通信情報保護學會論文誌
第6卷 第3號, 1996. 9

바다-Ⅲ 시스템의 데이타베이스 서버를 위한 권한부여 모델의 구현

김 영 균*, 조 옥 자*

Implementation of the Authorization Model for the Database Server of BADA-Ⅲ system**

Young-Kyun Kim*, Ok-Ja Cho*

요 약

WWW 환경에서 멀티미디어 정보 서비스를 제공하기 위해 사용되는 데이타베이스 서버는 데이타베이스에 구축된 정보 서비스를 인가된 사용자가 안전하게 사용하도록 해주는 접근 통제 기법을 제공해야 한다. 본 논문에서는 바다-Ⅲ 시스템의 객체지향 데이타베이스 서버에서 데이타의 적절한 접근 통제를 시행하는 권한부여 모델과 권한부여 정책들을 정의하고, 사용자의 접근 권한을 평가하기 위한 알고리즘을 제안한다. 또한 바다-Ⅲ 시스템의 객체지향 데이타베이스 서버에 제안된 권한부여 모델을 통합하여 구현한다. 제안된 모델은 WWW 정보 서비스 환경을 고려하여, 데이타베이스 서비스 제공자가 구축한 데이타베이스를 쉽고 효율적으로 보호할 수 있는 구조를 제공한다.

Abstract

Database servers that are used to provide multimedia information services in World Wide Web(WWW) environment have to support the access control mechanism that allows authorized users to access the constructed databases.

In this paper, we define an authorization model as well as authorization policies to enforce the proper access control on databases in the BADA-Ⅲ object-oriented database server and propose an access evaluation algorithm. Also we implement this model and the algorithm in the BADA-Ⅲ database server. Considering the service environment of the WWW, we expect that database service providers can simply and effectively protect thier data using the proposed model.

---

* 한국전자통신연구소 컴퓨터연구단 소프트웨어연구부 데이타베이스연구실

** 본 논문은 한국전자통신연구소에서 수행하는 "데이타베이스 서비스 시스템 개발 사업" 결과물의 일부임.

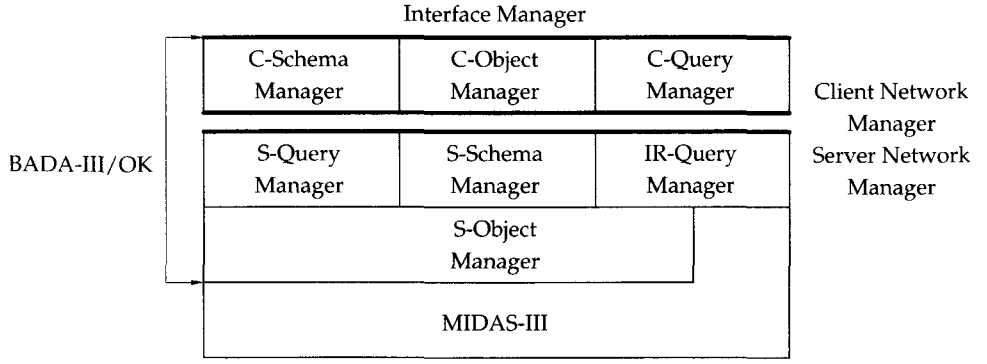
1. 서 론

바다-Ⅲ 시스템[8,14,16]은 초고속 정보 통신망에서 전세계의 다양한 사용자에게 여러 종류의 멀티미디어 데이타베이스 서비스를 제공하기 위한 연구 개발 시스템으로서, 객체지향 데이타베이스 시스템 기술과 현재 급속도로 이용이 확산된 WWW(World Wide Web) 기술을 결합한 것이다.

바다-Ⅲ 시스템의 핵심 요소인 객체지향 데이타베이스 시스템의 엔진은 WWW에서의 데이타 관리와 저장을 담당하는 효율적인 데이타베이스 서버의 역할을 수행하며, 그림 1에서와 같이 바다-Ⅲ/객체지향 커널(Object-oriented Kernel:OK)과 바다-Ⅲ/MIDAS로 구성된다. 객체지향 커널의 중요한 특징은 객체지향 질의 뿐만 아니라 전문 검색을 위한 정보 검색형 질의를 제공하고, 데이타베이스 외부의 다양한 형태의 미디어에 저장된 멀티미디어 자료를 객체지향 관리가 가능하도록 하는 오프라인(off-line) 미디어를 지원하는 것이다. 그리고 기본적으로 객체지향 데이타 모델을 지원하며, 스키마 관리 기능과 질의 기능 등을 제공한다. 반면에, 바다-Ⅲ/MIDAS는 디스크 관리 기능, 효율적인 접근을 위한 인덱스 관리 기능, 트랜잭션 및 회복 기능 등을 지원하는 하부 자료 저장 시스템이다[15].

초고속 정보 통신망을 기반으로 하여 바다-Ⅲ 시스템의 데이타베이스 서버에 구축된 멀티미디어 정보 서비스를 제공할 때 기본적인 데이타베이스 시스템의 기능들 뿐만 아니라 구축된 특정 정보 서비스를 인가된 사용자만이 안전하게 사용할 수 있도록 통제해 주는 접근 통제 기법(access control mechanism)이 필요하다. 특히, 바다-Ⅲ 시스템의 서비스 환경이 기본적으로 정보를 수집하여 데이타베이스로 구축하는 데이타베이스 생산자, 이를 통신망을 이용하여 가입자에게 서비스하는 데이타베이스 유통업자, 서비스 이용자 그리고 이들간의 통신을 제공하는 통신업자로 구성된다고 가정할 때, 데이타베이스 생산자와 유통업자가 자신들이 구축한 데이타베이스에 대해서 적절한 접근 통제 정책을 세우고, 이를 효과적으로 유지하기 위해서는 데이타베이스 서버에 적절한 접근 통제 기능이 필수적으로 제공되어야 한다.

따라서 본 논문에서는 WWW을 통하여 멀티미디어 서비스를 제공하는 바다-Ⅲ 시스템의 데이타베이스 서버가 데이타베이스 접근 통제 기능을 지원하기 위해 객체 관리를 담당하는 바다-Ⅲ/객체지향 커널의 권한부여 모델(authorization model)과 권한부여 정책을 제안



〈그림 1〉 바다-Ⅲ 시스템의 데이타베이스 서버의 구조

한다. 또한 사용자가 질의어를 통해 실제 데이타베이스를 접근할 때 그 사용자의 권한 평가를 수행하는 알고리즘을 제시한다. 그리고 제안된 권한부여 모델을 현재 구현된 객체지향 커널에 통합시켜 개발한 바다-Ⅲ/객체지향 커널의 구현 구조를 제시한다.

제안된 권한부여 모델의 특징은 기존의 데이타베이스 관리 시스템으로서 갖는 기본적인 보안 메카니즘을 지원할 뿐만 아니라 특히, WWW 서비스 환경에서 데이타베이스 생산자와 유통업자가 보다 효율적이고, 간편하게 접근 통제 정책을 시행할 수 있도록 지원하는 것이다.

