웹에서의 XML 문서 접근 제어 시스템의 설계 및 구현

이 용 규†

요 약

지금까지 XML 문서는 사용자에게 문서내의 모든 내용을 공개하였지만, 전자상거래와 같은 특정 분야의 경우에는 사용자에 따라 문서의 일부만을 공개하는 것이 필요하다. 따라서, 본 논문에서는 사용자에게 XML 문서의 권한을 부여하고 접근 권한을 근간으로 XML 문서에 대해서 접근을 관리하는 접근 제어 시스템을 설계하고 구현한다. 이를 위하여 권한 주체의 기본 단위로 사용자 그룹을 설정하고, XML 문서의 엘리먼트를 권한 객체의 기본 단위로 설정한다. 그리고, XML 문서의 생성자는 문서를 생성할 때 사용자 그룹에게 문서 내의 엘리먼트에 대한 접근 권한을 부여한다. 사용자가 XML 문서를 접근할 경우, 사용자가 속한 그룹의 해당 문서에 대한 접근 권한을 접근 제어 리스트를 이용하여 검사하고 접근 권한에 맞는 문서의 특정 부분만을 보여준다. 그 결과 XML 문서에 대한 접근 관리가 가능하다.

Design and Implementation of an Access Control System for XML Documents on the Web

Yong-Kyu Lee†

ABSTRACT

Until now the XML document is allowed users to access the whole content of it. However, for some applications such as those in the field of electronic commerce, there are cases that the whole content should not be delivered. Therefore, access authorization is required for XML documents in order to protect illegal accesses to some critical parts of them. In this paper, we design and implement a system which authorizes users to XML documents and controls access to them based on the access rights. We set the user group as a basic unit of the authorization subject and the element of an XML document as a basic unit of authorization object. The owner of a document authorizes user groups to access the elements of it. When an XML document is accessed, the access rights of the requester are checked using an access control list and only the authorized parts are delivered. As the result, we can authorize XML documents, which has been previously impossible.

1. 서 론

웹에서 XML(eXtensible Markup Language)[1, 2] 문서는 HTML과 SGML[3]에서와 같이 자료의 공유 측면에서 문서의 모든 내용에 대하여 문서에 접근하는 사용자에게 제한없이 공개한다. 이 경우에 일반적인 문서들은 자료의 공유 측면에서 사용자에게 모든 정보를 전달하는 이점이 있다. 그렇지만, 예를 들어 전자상거래와 같은 특정 분야의 경우, 타인에게 공개하지 않아야 하는 개인 정보 등에 대한 보안이 필요하며, 이에 따라 문서에 대한 접근 관리가 요구된다. 따라서, 문서를 생성할 때, 문서의 내용에 대해서 사용자들에게 접근 권한을 부여하고, 접근 권한을 소유하는 사용자만이 문서의 특정 내용에 접근 가능하도록 하기 위한 접근 권한 관리가 필요하다. 즉, 기존의 데이터베이스에서 사용자에게 접근 권한을 부여하는 것과 마찬가지로 XML 문서를 대상으로 한 검색에 있어서도 사용자가 XML 문서의 특정 항목에만 접근할 수 있도록

† 종신회원 : 동국대학교 컴퓨터멀티미디어공학과 교수
논문접수 : 2000년 10월 30일, 심사완료 : 2000년 12월 29일

권한을 부여함으로써, 사용자의 권한 범위에 해당하는 데이터만을 제공할 수 있도록 하는 기능이 필요하다. 이를 위해서는 사용자의 접근 권한을 관리할 수 있어야 하며, 사용자가 XML 문서에 접근할 때 권한에 따라 제어할 수 있어야 한다. 쉬운 방법은 권한에 따라 사용자 별로 별도의 문서를 보유하는 것이나, 이는 기억 장소의 낭비와 문서의 중복에 따른 문서 변경의 어려움 등의 문제점이 있으므로 하나의 문서로부터 사용자의 권한에 따라 권한이 없는 부분은 제외하고 접근이 허용된 부분만을 전달하는 것이 필요하다.

접근 관리에 대한 관련 연구로는 객체 지향 데이터베이스의 권한 관리 모델[4, 5]이 있다. 객체 지향 데이터베이스의 모델은 접근의 대상인 객체와 객체에 대한 권한을 공유하는 사용자 주체 그리고, 주체가 객체에 대해서 행사할 수 있는 권한 유형의 기본적 구성 요소를 갖는다. 그리고, 각각의 권한 주체들에 대하여 권한 연산을 통해서 권한을 관리하고, 권한에 따라 사용자의 객체에 대한 접근을 제어한다. 최근의 XML 스키마 모델[6-10]은 XML 문서를 데이터로 보는 관점에서 객체 지향 개념을 적용하여 XML의 엘리먼트를 문서 구조의 기본 단위인 객체로 표현한다. 따라서, 객체 지향 데이터베이스의 접근 권한 모델을 응용한다면 XML 문서의 접근 권한 관리가 가능하다. 그러나, 데이터베이스에서는 클래스의 인스턴스에 대한 접근 관리가 일반적인데 반하여, XML은 데이터의 중복을 최소화하기 위해서는 문서 인스턴스내의 엘리먼트에 대한 접근 관리가 필요하다.

