역할기반 접근제어를 적용한 객체지향 데이터베이스의 보안모델†

조기천*, 김은희*, 신문선*, 류근호*, 신기수**
*충북대학교 데이터베이스연구실
**한국통신
email : *{kicheon, ehkim, msshin, khryu}@dblab.chungbuk.ac.kr
**ksshin@kt.co.kr

Configuring RBAC to Object-Oriented Database Security Model

Ki Cheon Cho*, Eun Hee Kim*, Moon Sun Shin*, Keun Ho Ryu*, Ki Soo Shin**
*Database Laboratory, Chungbuk National University
**Korea Telecom

요 약

객체지향 데이터베이스 보안모델은 객체지향 패러다임의 각 속성에 대한 보안정책을 정의한다. 객체지향 데이터베이스의 보안 목적은 데이터베이스에 대한 권한이 없는 사용자의 접근을 제어함으로써 정보의 고의적인 파괴나 변경을 방지하고, 우발적인 사고로부터 데이터를 보호하는 것이다. 일반적인 데이터베이스 보안모델은 임의접근제어(DAC)나 강제접근제어(MAC)를 이용하여 보안문제를 해결하였지만, 이 논문에서는 역할기반 접근제어를 객체지향 데이터베이스에 적용해서 보안문제를 해결한다. 따라서, 기존의 객체지향 데이터베이스의 보안속성과 보안정책에 기반한 주체, 객체, 접근모드들을 정의하였고, 개념적인 클래스를 설계해서 객체지향 데이터베이스의 보안모델을 제시하였다.

1. 서론

지금까지의 객체지향 데이터베이스 보안모델에 대한 연구는 객체지향 패러다임의 각 속성에 대한 보안정책을 정의하는 것으로써 설계가 된다. 객체지향 데이터베이스에서 보안의 목적은 관계형 데이터베이스와 마찬가지로 데이터베이스에 대한 권한이 없는 사용자의 접근을 제어함으로써 정보의 고의적인 파괴나 변경을 방지하고, 우발적인 사고로부터 데이터를 보호하는 것이다[8,9].

객체지향 데이터베이스의 보안에 대한 연구는 관계형 데이터베이스에 기초해서 연구가 되어왔으며, 임의적이나 강제적으로 데이터에 대한 접근을 제어하는 접근제어를 많이 적용시켰다. 객체지향 데이터베이스의 임의접근제어를 적용한 모델에는 ORION[2] 모델이 있고, 강제접근제어를 적용한 모델에는 Millen-Lunt[3]모델이 있다. ORION모델은 주체, 객체, 접근모드에 대해서 각각의 격자구조 형태를 이루고 있으면서, 객체에 대한 접근제어를 객체승인행렬을 통해서 실현하게 된다. 하지만, 임의접근제어정책을 구현한 모델이기 때문에 권한이 부여된 주체가 임의적으로 다른 주체에게 임의적으로 권한을 부여하거나 철회할 수 있다는 단점을 가지고 있다. 따라서, 강제접근제어정책을 적용한 모델들에 대한 연구가 활발히 진행되었다. 그 중의 하나인 Millen-Lunt 모델의 보안등급은 주체와 객체가 생성되거나, 주체와 객체에 대한 접근모드가 재정의 되어질 때마다 부여된다[1,2].

이 논문에서는 관련연구로서 역할기반접근제어에 대한 개념적인 구성요소들을 정의하고, 강제접근제어정책을 사용한 Millen-Lunt모델에 대한 보안공리를 정의한다. 그리고, 이러한 연구를 바탕으로 역할기반접근제어를 통한 객체지향 데이터베이스의 보안모델을 제시하기 위해서 주체, 객체, 접근모드를 정의했다. 그리고, 개념적인 클래스 설계를 통한 보안모델을 정의한다.

† 이 연구는 2001년도 한국통신의 무선망가입자망연구팀의 위탁과제 연구에 의해 수행됨

2. 관련연구

2.1. 역할기반 접근제어(RBAC) 모델

RBAC의 개념은 1970년대에 개발된 온라인 시스템에서 다중 사용자(multi-user)와 다중 응용(multi-application)에 의해서 시작되었다. RBAC의 핵심적인 개념은 권한(Permission)이 역할(Role)과 연관되어 있고, 사용자는 적절한 권한에 배정된다는 것이다. RBAC는 임의접근제어나 강제접근제어를 수행하면서 역할이라고 하는 개념을 사용해서 접근제어를 수행하게 된다[4].