일반적인 데이타베이스 시스템의 접근 통제 기법은 자율적(discretionary) 접근 통제 기법과 강제적(mandatory) 접근 통제 기법으로 구분되며, Gemstone, O2, UniSQL, Versant 등과 같은 상용 객체지향 데이타베이스 시스템들은 자율적 접근 통제 기법을 채택하고 있다. 데이타베이스 시스템에서 자율적 접근 통제 기법은 특정 사용자에 대한 권한부여 대상(target)의 종류에 따라서 다음과 같은 세가지 형태의 접근방법들로 구분할 수 있다.

- 클래스 또는 어트리뷰트 중심 접근 통제 기법
- 뷰 중심 접근 통제 기법
- 메소드 중심 접근 통제 기법

첫번째 방법은 사용자에게 부여할 권한의 대상으로 클래스 또는 어트리뷰트를 사용하는 것이다[3,6,7,10]. 이 방법에서는 사용자의 질의 요청이 있을 경우, 그 질의를 데이타의 권한부여 대상에 대한 기본적인 질의 연산(읽기, 쓰기)으로 분해하여, 각각의 연산을 권한부여 규칙의 집합인 접근 통제 리스트와 비교하므로써 사용자의 접근 통제를 실시한다.

두번째는 첫번째 방법과 비슷하지만 권한부여 대상을 뷰(view)로 제한시키는 방법이다[1]. 이 방법의 특징은 객체의 값에 기반한 접근 통제가 가능하다는 것이며, 기본적으로 데이타베이스 시스템이 뷰를 지원할 수 있어야한다.

세번째 방법은 객체지향 시스템의 기본적 특성인 캡슐화 개념를 활용한 방법이다[5,11]. 캡슐화를 따르면 객체는 보호된 인터페이스인 메소드를 통해서만 접근될 수 있기 때문에 객체의 메소드에 실행 권한을 부여하므로써 그 객체의 접근 통제가 이루어진다. 즉, 권한부여 대상으로 메소드를 이용하는 것이다. 이 방법은 객체지향 개념을 가장 적절히 활용한 방법으로 많은 연구가 되고 있으나, 실제 데이타베이스 응용들에서 데이타베이스 사용자가 메소드를 정의하여 사용하기가 쉽지 않다는 응용 적용성의 문제가 존재한다.

대부분의 상용 시스템들은 첫번째 방법을 가장 많이 사용하고 있으며, 첫번째와 두번째 방법을 통합하여 지원하는 시스템도 존재한다. 본 논문에서 제안된 바다-Ⅲ 시스템의 데이타베이스 서버를 위한 권한부여 모델은 첫번째 방법을 취한다.

자율적 접근 통제를 시행하기 위해서는 데이타베이스를 사용하는 모든 사용자에 대한 접근 권한 리스트를 유지해야 하기 때문에 사용자가 증가할 수록 이에 상응하는 권한부여 규칙의 수가 증가하게 되고, 이는 상당한 기억 공간을 차지하게 된다. 이러한 문제는 자율적 접근 통제 기법에서 중요한 요소이며, 접근 통제 리스트에는 가능한 최소한의 권한부여 규칙들이 정의되어야 한다. 이를 해결하기 위한 방법으로 묵시적 접근 권리(implied access right)와 부정적 권한부여(negative authorization) 개념이 제안되었다[6,7]. 일부 데이타베이스 시스템들이 이러한 권한부여 기능들을 지원하고 있으며, 본 논문에서 제안되는 권한부여 모델도 묵시적 접근 권리와 부정적 권한부여 개념들을 모두 지원한다.

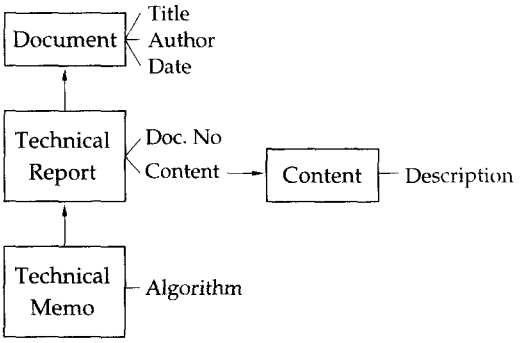
본 논문은 다음과 같이 구성된다. 먼저, 2장에서 권한부여 모델의 기반이 되는 객체지향 커널의 데이타 모델을 설명한다. 3장에서는 접근 통제 기법의 중요한 부분인 정형화한 권한부여 모델을 정의하고, 다양한 권한부여 정책과 사용자 질의에 의해 파생되는 접근 권한 평가를 수행하기 위한 알고리즘을 제시한다. 그리고 제안된 권한부여 모델을 객체지향 커널에 통합시켜 구현한 구조를 4장에서 설명하고, 또한 타 시스템들과의 기능 비교도 수행한다. 마지막으로 5장에서는 결론과 추후 연구방향에 대해 언급한다.

2. 바다-Ⅲ/객체지향 데이타 모델

바다-Ⅲ/객체지향 커널의 데이타 모델에서 객체는 단순 객체와 복합 객체로 구분된다. 단순 객체는 객체의 상태가 값(value)으로 표현되는 반면에 복합 객체에서는 다른 객체를 참조하기 위해서 객체의 상태가 그 객체의 식별자로 표현된다. 따라서 복합 객체에 의해 연결되는 객체들은 참조 관계(reference relationship)로 표현된다. 또한, 모든 객체는 객체지향 커널에서 제공하는 유일한 식별자를 갖으며, 값은 그 자체가 식별자인 객체로 기본 자료형(정수, 실수, 문자, 집합형, 대규모 자료형 등)에 속한다.

동일한 어트리뷰트들의 집합을 갖는 객체들은 하나의 클래스로 그룹화되며, 각 객체는 오직 한 클래스에만 소속될 수 있다. 데이타베이스에 정의되는 클래스들의 이름은 고유하며, 그 구성은 객체의 정적 성질을 표현하는 어트리뷰트와 동적 성질을 나타내는 메소드로 이루어진다.

그리고 데이타 모델은 클래스들 사이의 상속 관계를 지원한다. 상속은 어트리뷰트와 메소드 모두를 계승하는 완전 상속이며, 오직 한 클래스에서만 멤버를 계승하는 단일 상속이다. 상속시 발생할 수 있는 충돌 문제를 해결하기 위하여 계승받는 멤버의 이름을 재정의(override)할 수 있으며, 어트리뷰트의 영역에 대한 제약을 강화시키거나 메소드의 구현을 변경하는 것과 같은 계승된 멤버의 특성 변경이 재정의를 통하여 수행될 수 있다.



〈그림 2〉 Document 데이타베이스의 스키마 구조

바다-Ⅲ/객체지향 커널의 데이타 모델의 구성 요소로 표현된 예제 스키마가 그림 2에 제시되어 있다. 위 그림에서 Document와 Technical_Report 사이와 Technical_Report와 Technical_Memo 사이는 상속 관계가 정의되며, Technical_Report는 Content를 참조하는 복합 객체이다.

3. 바다-Ⅲ/객체지향 커널의 권한부여 모델

기존의 권한부여 모델에서는 데이타베이스에 접근하는 사용자, 접근을 허용 또는 제한할 데이타 항목 그리고 사용자의 데이타 접근 형태를 권한부여 모델의 필수적인 구성요소로 정의하고 있다[7]. 이러한 세가지의 항목들을 하나의 튜플로 묶어 정의하므로써 어떤 사용자가 특정 데이타에 연산을 수행할 수 있는 권리를 갖는다는 사실을 정의하고 있다.