본 논문에서는 XML 문서의 접근 권한 및 접근 관리 시스템을 구현한다. 먼저 객체 지향 데이터베이스에서의 접근 권한 모델을 응용하여 XML 문서에 대한 접근 권한 유형과 유형들 간의 계층 구조, 그리고 권한 주체와 주체들 간의 계층 구조를 정의한다. 권한 유형과 권한 주체의 계층 구조는 DAG(Directed Acyclic Graph) 형태를 갖도록 한다. 이와 같은 정의를 토대로 접근 관리 시스템을 구현하기 위해서 사용자에 대한 권한 관리와 XML 문서에 대한 접근 관리의 두 부분으로 나누어 처리한다. 사용자의 권한을 관리하기 위해서 사용자 그룹을 권한 주체의 기본 단위로 설정하고, 권한 주체별 접근 권한을 접근 제어 리스트에 저장하여 권한 주체에 대해 권한의 검사, 부여, 취소의 권한 연산을 처리한다. 여기서 접근의 기본 단위를 설정하여야 하는데, 문서 인스턴스 단위로 설정하면 구현이 쉬우나 인스턴스 내의 엘리먼트에 대한 접근 관리가 불가능하므로 우리는 엘리먼트를 접근 관리의 기본 단위로 설정한다. 그러나 권한 주체마다 모든 접근 권한을 표시해야 하기 때문에 엘리먼트 단위의 접근 관리를 위해서는 접근 제어 리스트의 크기가 커지게 되는 문제점이 발생한다. 이를 해결하기 위해서 본 논문에서는 DAG를 완전 k-ary 트리로 변환하고 접근 권한을 부모와 자식 노드 간에 상속받을 수 있도록 함으로써, 접근 제어 리스트의 크기를 상당히 줄이면서도 권한 주체의 접근 권한을 빨리 알아낼 수 있는 새로운 방법을 제시한다.

그리고, XML 문서에 대한 접근 관리는 데이터의 구조를 정의한 스키마 문서와 실제 데이터를 표현한 인스턴스 문서로 나누어 관리한다. 사용자가 XML 문서에 접근할 때, 권한 주체의 스키마 문서와 인스턴스 문서에 대한 접근 권한을 저장한 접근 제어 리스트를 사용하여 접근이 허용된 내용만을 사용자에게 보여주도록 처리한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구에 대해서 서술하고, 3절에서는 XML 문서의 권한 유형에 대하여 정의한다. 4절에서는 접근 권한 관리를 기술하고, 5절에서는 XML 문서의 접근 관리에 대하여 설명한다. 그리고, 6절에서는 성능 평가 결과를 제시하고, 7절에서는 구현 결과에 대해서 기술하며, 마지막으로 8절에서는 결론 및 향후 연구에 대해 서술한다.

2. 관련 연구

본 절에서는 객체 지향 데이터베이스에서의 접근 권한 모델[4, 5]과 XML DTD(Document Type Definition)의 확장 개념으로 개발된 XML 스키마 모델[6-10], 그리고 접근 제어 기법[11]에 대하여 기술한다.

2.1 객체 지향 데이터베이스의 권한 관리

객체 지향 데이터베이스의 권한 관리에서는 권한 주체, 접근의 대상인 객체 그리고 권한 연산의 기본 요소들이 존재한다. 여기서 권한 주체는 접근 권한에 따라 데이터 객체에 대하여 접근하는 사용자를 말하며 권한 주체들 간에는 권한의 포함 관계에 따라 DAG 구조가 존재한다. 즉, DAG 구조상에서 상위의 권한 주체는 하위 권한 주체의 권한을 모두 소유하게 된다. 그리고 데이터 객체에 대한 접근 관리를 위해서 권한 주

체별 접근 권한을 목록에 저장하고, 목록을 이용하여 권한의 처리를 하는 권한 연산들이 지원된다. 권한 연산에는 특정 주체의 객체에 대한 권한이 명시적으로 목록에 저장되어 있는지 또는 목록으로부터 유추할 수 있는지를 결정하는 검사 연산과 권한을 목록에 추가하는 부여 연산이 있으며, 권한을 삭제하는 취소 연산이 있다. 따라서 이와 같은 목록과 연산을 이용하여 사용자 주체별로 데이터 객체에 대한 접근 관리가 가능하다.

2.2 XML 스키마

객체 지향의 관점에서 XML 문서를 독립적인 객체로 하여 문서 내의 구성 요소들을 구조적으로 명확하게 표현하려는 연구가 진행중이다. XML에 대한 표준을 주도하는 W3C에서는 객체 지향 개념을 적용한 XML-Data[9]와 SOX(Schema for Object-oriented XML)[10]를 제안하고 있고, 두 제안은 공통적으로 스키마 모델을 적용하고 있다. 스키마 모델은 기존의 DTD 작성시의 어려움과 복잡성을 극복하고 객체 지향 개념의 상속성, 재사용성 등의 효과를 얻기 위해 도입되었으며, XML의 문서 구조를 나타내기 위하여 DTD 대신 데이터베이스에서 사용하는 스키마를 응용한 것이다.