역할은 사용자가 어떤 작업에 대한 권한을 부여받을 수 있는 일종의 권리이므로, RBAC는 현실 세계의 조직체계와 비슷하다. 임의접근제어나 강제접근제어를 통해서 현실 세계를 표현하는데 한계가 있기 때문에 개선책으로 RBAC가 연구되고 있다[5,6].

[그림1]은 96년에 Ravi Sandhu가 정의한 RBAC 모델이다[4].

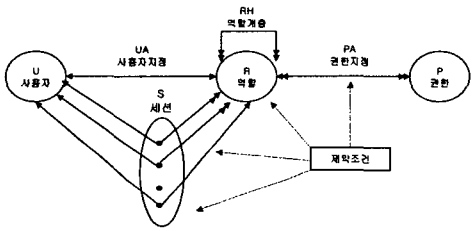


그림 1 역할기반 접근제어 모델

[그림1]에서 RBAC는 개념적으로 보면 크게 사용자(U), 역할(R), 권한(P)으로 구성이 된다.

RBAC 모델은 사용자 집합(U), 세션의 집합(S), 정규권한(P), 권한지정(PA), 사용자지정(UA), 역할계층(RH) 그리고, 제약조건(C)으로 구성된다.

일반적으로 이 모델에서 사용자는 사람을 나타내며, 역할은 사용자와 권한의 집합으로 구성된다. 또한, 역할은 개념적인 조직 내에서의 작업 함수(Job Function) 또는 작업의 제목(Job Title)을 나타낸다. 그리고, 권한은 자원에 대한 특정 접근권한을 나타낸다. 권한은 하나 혹은 그 이상의 자원에 적용될 수 있기도 하다[4].

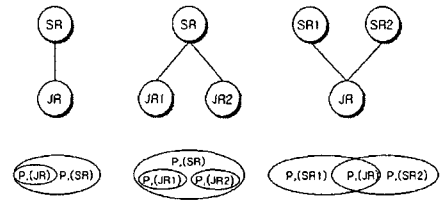
2.2 역할계층

역할계층은 조직내의 권한과 책임을 부분순서로 나타내기 위한 일반적인 방법이며, 격자구조(Lattice structure)로 표현된다.

역할계층의 특징은 상위역할이 하위역할의 권한을 상속받고, 상위역할의 권한을 하위역할이 위임받을 수 있는 구조이다.

역할계층은 다음과 같이 정의가 되어진다.

• RH ⊆ R×R : 역할계층 또는 역할지배관계로 불리는 역할 R에서 부분순서로 표현된다.



(a) SI (b) CS (c) CJ

그림 2 역할의 구성

[그림2]에서처럼 역할계층의 상속에는 단순상속(SI:Simple Inheritance), 공통상위상속(CS:Common Senior inheritance), 공통 하위상속(CJ:Common Junior inheritance)의 세 가지 유형으로 구분된다[7].

2.3 객체지향 데이터베이스

객체지향 데이터베이스의 강제접근제어정책을 수행하는 Millen-Lunt 모델의 보안속성에 대해서 정의한다[3].

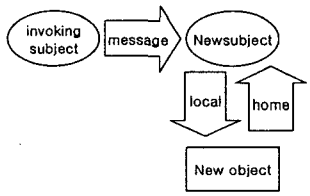


그림 3 주체와 객체의 보안등급

[그림3]은 보안속성에서 사용되는 'invoking' 주체와 'home' 객체를 설명하기 위한 그림이다.

1) 계층 속성

객체의 등급이 객체의 클래스의 등급을 포함해야만 한다. 이것은 객체가 인스턴스로 있는 클래스에 정의되어 있는 메소드와 변수들을, No Read-Up 속성을 위배하지 않으면서 상속할 수 있다는 것을 보장한다.

2) 주체 보안등급 속성

주체의 보안등급이 'invoking'주체의 등급과 'home' 객체의 등급을 포함한다. 이 속성은 'home' 객체와 'invoking' 주체의 최소상위한계의 보안등급과 주체의 보안등급이 같아야 한다는 것을 말한다.