3.1 권한부여 모델의 구성

제안된 모델은 권한부여 주체, 권한부여 대상, 권한이 부여된 주체가 대상에 대해 수행하는 접근 연산 그리고 권한부여의 타입으로 구성된다. 권한부여 주체는 데이타베이스를 이용하기 위해 접근하는 사용자이며, 권한부여 대상은 데이타베이스에 저장된 정보 즉, 객체를 의미한다. 그리고 접근 연산은 데이타베이스에 대한 질의어로서 그 종류로는 삽입, 삭제, 갱신, 검색 그리고 메소드의 실행이 있다.

일반적인 권한부여 정책은 개방 시스템(open system) 보호 정책 또는 폐쇄 시스템(closed system) 환경에서의 보호 정책으로 구분된다[13]. 대부분의 데이타베이스 관리 시스템들은 폐쇄적 환경 정책을 채택하여 사용자가 데이타에 대해 명시적인 접근 권한이 부여된 경우만 접근을 허용하는 긍정적 권한부여 방법을 사용하고 있다. 그러나, 인터넷을 통한 정보 서비스 환경에서는 많은 양의 데이타가 존재하고, 또한 불특정 다수의 사용자가 존재하기 때문에 긍정적 권한부여 보다는 부정적 권한부여가 훨씬 효율적인 상황이 발생한다. 따라서, 제안된 권한부여 모델은 두 가지 형태의 권한부여 정책을 모두 채택하여 사용자가 특정 객체에 연산을 수행할 수 있는 권리를 나타내는 긍정적 권한부여와 수행 권리를 제한시키는 부정적 권한부여를 모델의 구성요소로 포함시킨다.

따라서 이러한 네가지 구성요소들을 하나의 튜플 형태로 정의한 것이 권한부여 규칙(authorization rule)이며, 이는 다음과 같은 권한부여 규칙 객체로 정의된다.

■ **정의 1** 권한부여 규칙 객체(Authorization Rule object)

권한부여 규칙 AR의 영역이 S × O × R × T일 때, 권한부여 규칙 객체는 어트리뷰트 s, o, r, t로 구성된다. 즉, AR = (s, o, r, t) where s ∈ S, o ∈ O, r ∈ R, t ∈ T.

S : 권한부여 주체, O : 권한부여 대상,
R : 접근연산, T : 권한부여 타입

3.1.1 권한부여 주체

기존의 권한부여 모델들에서 특정 데이타베이스의 정보를 검색하거나 또는 새로운 정보를 삽입할 목적으로 이 데이타베이스에 접근하는 개인 사용자나 그룹이 권한부여의 주체가 된다. 즉, 개인 사용자 p_i, $p_j \in P$ 그리고 그룹 g_i, $g_j \in G$가 존재할 때, 권한부여 주체인 S는 P와 U의 합집합이다. 즉, $S = P \cup G$이다.

WWW 서비스 환경에서 특정 웹 서버에 접근하는 사용자 수를 고려할 때, 개인 사용자별로 데이타베이스에 대한 권한부여 규칙을 유지시키는 방법은 실효성이 없으며, 혹시 개인별로 유지하더라도 수많은 권한부여 규칙의 관리 오버헤드를 감수해야 한다. 따라서 이러한 오버헤드를 줄이기 위한 방법으로 본 논문에서 제시된 권한부여 모델은 데이타베이스에 그룹들을 생성하고, 이 그룹들에 여러 개인 사용자들을 등록시켜서 개인 사용자에 대한 권한부여 규칙을 정의하는 대신에 해당 그룹에 대한 권한부여 규칙을 정의할 수 있도록 한다. 결과적으로, 질의어를 수행할 때, 특정 객체에 대한 각 사용자의 접근 권한 평가는 그 사용자가 소속된 그룹의 접근 권한 평가로 대체된다.

■ **정의 2** 권한부여 주체

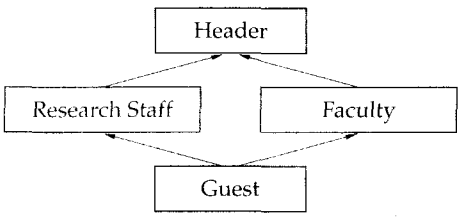
접근권한 주체를 S라 할 때,

$$S = G_i \cup G_j \cup G_k$$

where $\exists p_i \in G_i$, $\exists p_j \in G_j$, $\exists p_k \in G_k$
in $\forall p_i, p_j, p_k \in P$.

그리고 데이타베이스 내에 생성된 그룹들 사이에는 관계가 설정될 수 있으며, 이 관계를

그룹 권한 계층구조(group privilege hierarchy)라 한다. 즉, 두개의 그룹 $g_i, g_j \in G$가 존재할 때 $g_i < g_j$는 g_j와 g_i 사이의 권한 포함 관계를 표현하고, g_j는 g_i의 상위 그룹이며 자신의 하위 그룹 gi에 대해 정의된 권한부여 규칙들을 묵시적으로(implicitly) 갖는다. 예를 들면, 그림 3에서 만약 Faculty 그룹이 O_i 객체에 검색 권한이 있는 권한부여 규칙이 존재한다면 그룹 권한 계층구조의 의미에 의해서 Header 그룹은 묵시적으로 객체 O_i에 대해 검색 권한을 갖는다.



〈그림 3〉 그룹 권한 계층구조의 예

이와 같은 그룹들간에 권한부여 규칙의 계승은 상위 그룹에 대하여 정의할 권한부여 규칙의 수를 감소시키기 때문에 시스템 관리자의 접근권한 관리 작업을 단순화시키는 잇점을 갖는다[4,9,12].

■ 정의 3 그룹 권한 계층구조

g_i, g_j가 그룹이면, 그룹 권한 계층구조를 HG는 다음과 같다.

$$HG = g_i < g_j$$

satisfying $\exists g_i, g_j \in G$, $AR(g_i) \subseteq AR(g_j)$, where "$<$" denotes that g_j is a super group of the group g_i.

3.1.2 권한부여 대상

권한부여 대상은 데이타베이스에서 보호해야 할 대상이 무엇인지를 표현하는 것으로, 이는 시스템이 어떤 수준까지의 보호 메카니즘을 지원할 것인가를 결정하는 중요한 요소이다. 일반적으로 객체지향 기반 권한부여 모델에서 객체의 보호 단위로는 클래스, 인스턴스, 어트리뷰트 그리고 메소드가 있다.

바다-Ⅲ 시스템은 멀티미디어 서비스를 구축하는 데이타베이스 생산자에게 이용되기 때문에 가능한 서비스 제공자가 구축된 데이타에 대해 효율적이고 간편하게 접근 통제를 수행할 수 있어야 한다. 웹 서비스 환경에서 객체 인스턴스 수준의 접근 통제 기법을 지원하는 것은 수많은 객체 인스턴스 각각에 대한 권한부여 규칙의 정의와 관리에 대한 오버헤드 그리고 접근 권한 평가에 대한 시간적 부담을 초래한다.

따라서 이러한 바다-Ⅲ 시스템의 서비스 환경을 고려하여 제안된 권한부여 모델은 클래스와 어트리뷰트 그리고 메소드를 권한부여 대상으로 선정하고, 이 수준까지의 접근 통제를 지원한다.