XML 스키마에서는 XML 문서와 문서내의 엘리먼트를 객체로 취급하므로 객체 지향 데이터베이스에서의 객체와 동일한 관점으로 발전하고 있다. 따라서 객체 지향 데이터베이스에서 객체에 대한 권한 관리가 필요한 것처럼 XML에서도 특정 스키마에 따르는 인스턴스 문서와 문서내의 엘리먼트에 대한 권한 관리가 필요하다. 다만 데이터베이스에서는 클래스의 객체가 접근 관리의 기본 단위인데 반하여, XML에서는 문서 인스턴스 내의 엘리먼트를 접근관리의 기본단위로 하여야 권한에 따라 한 문서의 여러 본을 관리하지 않아도 된다.

2.3 접근 제어 기법

보안을 필요로 하는 정보에 대한 접근 제어 정책은 권한의 소유자에 의해 객체에 대한 접근 권한이 정해지는 임의적 접근 제어(DAC : Discretionary Access Control), 정보의 보안 수준과 사용자의 보안 등급에 따라 접근을 제어하는 강제적 접근 제어(MAC : Mandatory Access Control), 그리고 역할과 해당 역할의 권한을 정의하는 역할 기반 접근 제어(RBAC : Role-Based Access Control)로 나눌 수 있다[11]. 또한 이러한 정책을 실현하기 위한 접근 제어 메카니즘으로는 권한 주체와 권한 객체사이의 권한 관계를 매트릭스로 표현하는 접근 제어 매트릭스(Access Control Matrix), 권한 주체에 대해 권한이 부여된 객체들의 리스트로 표현하는 접근 제어 리스트(Access Control List), 그리고 권한 객체에 대해 접근이 허용된 주체들의 리스트로 표현하는 자격 리스트(Capability List) 등이 있다. 본 논문에서는 이들 중 사용자에 대한 권한 관리가 편리한 접근 제어 리스트를 XML 문서의 접근 제어에 활용하고자 한다.

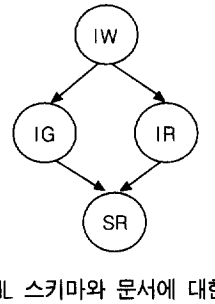
3. XML 문서의 권한 유형

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

3.1 XML 스키마와 문서에 대한 권한 유형

- Schema Read(SR) : XML 문서의 정의 부분을 읽을 수 있다.
- Instance Read(IR) : 스키마 문서와 인스턴스 문서를 읽을 수 있다.
- Instance Generate(IG) : 스키마 문서를 읽을 수 있으며, 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 있다. (그림 1)은 본 논문에서 정의



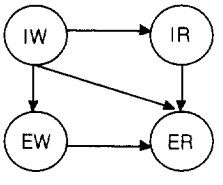
(그림 1) XML 스키마와 문서에 대한 권한 유형

한 스키마 문서와 인스턴스 문서의 권한 유형을 계층 구조로 나타낸 것이며, 그림에서 IW 권한의 소유자는 IG와 IR 권한도 묵시적으로 소유하는 것을 보여주고 있다.

3.2 XML 문서 내의 엘리먼트에 대한 권한 유형

- Element Read(ER) : 엘리먼트의 데이터를 읽을 수 있다.
- Element Write(EW) : 엘리먼트의 데이터를 읽 고 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. 여기서도 상위의 권한 유형은 하위의 권한 유형에 대한 묵시적 권한을 보유한다.



(그림 2) XML 문서내의 엘리먼트 권한 유형의 계층 구조

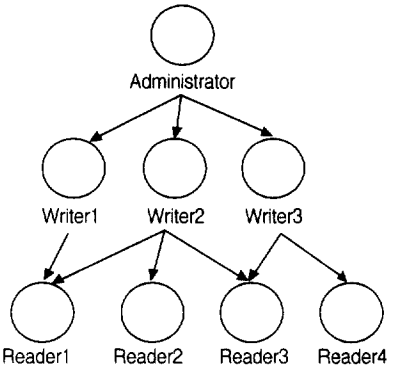
4. 접근 권한 관리

사용자의 권한 관리는 권한의 주체에 대해서 권한을 부여하거나 회수하는 것을 의미한다. 다음은 사용자의 권한을 관리하기 위해 필요한 권한 주체와 접근 권한의 관리 그리고 권한 연산에 대하여 설명한다.

4.1 권한 주체의 관리

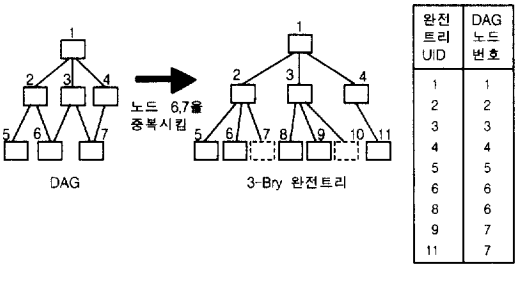
동일한 접근 권한을 공유하는 사용자 그룹을 권한 주체라고 한다. 권한 주체는 특정 기능이나 역할에 따라 세부적으로 구분하여 계층적으로 표현할 수 있으며, 상위 주체는 하위 주체들이 소유한 모든 권한을 행사할 수 있다. (그림 3)은 권한 주체 계층의 한 예로 권한 유형에 해당하는 주체의 역할을 계층적으로 나타낸 그림이다.