3) 객체의 지역성

주체는 'home'객체에서만 메소드를 실행하고 변수를 읽거나 쓸 수 있다. 이 속성은 주체의 작업이 자신의 'home'객체에만 한정된다는 것을 말한다.

4) *-속성

주체는 자신의 보안등급이 객체의 보안등급과 같을

때에만 'home'객체에 대해서 쓰기 작업을 할 수 있다. 이 속성은 주체가 하위등급에 쓰기 작업을 하는 것을 방지한다.

5) 주체의 반환값 속성

주체는 자신을 호출한 주체의 보안등급과 같을 때에만 호출한 객체에게 반환값을 보낼 수 있다. 이 속성은 상위주체로부터 하위주체로의 정보의 흐름을 방지한다.

6) 객체 생성 속성

새롭게 생성된 객체의 보안등급은 객체의 생성을 요구했던 주체의 보안등급을 포함한다. 이 속성은 주체가 하위등급 객체를 생성하면서 하위등급에 쓰기 작업을 못하도록 한다.

3. RBAC를 적용한 OODB 보안모델

객체지향 데이터베이스에서 사용되는 주체, 객체, 그리고 접근모드에 대한 정의를 하고자 한다.

1) 주체

• 주체(Sub) = {User}

주체는 객체지향 데이터베이스를 사용하고자 하는 일반적인 사용자이다. 즉, 데이터베이스에 접근하고자 하는 일반적인 개념의 주체를 의미한다. 주체(Sub)는 일반적으로 시스템 사용을 원하는 사람을 의미한다.

2) 객체

• 객체(O) = $\{R_1, R_2, ..., R_n\}$

객체는 역할들로 구성된다. 또한, 객체가 직접적으로 자원에 대한 접근모드를 가지기 때문에 RBAC의 역할계층과 유사하다. 그러므로, 일반적인 의미의 객체를 역할로 정의한다.

물론 객체에는 능동객체와 수동객체가 있는데, 능동객체는 다른 객체를 호출해서 메소드를 실행시키고, 수동객체는 이런 능동객체의 호출을 받음으로써 메소드를 실행하는 객체를 의미한다. 이 논문에서 객체의 의미는 두 가지를 모두 포함한다.

하나의 객체인 클래스의 계층적인 구조를 나타내는 것은 역할계층을 이용하여 나타내도록 한다.

3) 접근모드

일반적인 접근제어에서 정의된 권한과 일치하는 갱신(Update), 읽기(Read), 쓰기(Write)의 접근모드를 갖는다.

3.1 보안모델의 개념적 설계

클래스를 포함하는 객체들은 어느 역할에 할당되는지에 따라서 그 클래스가 다른 슈퍼클래스나 서브클래스에 상속을 할 것인지 상속받을 것인지를 결정하게 된다.

따라서, 역할에 할당된 클래스들에 의해서 역할계층을 이용한 상속이 이루어진다. 또한, 역할기반 접근제어를 이용해서 각 객체들에 적절한 보안등급도 부여된다

그리고, 데이터베이스의 자원에 대한 접근모드는 역할기반 접근제어 모델의 권한과 일치하기 때문에 객체지향 데이터베이스에 효율적인 접근제어를 실행시킬 수 있다.

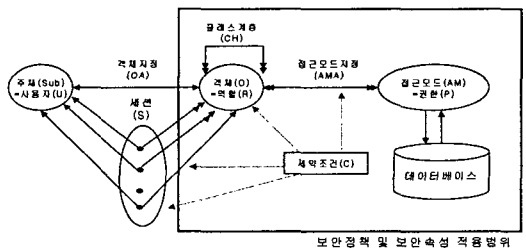


그림 4 RBAC를 적용한 OODB 모델

[그림4]는 역할기반 접근제어 모델에 객체지향 데이터베이스의 주체, 객체, 그리고 접근모드를 정의한 것이다.

4. 객체의 보안속성

4.1 보안속성

1) 객체의 보안등급

$$L(O) \in \{U, C, S, TS\}$$

객체의 등급은 네 가지의 보안등급 중에 하나가 부여된다.

