대한 처리의 두 부분으로 나누어 구현하였다. 접근 관리 시스템의 주 화면에서 스키마 문서를 선택하면 권한이 있을 경우 [그림 9]처럼 읽을 수 있다.



[그림 9] 스키마 읽기 화면

주 화면에서 인스턴스 문서를 선택하면 [그림 10]과 같이 인스턴스 문서를 읽을 수 있다.



[그림 10] 인스턴스 읽기 화면

6. 결론 및 향후연구

본 논문은 XML 문서의 접근 권한과 웹에서의 접근을 관리 할 수 있는 시스템을 설계하고 구현하였다. 먼저 사용자의 계층 구조를 트리 형태로 정의하고, 각 사용자 그룹에 대하여 XML 문서의 엘리먼트까지 접근 권한을 부여한 후, 이 정보를 이용하여 웹에서 문서를 접근할 때 사용자를 확인하여 권한에 따라 문서의 해당 부분만을 읽을 수 있도록 하였다.

이렇게 구현한 접근 권한 관리 도구를 대학의 학생 성적 관리에 적용하였다. 적용 결과 사용자의 권한에 따라서 XML 문서에서 특정 부분을 보호할 수 있었으며, 사용자의 권한에 따라 필요로 하는 문서를 자동 생성할 수 있었으므로 각각의 문서들을 별도로 작성하는 것보다 효율적이었음을 확인할 수 있었다.

향후에는 트리 구조의 권한 주체 계층을 그래프 구조로 확장하는 연구가 필요하다.

참고문헌

[1] Tim Bray and Jean Paoli, "Extensible Markup Language(XML) 1.0, " http://www.w3c.org/TR/1998/REC-19980210.html, 1999.

[2] Andrew Layman, Edward Jung, Eve Maler, Henry S. Thompson, Jean Paoli, John Tigue, Norbert H. Mikula, Steve De Rose, "XML-Data," http://www.w3.org/TR/1998/NOTE-XML-data-0105/, 1998.

[3] Lauren Wood and Vidur Apparao, "Document Object Model(DOM) Level 1 Specification," http : // www.w3.org/TR/-REC DOM Level-1, 1999.

[4] Bertino and Martino, *Object-Oriented Database Systems*, Addison Wesley, 1993.

[5] Won-Kim, *Introduction to Object-Oriented Databases*, MIT Press, 1994.