(그림 3)에서 보는 것처럼 권한 주체의 계층 구조는 DAG 형태를 갖는다. 여기서 어떤 노드의 부모 노드는



(그림 3) 권한 주체 계층의 예

자식 노드의 모든 권한을 묵시적으로 행사할 수 있으므로 접근 권한을 중복하여 기술하지 않고 문서에 대한 접근 시 권한 연산을 빨리 수행할 수 있도록 DAG를 완전(complete) k-ary 트리[12]로 변환하여 노드마다 UID(Unique ID)를 부여한다. 이렇게 하면 노드들 간의 연관 관계를 별도로 기록하여 놓지 않아도 완전 트리의 특성을 이용하여 알 수 있게 된다. (그림 4)는 이를 설명하고 있는데, DAG의 공유 노드는 완전 트리에서 중복되어 나타나며 점선으로 표시된 노드는 가상의 노드이다. 그림에서 DAG의 노드 6과 7이 완전 트리에서 중복되는 것을 보여주고 있으며, 표는 DAG와 완전 트리의 노드들의 대응 관계를 표시하고 있다.



완전 트리 UID	DAG 노드 번호
1	1
2	2
3	3
4	4
5	5
6	6
8	6
9	7
11	7

(그림 4) DAG의 k-ary 완전 트리로의 변환

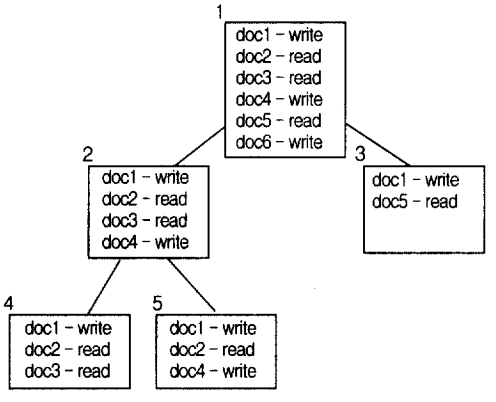
이와 같이 UID를 부여하면 노드 i의 부모 노드와 j 번째 자식 노드의 UID는 다음의 식 (1)과 식 (2)에 의해 바로 알 수 있게 된다.

$$\text{노드 } i\text{의 부모 노드 : } Parent(i) = \left\lfloor \frac{(i-2)}{k} + 1 \right\rfloor \quad (1)$$

$$\text{노드 } i\text{의 } j\text{번째 자식 노드 : } Child(i,j) = k(i-1) + j + 1 \quad (2)$$

(주 : 기호 ㄴ n ㄱ은 실수 n보다 크지 않은 최대의 정수를 나타냄)

여기서 어느 노드가 소유하는 접근 권한을 모두 기록할 필요없이 모든 자식 노드들이 공유하는 권한은 부모 노드에만 기록하여 이로부터 상속받을 수 있도록

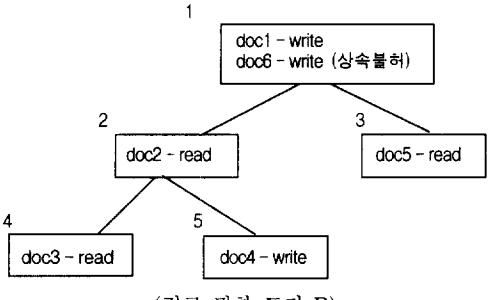


(접근 권한 트리 A)

문서ID	접근 허용 권한 주체 ID
doc 1	1(w), 2(w), 3(w), 4(w), 5(w)
doc 2	1(r), 2(r), 4(r), 5(r)
doc 3	1(r), 2(r), 4(r)
doc 4	1(w), 2(w), 5(w)
doc 5	1(r), 3(r)
doc 6	1(w)

(CTAL)

(그림 5) 권한의 중복 등록



(접근 권한 트리 B)

문서 ID	접근 허용 권한 주체 ID
doc 1	1(w)
doc 2	2(r)
doc 3	4(r)
doc 4	5(w)
doc 5	3(r)
doc 6	1(w, 상속 불허)

(IAL)

(그림 6) 권한의 상속

하면 접근 제어 리스트의 크기를 많이 줄일 수 있다. 예를 들어 (그림 5)와 같은 권한 계층 구조와 이에 따른 접근 제어 리스트 CTAL(Complete Tree Access List)은 (그림 6)의 권한 트리와 접근 제어 리스트 IAL (Inherited Access List)과 같이 단순화 할 수 있다. 그러나 자식 노드에게 상속되지 않는 권한은 상속 불허를 표시하여야 한다. 즉, 권한 트리 B의 노드 1의 'doc6'에 대한 권한은 상속이 불허됨을 나타낸다.

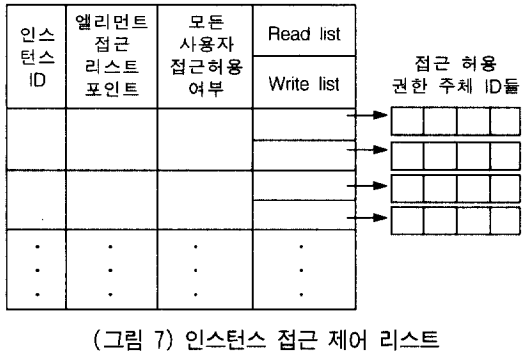
(그림 6)에서 어느 노드의 접근 권한은 다음의 식 (3)에 의해 구할 수 있다.

$$\text{접근 권한}(i) = \text{권한}(i) \cup \text{상속 가능 권한}(\text{조상 노드 들}) \cup \text{권한}(\text{자손 노드들}) \quad (3)$$