■ 정의 4 권한부여 객체

권한부여 객체를 O로 표현할 때,

$$O = C \cup A \cup M,$$

where $\forall c_i \in C$, $\forall a_j \in A$, $\forall m_k \in M$.

C : 클래스, A : 어트리뷰트, M : 메소드

3.1.3 접근 연산

접근 연산은 권한부여 주체가 권한부여 대상에 대해 수행할 수 있는 권리이며, 그 종류로 객체의 삽입, 삭제, 갱신 그리고 검색과 객체의 메소드 실행 권리를 정의한다.

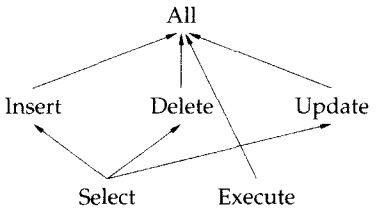
■ 정의 5 접근 연산

접근 연산을 R로 표현할 때,

$$R = \{\text{insert, delete, update, select, execute}\}.$$

insert : 삽입, delete : 삭제, update : 갱신, select : 검색, execute : 메소드 실행

또한, 정의된 각각의 접근 연산들 사이에는 그림 4에서와 같은 연산의 계층구조가 형성될 수 있다. 이 계층구조는 부분 순서 집합(partially ordered set)의 특성을 갖으며, 접근 연산 계층구조라 한다. 이 계층구조에는 아크로 연결된 상위 연산에 대한 권리가 상호 연결된 하위 연산에 대한 권리를 포함하는 함축적인 의미가 내포되어 있다. 예를 들어, 어떤 접근 권한 주체가 특정 객체에 대해 삭제 또는 삽입 연산에 대한 권리를 갖으면 이 사용자는 그 객체에 대해서 묵시적으로 검색 연산에 대한 권리를 갖는다. 그러나 역은 성립하지 않는다.



〈그림 4〉 접근 연산 계층구조

3.1.4 권한부여의 타입

권한부여 주체가 특정한 권한부여 대상(또는 객체)에 대해 접근이 불가능한 이유는 두 가지로 정리된다. 첫째는 그 객체에 대해 긍정적 권한부여 규칙이 존재하지 않는 경우이며, 둘째는 그 객체에 대해 부정적 권한부여 규칙이 존재하는 경우이다. 따라서, 전형적인 권한부여 타입으로 긍정적(positive) 권한부여와 부정적(negative) 권한부여가 존재한다. 긍정적 권한부여는 주체가 객체에 대해 정의된 연산을 수행할 수 있는 가능성을 의미하며, 역으로 부정적 권한부여 타입은 그 연산을 수행할 수 없다는 불가능을 의미한다.

■ **정의 6** 권한부여 타입

권한부여의 타입을 T라 할 때,

$T = \{+, -\}$, where

"+" : 긍정적 권한부여,

"−" : 부정적 권한부여.

묵시적인 접근 권리를 재정의할 필요성이 있거나, 또는 특정 객체에 대해 정밀한 권한부여 규칙을 정의하는 환경에서 부정적 권한부여 타입은 유용하게 사용될 수 있기 때문에 제안된 권한부여 모델에서는 두가지 형태의 권한부여 타입을 모두 지원한다.

3.2 모델의 보안 정책

특정 데이타베이스에 대해서 서비스 제공자가 정의하는 권한부여 규칙들이 기본적으로 권한부여 규칙의 집합을 구성하지만, 이 집합이 데이타베이스 보안을 유지하기 위해서 사용되는 권한부여 규칙의 전체적인 집합은 아니다. 즉, 사용자에 의해 명시적으로 정의되는 권한부여 규칙들을 기반으로 하여 그룹 권한 계층구조, 객체인 클래스 계층구조와 복합 객체 구조 그리고 접근 연산의 계층구조들 사이의 상호관계에서 새로운 권한부여 규칙들이 유도된다. 이렇게 유도되는 권한부여 규칙 객체를 묵시적 권한부여 규칙 객체라 한다.

먼저, 다음의 정의에서 사용되는 $a \rightarrow a'$ 기호의 의미는 다음과 같다. $a_0 = a$, $a_n = a'$이고, $a_0 \rightarrow a_1 \rightarrow \ldots \rightarrow a_n$인 권한부여 규칙들 $a_0, a_1, \ldots, a_n$이 권한부여 규칙 집합에 존재함을 표현한다.

■ **정의 7** 묵시적 권한부여 규칙 객체 (implicit authorization rule object)

사용자가 정의한 명시적 권한부여 규칙 객체의 집합을 MAR이라 할 때,

$a \rightarrow a' \in ARO$ where $\exists a' \notin MAR$, $\exists a \in AR$, $AR \supseteq MAR$.

묵시적 권한부여 규칙의 존재는 데이타베이스에 저장된 명시적 권한부여 규칙과 이들에서 유도된 묵시적 권한부여 규칙들 사이의 일관성 유지의 복잡성을 증가시킨다. 또한, 이 두가지 형태의 권한부여 규칙들에서는 부정적 권한부여 규칙과 긍정적 권한부여 규칙이 모두 사용되기 때문에 항상 권한 충돌 문제가 내포된다. 따라서 이러한 문제를 해결하기 위해서는 묵시적 권한부여 규칙의 유도 정책과 권한부여 규칙의 충돌 문제를 해결하는 정책이 제공되어야 한다.

3.2.1 묵시적 권한부여 규칙의 유도 정책

권한부여 주체인 그룹들에는 그룹 권한 계층구조가 존재하고, 이는 묵시적 권한부여 규칙의 생성 기회를 함축한다. 즉, 그룹 권한 계층구조에서 상호연결된 그룹들에서 상위 그룹이 하위 그룹의 권한을 묵시적으로 포함하기 때문에 하위 그룹이 갖는 명시적 권한부여 규칙을 상위 그룹은 묵시적으로 갖는다. 그렇지만, 이 역은 성립하지 않는다.

〈권한부여 정책 1〉 그룹 권한 계층구조의 묵시적 권한 유도

상위 그룹은 연결된 하위 그룹에 정의된 모든 권한부여 규칙들을 묵시적으로 갖는다.

Implicit Authorization: $\forall g_c, g_p \in G$, $\forall o \in O$, $\forall r \in R$, $\forall t \in T$:
$(g_c, o, r, t) \rightarrow (g_p, o, r, t)$
iff $g_c < g_p$, where "$<$" denotes that g_p is supergroup of g_c

예를 들어, 그림 2의 스키마 구조에 대해서 그림 3에 정의된 각 그룹들에 대한 권한부여 규칙이 다음과 같이 주어졌다고 가정하자.

권한부여 규칙 AR_1
= (Guest, Document, select, +)
권한부여 규칙 AR_2
= (Guest, Technical_Report.{number, content}, select, +)
권한부여 규칙 AR_3
= (ResearchStaff, Content.description, select, +)
권한부여 규칙 AR_4
= (ResearchStaff, Technical_Memo.algorithm, select, -)
권한부여 규칙 AR_5
= (Header, Technical_Memo, delete, +)

그룹간의 묵시적 권한 유도의 예로, ResearchStaff이 Technical_Report.number를 검색할 경우는 명시적인 권한부여 규칙이 존재하지 않는다. 그러나 ResearchStaff이 Guest 그룹의 상위 그룹이고, Technical_Report.number를 Guest가 검색할 수 있는 권한이 권한부여 규칙 AR_1에 명시적으로 정의되어 있기 때문에 ResearchStaff은 Technical_Report.number를 검색하는 권한이 있다.