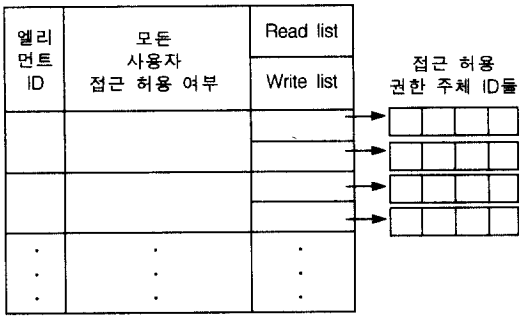
4.2 접근 권한의 관리

접근 권한의 관리는 사용자 그룹의 XML 문서에 대한 접근 권한을 유지하고 사용자가 특정 XML 문서에 접근할 때 권한을 검사하기 위한 것이다. 사용자에게 XML 문서의 엘리먼트 단위까지 접근을 관리하기 위해서는 문서 인스턴트에 대한 접근 제어 리스트가 필요하며 엘리먼트에 대해서도 접근 제어 리스트의 관리가 필요하다. 즉, 앞에서 설명한 권한 트리를 접근 제어 리스트 IAL의 형태로 바꾸어 권한을 관리한다. 그러나 여기서 접근 관리가 필요한 인스턴스와 엘리먼트들만을 관리하면 된다. (그림 7)과 (그림 8)은 인스턴스 접근 제어 리스트와 엘리멘트 접근 제어 리스트를 설명하고 있다.

(그림 7)과 (그림 8)에서 모든 사용자에게 접근이 허용된 경우는 read list와 write list가 필요 없으며, '모든 사용자 접근 허용 여부' 필드는 이를 표시하기 위한 것이다.



(그림 7) 인스턴스 접근 제어 리스트



(그림 8) 엘리먼트 접근 제어 리스트

4.3 권한 연산

권한의 연산은 앞에서 정의된 접근 권한 주체와 접근 제어 리스트를 이용하여 권한의 목록을 조사하거나 갱신하는 것으로 권한 연산의 종류로는 검사, 부여, 취소의 세 가지가 있다. 권한의 검사는 사용자의 권한 유무를 판단하는 것이고, 권한의 부여는 권한 주체의 객체에 대한 권한을 등록하는 것으로 (그림 9)는 인스턴스 문서에 대한 단순화된 권한 부여 알고리즘이며, 엘리먼트에 대해서도 마찬가지로 방법으로 부여한다. 마지막으로 권한의 취소는 권한 주체의 권한을 삭제하는 것이다.

```
function auth_grant(권한 주체 ID, 인스턴스 ID, 부여할
                    권한 유형){
GroupAuthTable에서 입력받은 권한 주체 ID의 존재
여부를 조사한다.
  if(권한 주체 ID가 존재할 경우){
    InstanceAuthList의 허가 그룹에서 권한 주체 ID의
    존재 여부를 조사한다.
    if(권한 주체 ID가 존재하지 않을 경우){
      InstanceAuthList에 권한 주체 ID와 입력받은 권한
      유형을 저장한다.
    }else{ 경고를 출력한다. }
  }else{ 경고를 출력한다. }
}
```

(그림 9) 인스턴스 문서의 권한 부여 알고리즘

4.4 인스턴스 문서의 등록

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

5. 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. 스키마 문서의 접근 관리는 스키마 문서의 읽기 역할을 관리하며, 인스턴스 문서의 접근 관리는 스키마 문서를 참조하여 인스턴스 문서의 읽기와 문서 내의 엘리먼트에 대한 수정을 처리한다. (그림 10)은 인스턴스 문서의 접근 관리에 대한 간략화된 알고리즘을 보여주고 있다.

```
function instance_read(권한 주체 ID, 인스턴스 ID){
  인스턴스 접근 제어 리스트로부터 권한 주체의 읽기
  권한을 조사한다.
  if(읽기 권한을 갖고 있으면){
    엘리먼트 접근 제어 리스트로부터 읽기 권한이 있는
    엘리먼트 ID들을 조사한다.

    읽기 권한이 있는 엘리먼트들로 문서를 생성하여
    출력한다.
  }
  else{ 경고를 출력한다. }
```

(그림 10) 인스턴스 문서의 읽기 알고리즘

6. 성능 평가

접근 제어 리스트를 실제로 적용할 경우 권한 주체와 권한 객체의 수가 많아짐에 따라 접근 제어 리스트의 크기가 매우 커지게 된다. 따라서, 이 절에서는 접근 제어 리스트 CTAL(Complete Tree Access List)과 IAL(Inherited Access List)의 크기를 비교 평가한다.

6.1 접근 제어 리스트의 크기 분석

접근 제어 리스트에서 필요로 하는 기억 장소의 크기를 알기 위해서는 접근 제어 리스트에서 관리하여야 할 문서별 접근 허용 주체의 수를 알아야 한다. 이는 앞의 (그림 5)와 (그림 6)에서 본 것처럼 접근 권한 트리의 (문서 ID, 접근 권한) 쌍의 수와 일치한다.

먼저, (그림 5)에서 설명한 권한 트리 A의 높이를 h라 하고, 분석을 쉽게 하기 위하여 full k-ary 트리라 가정하며, 자식 노드의 어느 권한이 부모 노드에서 기록되는 비율을 u라 한다. 또한 (그림 6)의 상속이 불허되는 권한은 두 방법 모두 동일하므로 분석에서 생략하기로 한다.

권한 트리 A에서 리프 노드의 평균 권한의 수를 m

이라 하면, 트리의 레벨 $(h-1)$의 권한의 수는 $k*m*u$가 된다. 이를 일반화 하면 레벨 i의 어느 노드의 권한의 수는 $k^{h-i}*m*u^{h-i}$이다. 또한, 레벨 i의 노드의 수는 k^{i-1}이므로 레벨 i의 모든 권한의 수의 합은 $k^{h-1}*m*u^{h-i}$이 된다. 따라서, 이 트리의 모든 권한의 수는 다음과 같다.

$$\sum_{i=1}^{h} k^{h-1}*m*u^{h-i} \qquad (4)$$

이제 (그림 6)의 권한 트리 B의 경우를 분석하기로 한다. 이를 위해 자식 노드의 권한이 부모 노드로 승격되게 될 확률을 p라 하자. 그러면 어느 리프 노드의 권한의 수는 $m-m*p$가 된다. 그리고, 레벨 i의 어느 노드의 권한의 수는 $m*p^{h-i}-m*p^{h-i+1}$이 되고, 레벨 i의 모든 권한의 수는 $k^{i-1}*m*p^{h-i}-k^{i-1}*m*p^{h-i+1}$이 된다. 또한, 루트 노드의 권한의 수는 $m*p^{h-1}$이므로, 트리 B의 전체 권한의 수는 다음과 같다.

$$\sum_{i=2}^{h}(k^{i-1}*m*p^{h-i}-k^{i-1}*m*p^{h-i+1})+m*p^{h-1}$$
$$= k^{h-1}*m-m*\sum_{i=1}^{h-1} p^{i}*(k^{h-i}-k^{h-i-1}) \qquad (5)$$

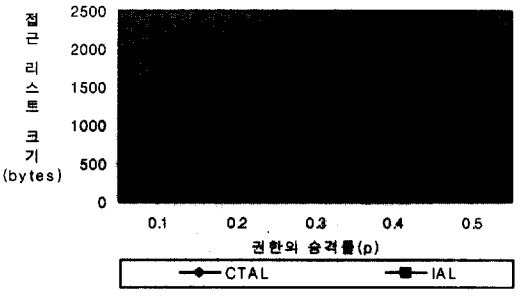
6.2 접근 제어 리스트의 크기 분석 결과

접근 제어 리스트 CTAL과 IAL의 크기를 비교하기 위하여 앞 절의 식과 <표 1>의 파라미터 값을 이용하였다.

〈표 1〉 파라미터의 값

파라미터	값
문서의 수	100
문서 ID	4bytes
권한주체 ID	4bytes
k	3
h	5
m	5
u	0.5

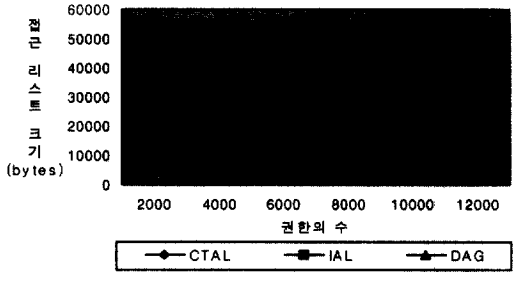
(그림 11)은 CTAL과 IAL의 기억 공간의 크기를 보여 주고 있다. 그림에서 보듯이 권한의 승격률(p)이 커질수록 IAL이 CTAL에 비하여 더욱 우수함을 알 수 있다.



(그림 11) 접근 제어 리스트의 크기 비교

6.3 접근 제어 리스트 크기 실험

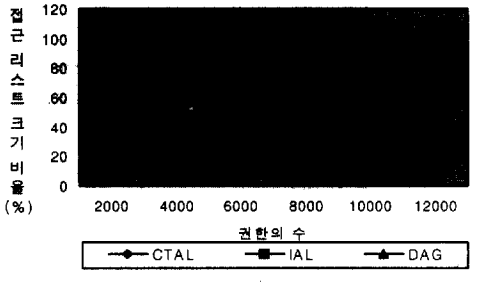
CTAL과 IAL의 크기를 DAG를 사용한 접근 제어 리스트와 보다 정확히 비교하기 위하여 실험을 실시하였다. 먼저, 문서의 수와 ID의 크기 등의 파라미터 값은 (m과 u 제외) <표 1>과 동일하게 설정하였으며, 난수 발생기를 이용하여 필요한 개수의 (문서 ID, 권한 주체 ID) 쌍을 임의로 10회 생성한 후 접근 제어 리스트의 평균 크기를 구하였다. (그림 12)는 권한의 수에 따른 접근 제어 리스트 크기의 변화를 보여주고 있으며, CTAL 보다는 DAG를 사용한 접근 제어 리스트가, DAG 보다는 IAL이 더 우수함을 보여주고 있다. 특히 권한의 수가 많아질수록 IAL의 크기가 약간씩 증가하다 일정 시점 후에 오히려 줄어드는 현상은 권한의 대부분을 상속에 의하여 처리할 수 있기 때문이다. 이 결과는 DAG를 트리로 변환할 때 전체 리프 노드 중 20%가 중복되었다고 가정한 결과이며, 다른 파라미터 값(10%, 30%, 40%)에 대한 실험에서도 유사한 결과를 얻었다.