두번째로 접근 연산 계층구조에서 유도되는 묵시적 권한부여 규칙이 존재한다. 권한부여 정책 1과 비슷한 경우로서 상호연결된 접근 연산들 사이에 상위 연산을 수행할 수 있는 권리는 하위 연산을 수행할 수 있는 권리를 함축하고 있다. 그러나 이러한 묵시적 권한은 긍정적 권한부여 규칙에서만 유효하고, 부정적 권한부여 규칙에는 적용되지 않는다. 예를 들어, 특정 객체에 대해 갱신 연산을 수행할 수 없는 규칙에서 그 객체에 대해 검색 연산을 수행하지 못하는 묵시적 권한부여 규칙은 유도되지 않는다.

〈권한부여 정책 2〉 접근 연산 계층구조의 묵시적 권한 유도

접근 연산 계층구조에서 상위 접근 연산을 수행할 수 있는 권리는 묵시적으로 하위 접

근 연산에 대한 권리를 갖는다.

Implicit Authorization: $\forall g \in G$, $\forall o \in O$, $\forall r_i, r_j \in R$, $\forall t \in T$: $(g, o, r_j, t) \rightarrow (g, o, r_i, t)$ iff $r_i < r_j$ and $t == +$, where "$<$" denotes that r_j is a dorminant operation of r_i.

예를 들면, Header 그룹이 Technical_Memo.algorithm을 검색하는 질의를 수행한 경우는 Header 그룹이 이 어트리뷰트를 검색하는 권한이 존재한다는 명시적 권한부여 규칙이 없지만, 이 그룹은 algorithm을 삭제하는 권한이 권한부여 규칙 AR_5로 존재한다. 따라서 그림 4의 접근 연산 계층구조에서 삭제 연산은 검색 연산의 상위 연산으로 정의되기 때문에, 최종적으로 Header그룹은 Technical_Memo.algorithm을 검색하는 권한을 갖는다.

2장에서 언급된 바다-Ⅲ/객체지향 커널의 기본 데이타 모델은 데이타베이스에 정의되는 클래스들 간에는 클래스 상속 계층구조와 클래스 참조 관계를 지원한다. 그러므로 이러한 어트리뷰트와 메소드가 계승되는 상속 계층구조에서 묵시적인 권한부여 규칙이 존재할 뿐만 아니라 클래스와 클래스 멤버 사이의 묵시적 권한 함축이 존재한다. 따라서 권한부여 대상에 대한 묵시적 권한부여 규칙의 유도 정책은 다음과 같이 정의된다.

〈권한부여 정책 3〉 클래스의 묵시적 권한 유도

클래스에 대해 특정 연산을 수행할 수 있는 권리는 이 클래스에 정의된 어트리뷰트와 메소드에 대해서 동일한 권리를 갖는다.

Implicit Authorization: $\forall g \in G$, $\forall o_i, o_j \in O$, $\forall r \in R$, $\forall t \in T$: $(g, o_i, r, t) \rightarrow (g, o_j, r, t)$ iff $o_j \in o_i$

예를 들면, Guest 그룹이 Document.author를 검색하려 할 때의 접근 권한의 결정은 권한부여 규칙 AR_1에 의하면 Guest 그룹이 author 어트리뷰트를 포함하는 클래스 Document에 대해서 검색하는 권한이 존재한다. 따라서 AR_1권한부여 규칙과 권한부여 정책 3에 의해서 Guest 그룹은 하나의 어트리뷰트 author를 검색하는 권한을 갖는다.

〈권한부여 정책 4-1〉 상속 계층구조의 묵시적 권한 유도

상위 클래스의 멤버에 대해 권한을 갖는 그룹은 하위 클래스에 계승된 멤버들에 대해서도 동일한 권한을 갖는다.

Implicit Authorization: $\forall g \in G$, $\forall o_i, o_j \in O$, $\forall r \in R$, $\forall t \in T$: $(g, o_i, r, t) \rightarrow (g, o_j, r, t)$ iff O^p is a superclass of O^q & $o_i \in O^p$ & $o_j \in O^q$ & $o_i == o_j$.

Guest 그룹이 Technical_Report.title을 검색하는 질의어인 예를 보면, 먼저 이 그룹이 Technical_Report 클래스의 title 어트리뷰트에 대한 명시적 검색 권한이 없다. 그러나 title 어트리뷰트는 Document 클래스에서 계승된 어트리뷰트이기 때문에, 규칙 AR_1과 정책 4-1에 의해서 Guest는 Technical_Report.title에 대한 검색 권한을 갖는다.

〈권한부여 정책 4-2〉

계승된 멤버에 대해 권한을 갖는 그룹은 그 멤버를 정의한 클래스의 멤버에 대한 동일한 묵시적 권한을 갖을 수 없다.

Implicit Authorization: $\forall g \in G$, $\forall o_i, o_j \in O$, $\forall r \in R$, $\forall t \in T$: $(g, o_j, r, t) \hat{\rightarrow} (g, o_i, r, t)$ iff O^p is a superclass of O^q & $o_i \in O^p$ & $o_i \in O^q$ & $o_i = o_j$.

이 정책에 대한 예는 다음과 같다. 권한부여 주체로서 Guest와 ResearchStaff가 존재하고, 이들 사이에 그룹 권한 계층구조가 형성되어 있지 않고, 권한부여 규칙 AR__6 = (Guest, Document.author, select, +)와 AR__7 = (ResearchStaff, Technical__Report.title, select, +)이 정의되었다고 가정하자. 만일 ResearchStaff이 Document.author를 검색하려 할 때, Guest 그룹은 이 어트리뷰트를 검색하는 권한을 갖지만, ResearchStaff이 Guest 그룹에 대해 상위 그룹으로 정의되지 않았을 뿐만 아니라 명시적으로 Document.author에 대한 권한을 갖지 않기 때문에 이 어트리뷰트에 대해 검색 권리를 갖지 못한다.

객체지향 커널의 데이타 모델에서 복합 객체는 단지 다른 객체를 참조하는 객체로 정의된다. 따라서 한 객체의 어트리뷰트의 값이 다른 객체의 식별자를 갖고, 그 어트리뷰트에 대해 검색 권한이 있는 경우에, 대부분의 권한부여 모델들에서 취한 방법은 사용자는 참조된 객체의 식별자 값만을 검색할 수 있을 뿐 그 객체의 멤버에 대해서는 검색 권한을 제한하고 있다.

〈권한부여 정책 5〉 참조 객체에서 권한의 제한
복합 객체인 경우, 참조되는 객체의 존재에 대한 권한은 참조된 객체의 내용에 대한 권한을 내포하지 않는다.

Implicit Authorization: $\forall g \in G$, $\forall o_i, o_j \in O$, $\forall r \in R$, $\forall t \in T$:
$(g, o_i, r, t) \hat{\rightarrow} (g, o_j, r, t)$
iff $F(o_i) == G(o_j)$, where
F : function that return the value of object
G : function that return the object id.