(그림 12) 접근 제어 리스트의 크기 비교

(그림 13)은 CTAL의 크기를 100%로 하였을 때 DAG를 사용한 접근 제어 리스트와 IAL에서 필요로

하는 공간의 비율을 차트로 표현한 것이다. 여기서도 IAL의 우수함을 확인할 수 있다.



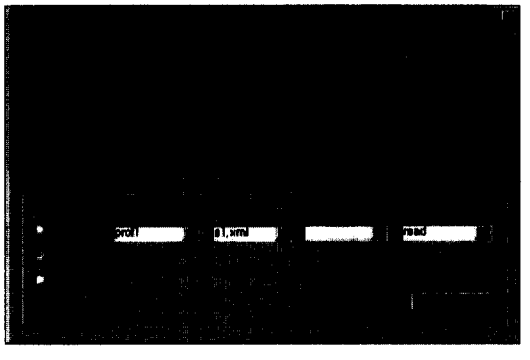
(그림 13) 접근 제어 리스트의 크기 비율

7. 구 현

본 연구에서는 XML 문서의 접근 권한 관리 시스템과 XML 문서의 접근 관리 시스템을 구현하였다. 시스템 구현 도구로는 Windows/NT 4.0 운영체제에서 마이크로소프트사의 XML 파서, XML-Data[10], ASP(Active Server Pages), DOM(Document Object Model)[13], 인터넷 익스플러러 5.0, Visual C++ 6.0을 이용하였다. 그리고 개발된 시스템을 대학의 성적 관리에 적용하였다.

7.1 사용자 권한 관리의 구현

사용자 권한 관리는 사용자인 권한 주체와 XML 문서에 대한 권한을 관리하는 것으로 사용자를 포함한 권한 주체의 등록과 XML 문서의 등록이 있고, 등록된 권한 주체와 XML 문서에 대한 접근 권한 연산으로 구성된다. (그림 14)는 권한 연산 화면을 보여주고 있다.



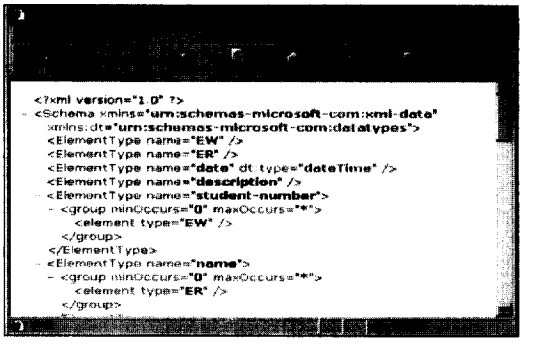
(그림 14) 권한 연산 화면

7.2 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. 사용자는 인증을 마친 후, 접근을 원하는 스키마 문서 또는 인스턴스 문서를 선택한다.

7.2.1 스키마 문서의 접근 관리 구현

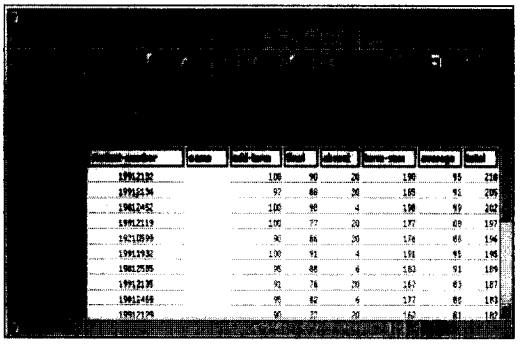
스키마 문서를 선택했을 경우에는 등록된 스키마 문서를 읽거나 삭제할 수 있도록 구현하였으며, 스키마의 삭제는 해당 인스턴스가 존재하지 않을 때만 삭제하도록 하였다. 스키마 문서의 읽기는 선택한 스키마 문서를 브라우저로 보여준다. (그림 15)는 스키마 문서의 읽기 화면을 보여주고 있다.



(그림 15) 스키마 문서의 읽기 화면

7.2.2 인스턴스 문서의 접근 관리 구현

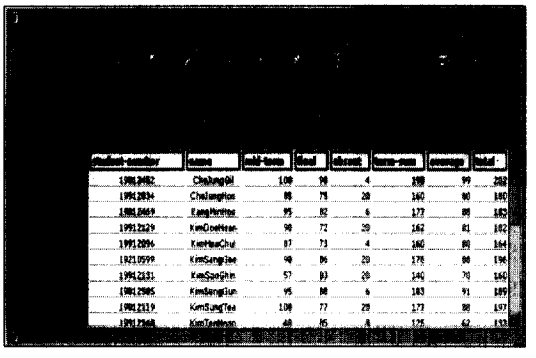
인스턴스 문서의 접근 관리에서는 사용자가 접근이 허용된 인스턴스 문서를 읽거나 수정, 또는 삭제할 수 있다. (그림 16)은 컴퓨터공학과에 속한 사용자가 'sl.



(그림 16) 'name' 엘리먼트의 데이터를 숨긴 화면

xml' 문서의 읽기 기능을 선택한 화면으로 'name' 엘리먼트에 대해서 읽기 권한이 존재하지 않는 경우를 보여주고 있다.

그러나, (그림 17)은 's1.xml' 문서내의 모든 엘리먼트를 읽을 수 있는 권한을 가진 사용자에 해당하는 화면을 보여주고 있다.



(그림 17) 모든 엘리먼트의 데이터를 보여준 화면

7.3 구현 결과 분석

본 논문에서는 앞 절에서 제안한 XML 문서를 위한 접근 제어 리스트 IAL과 이의 비교를 위해 CTAL을 구현하였다. 구현 결과, 앞 절에서의 성능 평가 결과와 동일하게 권한 주체와 권한 객체간에 허용된 권한의 수가 증가할수록 IAL의 성능이 CTAL보다 더욱 우수함을 확인할 수 있었다. 따라서 본 논문에서 제시한 방법을 접근 주체와 객체의 수가 많으며, 이들 사이에 허용된 권한의 수가 많은 응용 환경에 적용할 경우 매우 효과적일 것으로 사료된다.

8. 결론 및 향후 연구

본 논문에서는 XML 문서에 대한 접근 권한과 웹에서의 접근을 관리할 수 있는 시스템을 설계하고 구현하였다. 먼저, XML 문서에 필요한 권한 유형을 정의하였으며, 사용자의 계층 구조를 DAG 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 접근할 수 있도록 하였다. 이를 위하여 접근 제어 리스트를 사용하는데, 본 논문에서는 DAG를 완전 k-ary 트리로 변환하여 접근 권한을 부모와 자식 노드 간에 상속받을 수 있도록 함으로써 접근 제어 리스트의 크기를 상당히 줄이면서도 권한 주체의 접근 권한을 빨리 알아낼 수 있는 새로운 방법을 제시하였다.

이렇게 구현한 접근 권한 관리 시스템을 대학의 학생 성적 관리에 적용하였다. 학생 성적 관리를 위한 접근 관리 시스템은 학교 구성원인 사용자 주체 관리와 이들이 등록한 XML 문서에 대한 접근 관리로 구성된다. 사용자 권한 관리로는 학교 구성원들의 그룹을 권한 주체로 설정했고, 권한 주체들을 DAG 형태로 구성하여 XML 스키마와 인스턴스 그리고 엘리먼트에 대한 권한을 검사, 부여, 취소할 수 있도록 하였다. XML 문서에 대한 접근 관리로는 스키마 문서와 인스턴스 문서로 나누어 처리하였고, 사용자가 인스턴스 문서에 접근할 경우 접근 제어 리스트를 이용하여 사용자 주체 별로 접근 권한을 소유한 내용만을 보여주도록 처리하였다. 그 결과 사용자의 권한에 따라서 각기 다른 XML 문서를 작성할 필요없이 하나의 문서로부터 권한에 맞는 내용만을 보여줌으로써, 문서의 중복으로 인한 문제들을 해결하면서 문서를 권한에 따라 보호할 수 있었다.

향후에는 문서의 내용이 변경되었을 때 동일한 문서의 여러 버전을 관리할 수 있도록 확장하는 연구가 요구된다. 또한, 사용자에게 편리한 인터페이스를 제공하는 권한 관리 도구의 구현이 필요하다.

참 고 문 헌

[1] T. Bray and J. Paoli, "Extensible Markup Language(XML) 1.0," http://www.w3c.org/TR/1998/REC-1998210.html, 1999.

[2] C. F. Goldfarb and P. Prescod, The XML Handbook, Prentice Hall, 1998.

[3] C. F. Goldfarb, The SGML Handbook, Oxford Univ. Press, 1990.

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

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

[6] D. C. Fallside, "XML Schema Part 0 : Primer," http://www.w3.org/TR/xmlschema-0/, Sep. 2000.

[7] H. S. Thompson, et. al., "XML Schema Part 1 :

Structures," http://www.w3.org/TR/xmlschema-1/, Sep. 2000.

[8] P. V. Biron, K. Permanente, and A. Malhotra, "XML Schema Part 2 : Datatypes," http://www.w3.org/TR/xmlschema-2/, Sep. 2000.

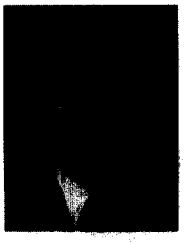
[9] A. Davidson, et. al., "Schema for Object-Oriented XML 2.0," http://www.w3.org/TR/NOTE-SOX, July 1999.

[10] A. Layman, et. al., "XML-Data," http://www.w3.org/TR/1998/NOTE-XML-data-0105/, Jan. 1998.

[11] R. S. Sandhu, et. al., "Role-Based Access Control Models," IEEE Computer, Vol. 29, No.2, pp.38-47, Feb. 1996.

[12] Y. K. Lee, et. al., Index Structures for Structured Documents, Proc. of the ACM Int'l Conf. on Digital Libraries, Bethesda, Maryland, Mar. 1996.

[13] L. Wood and V. Apparao, "Document Object Model (DOM) Level 1 Spec.," http://www.w3.org/TR/REC-DOM-Level-1/, 1999.



이 용 규

e-mail : yklee@dgu.edu

1986년 동국대학교 전자계산학과 (학사)

1988년 한국과학기술원 전산학과 (석사)

1996년 Syracuse University (전산학 박사)

1978년~1983년 정보통신부 국가공무원

1988년~1993년 한국국방연구원 선임연구원

1996년~1997년 한국통신 선임연구원

1997년~현재 동국대학교 컴퓨터멀티미디어공학과 교수

관심분야 : XML 및 웹, 저장시스템, 데이터베이스, 정보검색