예를 들어, Guest 그룹이 Content 클래스의 어트리뷰트 description의 값을 검색할 때, 권한부여 규칙 AR__2에 Technical__Report.content 어트리뷰트에 대해서 검색할 수 있기 때문에 description 정보를 갖는 Content 객체의 식별자는 검색할 수 있다. 그러나 참조 객체의 권한부여 정책에 의해 참조 객체 클래스 Content.desciption에 대한 권한부여 규칙 AR__3과 같은 명시적 권한부여 규칙이 없기 때문에 검색하는 권한을 갖을 수 없다.

3.2.2 권한부여 규칙의 충돌

제안된 권한부여 모델은 긍정적 권한부여 타입과 부정적 권한부여 타입을 모두 지원한다. 그러나 이러한 두가지 타입의 권한부여 규칙때문에 사용자가 정의한 권한부여 규칙들과 그룹 권한 계층구조, 클래스 상속 계층구조 그리고 접근 연산 계층구조에서 파생되는 묵시적 권한부여 규칙들 간에 충돌이 발생할 수 있다.

예를 들면, 앞에서 정의한 권한부여 규칙 AR__4와 AR__5의 경우이다. 즉, 권한부여 규칙 AR__4에 의해서 ResearchStaff 그룹은 Technical__Memo.algorithm을 검색할 수 없다. 그리고 그룹 계층구조의 묵시적 권한부여 규칙 유도에 의해서 Header 그룹은 ResearchStaff의 권한 AR__4 규칙을 묵시적으로 갖는다. 그러나 권한부여 규칙 AR__5에 의하면 Header 그룹은 Technical__Memo.algorithm을 삭제하는 권한을 갖기 때문에 연산 계층구조에 의해 검색 권한도 묵시적으로 갖는다. 따라서 이러한 경우가 충돌이 발생한 것이며, 이를 해결할 수 있는 정책이 모델에서 제공되어야 한다.

〈권한부여 정책 6〉 명시적 권한부여 규칙과 묵시적 권한부여 규칙의 충돌 해결
권한부여 주체에 대해서 명시적 권한부여 규칙과 묵시적 권한부여 규칙이 함께 존재하고, 충돌이 발생하면 명시적 권한부여 규

칙이 묵시적 권한부여 규칙보다 우선적으로 적용된다.

Conflict Resolution: $a' \in MAR$, $a \in AR : a \angle a'$
("$\angle$" denotes that a' precedes a)
iff t(a) == '+', t(a') == '-',
where t : function that return the type of an authorization rule.

3.3 접근 권한 평가 알고리즘

데이타베이스를 검색 또는 변경하는 사용자 질의어가 주어지면, 이 질의의 조건식에 서술된 모든 어트리뷰트들에 대해서 사용자의 접근 권한이 존재하는지를 검사한다. 이와 같은 접근 권한의 평가는 시스템의 질의 관리자에서 요청되며, 실제 평가는 스키마 관리자에 의해서 수행된다.

접근 권한을 평가하기 위해서는 사용자 이름, 클래스 식별자, 어트리뷰트 식별자 그리고 질의 타입에 대한 데이타가 필요하며, 이러한 데이타는 질의 관리자에 의해 제공된다. 최종적으로 스키마 관리자는 평가에 필요한 외부 데이타와 시스템 카탈로그에 저장된 권한부여 규칙, 그리고 3장에서 언급된 권한부여 정책들을 모두 고려하여 그림 5에 주어진 알고리즘에 의해서 접근 권한의 평가를 수행하여, 그 결과를 질의 관리자에게 반환한다.

```
/* evaluate class and attribute privilege */
boolean   authorization_evaluation(user, target_class, attr_id, op)
{
        /* search a rule for the target_class */
        flag = class_privilege_evaluation(user, target_class, op) );      // step 1
        if (flag == negative)
            return(0);      // permission denied
        /* in case of defined attribute */
        if (attr_id == local attribute)                                   // step 2
            flag = search_attribute_rule(user, attr_id, op);
            if (flag == (grant || denied)
                return(flag);
        /* attribute rule is undefined */
            else
                if (positive class authorization rule exists) // by authorization policy 3
                    return(1);
                else                                          // by authorization policy 4
                    return(0);
        /* in case of inherited attribute */
        else                                                              // step 3
            goto step 1;      // evaluate privilege of attr_id in the superclass
}

/* search corresponding authorization rule in catalog */
flag  search_attribute_rule(user, attr_id, op)
{
```

```
scan the Privilege_Member catalog;        // for matching the authorization rule
/* there is the positive authorization rule */
If (searched rule is positive )
      return(1);  // permission granted
/* there is the negative authorization rule */
If (searched rule is negative )
      return(0);  // permission denied
/* no authorization rule is matched */
else
      find the superset of op;
      If (op finded)
            goto search_attribute_rule(u, a, s_op);      // s_op is the acendent of op
      find the superset of group;
      If (group finded)
            goto search_attribute_rule(su, a, op);       // su is the super group of u
```

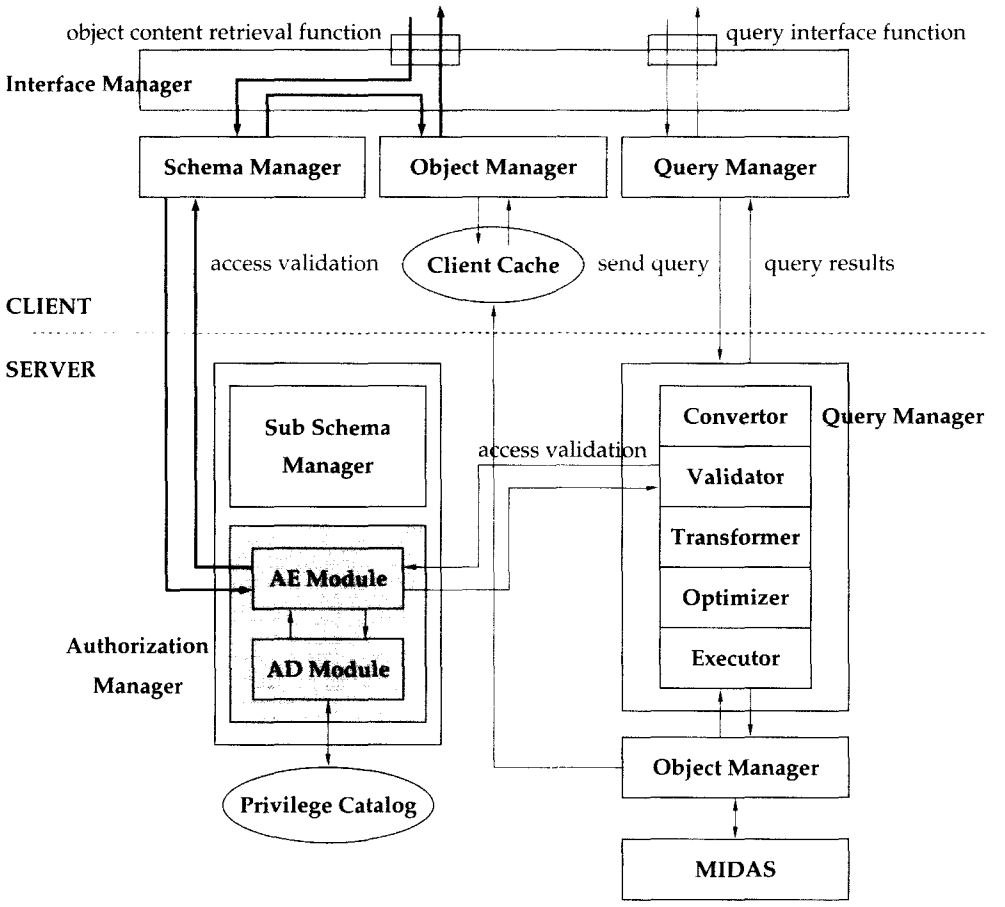
〈그림 5〉 접근 권한 평가 알고리즘

4. 권한부여 모델의 구현

대부분의 데이타베이스 시스템들에서와 같이 사용자 질의를 처리하는 과정에서 서버의 질의 관리자는 질의 조건식을 구성하고 있는 객체의 클래스 또는 어트리뷰트에 대한 사용자의 권한이 부여됐는지를 확인해야 한다. 이는 스키마 관리자의 하위 권한부여 관리자에게 접근 권한의 평가를 요청하므로써 수행된다. 하위 권한부여 관리자는 접근 실행(access enforcement) 모듈과 접근 평가(access decision) 모듈로 구성되고, 접근 평가 모듈이 시스템 권한 카탈로그에 저장된 권한부여 규칙을 참조한다.

그림 6은 1장에서 설명된 바다-Ⅲ 시스템의 데이타베이스 서버 구조에 본 논문에서 제안한 권한부여 모델을 통합하여 구현한 개략 구조를 보여준다. 현재 버젼의 바다-Ⅲ 시스템은 SQL과 같은 전형적인 질의어를 지원하지 않기 때문에 사용자 인터페이스에 질의를 위한 함수들이 제공된다. 질의 함수는 질의 수행이 성공했는지 아니면 실패했는가를 사용자에게 반환하며, 실제로 질의 조건식을 만족하는 객체들을 각각 탐색하기 위해서는 커서 인터페이스를 이용해야 한다.

이러한 객체의 커서를 사용하여 객체를 얻는 경우에는 다음과 같은 보안 헛점이 발생한다. 질의 조건식을 구성하는 객체 어트리뷰트에 대해 질의를 수행한 사용자의 권한이 존재하는 경우에는 질의 수행은 성공적으로 완료되고, 객체 커서를 통하여 각각의 객체를 얻을 수 있다. 그러나 이 때 객체의 어트리뷰트들 중에서 사용자에게 권한이 주어지지 않은 어트리뷰트들이 존재하는 경우에, 사용자는 권한이 없는 어트리뷰트의 값을 검색하게 되는 보안성 관점에서 심각한 노출 문제를 발생시킨다. 이를 해결하기 위한 방편으로 인터페이스 관리자에서 실제 객체의 어트리뷰트 값을 검색할 때 사용자의 접근 권한을 재평가할 수 있도록 한다(그림 6에서 굵은 화살선). 따라서 바다-Ⅲ 시스템의 데이타베이스 서버에서는 제안된 권한부여 모델이 지원하는 어트리뷰트 수준의 보안성이 유지될 수 있다.



〈그림 6〉 권한부여 모델의 구현 구조

4.1 시스템 카탈로그의 구성

권한부여 모델을 구현할 때 권한부여 규칙이 유지되는 장소에 따라서 시스템의 성능이 영향을 받는다. 접근 권한을 정의한 권한부여 규칙을 시스템 카탈로그에 유지시키는 방법과 접근 권한을 평가할 객체에 권한부여 규칙을 유지시키는 방법이 있으나, 현재 버전의 바다-Ⅲ/객체지향 커널에서는 스키마 관리자가 스키마 관리 뿐만 아니라 접근 권한에 대한 정의와 평가를 수행하기 때문에 시스템 카탈로그에 권한부여 규칙들을 유지시키는 방법을 채택한다.

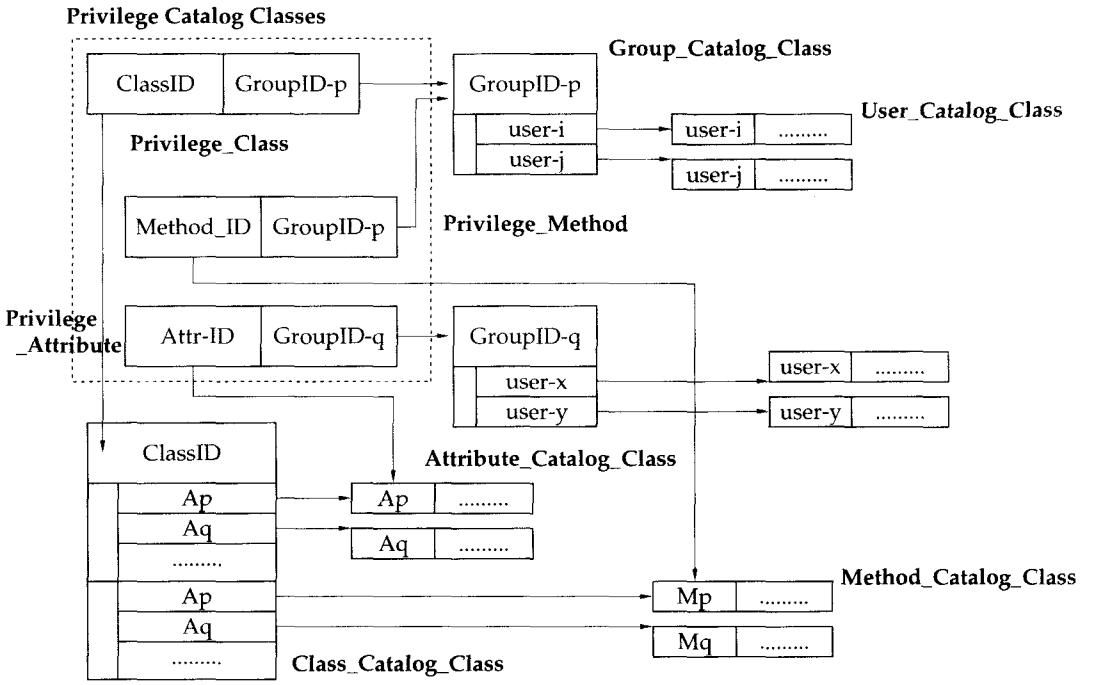
그림 7에는 바다-Ⅲ/객체지향 커널에서 접근 권한 관리를 위한 시스템 카탈로그를 구성하는 클래스들의 구조가 제시되어 있다.

4.2 권한부여 모델의 기능 비교

본 논문에서 제안한 권한부여 모델은 WWW 환경에서 정보 서비스를 제공하는 정보 제공자가 데이타 서버에서 효율적이고 간편하게 데이타베이스에 저장된 정보를 보호하는 수단을 제공할 뿐만 아니라 범용 데이타베이스 시스템에서 필요로 하는 권한 관리 기능들을 지원한다. 아래의 표 1은 바다-Ⅲ 시스템

의 데이타베이스 서버의 접근 통제 기능을 기존의 시스템들의 기능과 비교한 내용이다[2].

현재 제안된 권한부여 모델을 통합하여 구현한 바다-Ⅲ/객체지향 커널에서 데이타베이스 보안 유지를 위해 질의 관리자와 인터페이스 관리자에서의 중복된 접근 권한의 평가를 수행하기 때문에 데이타베이스 서버의 전체적인 성능면에 있어서 오버헤드가 발생할 수 있다. 이는 어트리뷰트 수준의 값을 검색하는 질의어가 제공되면 인터페이스 관리자에서 접근 권한 평가 작업이 생략되기 때문에 현재 설계중인 바다-Ⅲ 시스템의 질의어가 완성되면 쉽게 해결할 수 있다.



〈그림 7〉 시스템 카탈로그의 클래스 구성

[표 1] 권한부여 기능 비교

	Gemstone	O2	UniSQL	바다-Ⅲ
데이타 보호 단위	클래스 클래스 멤버 객체 값	클래스 클래스 멤버	클래스 클래스 멤버 객체 값	클래스 클래스 멤버
사용자 지원	개인 사용자 그룹	개인 사용자	개인 사용자 그룹	그룹
권한부여 타입	긍정 권한부여	긍정 권한부여	긍정 권한부여 부정 권한부여	긍정 권한부여 부정 권한부여
권한부여 규칙 유도	접근 연산	보호 객체 접근 연산	사용자 보호 객체	사용자 보호 객체 접근 연산

5. 결론 및 추후 연구방향

WWW 환경에서 멀티미디어 정보 서비스를 제공하는 데이타 서버에서 수 많은 사용자들에 대한 데이타베이스의 접근 권한 관리는 필수적인 요소이며, 이러한 권한 관리 기능은 데이타베이스 서버가 사용하는 하부의 데이타베이스 시스템에서 제공되어야 한다.

본 논문에서는 정보 통신망에서 정보 서비스를 제공하기 위해 한국전자통신연구소에서 개발중인 바다-Ⅲ 시스템의 데이타베이스 서버에서 사용자의 접근 통제를 시행하기 위해 필요한 권한부여 모델과 권한부여 정책들을 제안하였다. 제안된 권한부여 모델은 기존의 권한부여 모델들에서 제시한 대표적인 개념들 즉, 긍정적 또는 부정적 권한부여 기능, 묵시적인 권한부여 기능 등을 모두 지원하며, 바다-Ⅲ 시스템의 객체지향 커널에서 지원하는 데이타 모델의 제약에 따른 권한부여 정책들을 정의하였다. 그리고 실제 사용자의 질의가 주어졌을 때 권한 평가를 위해 필요한 접근 권한 평가 알고리즘을 제시하고, 제안된 권한부여 모델을 현재 구현된 프로토타입 버전의 바다-Ⅲ/데이타베이스 서버에 통합할 수 있도록 객체지향 커널의 통합 구조를 설계하고 구현하였다.

현재는 데이타베이스 관리자에 의해 정의되는 명시적 권한부여 규칙과 이로 부터 파생되는 묵시적 권한부여 규칙들 사이의 충돌을 탐지하여 전체적으로 일관된 권한부여 규칙의 관리를 수행할 수 있는 알고리즘 개발을 진행중이다. 앞으로 바다-Ⅲ 시스템의 데이타베이스 서버에서 SQL 형태의 정형화된 질의어와 뷰가 지원되면, 제안된 권한부여 모델이 이를 포함하도록 확장하고, 구현할 시스템 구조에 대한 확장 연구를 진행할 것이다.

참 고 문 헌

[1] Ahmad Baraani-Dastjerdi, et al, "A Model of Authorization for Object-Oriented Databases Based om Object Views," Proceedings of DOOD'95, Singapore, 1995, pp503-520.

[2] Barry & Associates Inc., DBMS Needs Assessment for Objects, 1994, pp20-30.

[3] Eduardo B. Fernandez, Ehud Gudes, Haiyan Song, "A Model for Evaluation and Administration of Security in Object-Oriented Databases," IEEE Trans. on Knowledge and Data Engineering, Vol. 6, No. 2, 1994, pp275-291.

[4] Eduardo B. Fernandez, Jie Wu, Minjie H. Fernandez, "User Group Structures in Object-Oriented Database Authorization," in Database Security VIII : Status and Prospects, Elsevier Science Publishers, 1994, pp57-76.

[5] Eduardo B. Fernandez, et al, "A Method-Based Authorization Model for Object-Oriented Databases," proceedings of Security for Object-Oriented Systems Workshops, Washington, Springer-Verlag, 1993, pp135-150.

[6] Elisa Bertino, Fabio Origgi, Pierangela Samarati, "A New Authorization Model for Object-Oriented Databases," in Database Security VIII : Status and Prospects, Elsevier Science Publishers, 1994, pp199-222.

[7] Fausto Rabitti, E. Bertino, W. Kim and D. Woelk, "A Model of Authorization for Next-Generation Database Systems," ACM Trans. on Database Systems, Vol. 16, No. 1, 1991, pp88-131.

[8] Mi-Ok Chae, et al, "Design of the Object Kernel of BADA-Ⅲ : An Object-Oriented Database Management

System for Multimedia Data Services," proceedings of Internation Workshop on Network and System Management, Korea, 1995, pp143-152.

[9] M. Nyanchama, S. Osborn, "Role-Based Security: Pros, Cons & Some Research Directions," ACM SIGSAC, Vol. 11, No. 2, 1993, pp11-17.

[10] M. S. Olivier, S.H. von Solms, "DISCO: A Discretionary Security Model for Object-oriented Databases," proceedings of IFIP 8th International Informantion Security Conf., Singapore, 1992, pp375-387.

[11] Nurith Gal-Oz, et al, "A Model of Methods Access Authorization in Object-oriented Databases," Proceedings of 19th VLDB Conf., 1993, pp52-61.

[12] T.C. Ting, S.A. Demurjian, M.Y. Hu, " Requirements, Capabilities, and Functionalities of User-Role Based Security for an Object-Oriented Design Model," in Database Security V : Status and Prospects, Elsevier Science Publishers, 1992, pp275-296.

[13] Silvana Castano et al, Database Security, Addison-Wesley, 1995, pp18-32.

[14] 이미영, 채미옥, 김평철, 전성택, "바다-Ⅲ/C++ : 바다-Ⅲ 에서의 C++ 결합 방법," 한국정보과학회 가을 학술발표 논문집, 22권, 2호, 1995, pp271-274.

[15] 이진수, 박순영, 채미옥, 김 준, 허대영, "MIDAS-II의 설계 및 구현," 한국정보과학회 가을 학술발표 논문집, 20권, 2호, 1993, pp183-196.

[16] 조옥자, 전성택, "BADA-Ⅲ의 스키마 관리기 설계," 한국정보과학회 가을 학술발표 논문집, 22권, 2호, 1995, pp267-270.

□ 著者紹介

김 영 균

전남대학교 전산통계학과(학사)
전남대학교 대학원 전산통계학과(이학석사)
전남대학교 대학원 전산통계학과(이학박사)
1995년 - 현재 한국전자통신연구소 소프트웨어공학연구실 선임연구원

※ 관심분야 : 객체지향 데이타베이스 시스템, 데이타베이스 보안, 분산 시스템 보안 등

조 옥 자

전남대학교 전산통계학과(학사)
1988년 - 현재 한국전자통신연구소 데이타베이스연구실 연구원

※ 관심분야 : 객체지향 데이타베이스 시스템, 멀티미디어 데이타베이스, 데이타베이스 보안