2) 클래스의 보안등급

$$L(Class) \leq L(Attr),\ L(Class) \leq L(Method)$$

클래스의 보안등급은 클래스 내에 정의된 애트리뷰트나 메소드의 보안등급에 의해 지배된다.

$$L(Class) \geq LUB[L(Role), L(Junior\_Role)]$$

클래스의 보안등급은 역할에 따른 접근모드를 갖을 수 있기 때문에 일반적으로 그 역할이 가지고 있는 보안등급 중에서 최소상한을 가지게 된다.

3) 역할의 보안등급

$$L(Role) \geq L(Junior\_Role)$$

역할은 클래스들로 이루어진 집합체로 정의가 되기 때문에 역할의 보안등급은 역할기반접근제어에서 부여할 수 있는 보안등급과 일치한다. 따라서, 자신보다 하위에 있는 쥬니어역할(Junior Role)에 대한 보안등급을 지배하게 된다.

4) 애트리뷰트의 보안등급

$$L(Attr) \geq L(Class)$$

클래스 내의 애트리뷰트는 클래스의 보안등급을 지배한다.

5) 메소드의 보안등급

$$L(Method) \geq L(Class)$$

메소드의 보안등급은 클래스의 보안등급을 지배한다.

6) 상속 계층

객체가 클래스, 애트리뷰트, 메소드로 정의가 되고, 애트리뷰트나 메소드는 클래스에 속한 속성들이기 때문에 클래스에 대한 계층 구조를 적용시켜서 보안등급을 정의하게 된다.

4.2 클래스 구조

클래스에 대한 정의를 [표1]과 같이 정의함으로써 객체의 상속, 역할할당, 제약조건 등을 만족시킬 수 있다.

표 1 클래스 구조

```
Class Class_Name
        OID : Unique number(or string)
        Role : Set of Role
        Attribute :
                attribute_name : type
        Method : method_name
                constraint_check();
End Class
```

[표1]은 객체지향 데이터베이스를 설계함에 있어서 개념적인 클래스 구조를 나타낸다.

1) 객체 식별자(OID)

객체 식별자는 객체에 부여하는 고유번호이며, 역할에 따른 상속을 가능하게 해준다.

2) 역할(Role)

어느 역할에 클래스가 할당되었는지를 나타낸다.

3) 메소드

역할기반 접근제어에서 정의하고 있는 아래 두 가지의 제약조건(constraint)에 대해서 검사를 수행한다.

- 의무분리 : 한 클래스가 서로 배제적인 슈퍼클래스로부터 애트리뷰트나 메소드를 상속받는지를 검사한다.
- 카디널리티 조건 : 접근모드에 대한 역할(클래스)의 개수를 제한한다.

5. 결론 및 향후연구

객체지향 데이터베이스를 위한 보안모델들은 많이 제시되었다. 하지만, 임의접근제어나 강제접근제어를 적용함으로서 두 개의 접근제어 정책이 가지고 있는 보안에 대한 취약성을 보완하지 못했다.

또한, 지금까지 역할에 따른 접근제어를 시도한 보안모델에 대한 연구가 거의 이루어지지 않았다. 따라서, 이 논문에서는 역할기반접근제어를 적용한 객체지향 데이터베이스의 보안모델을 제시하였다.

이 논문에서 제시되었던 보안속성이나 보안등급부여 속성으로는 복잡한 실세계의 객체들을 모두 나타낼 수 없다. 따라서, 향후 연구로는 복합객체, 다중상속, 버전 등의 개념을 역할접근제어에 적용하는 것이다.

6. 참고문헌

[1] James M. Slack, "Security In An Object-Oriented Database", ACM SIGSAC on New Security Paradigms Workshop, pp.155-159, 1993.

[2] Jay Banerjee, Hong-Tai Chou, Jorge F. Garza, Won Kim, Darrell Woelk, and Nat ballou, "Data Model Issues for Object-Oriented Application", ACM Transactions on Office Information Systems, Vol.5, No.1, Jan., 1987.

[3] Jonathan K. Millen, Teresa F. Lunt, "Security for Object-Oriented Database Systems", IEEE Computer Society Symposium , pp.260-272, 1992.

[4] Ravi Sandhu, Edward coyne, Hal Feinstein, Charles Youman "Role-Based Access Control Models" IEEE Computer, Volume 29, number 2, Feb. 1996.

[5] C. Ramaswamy and R. Sandhu, "RBAC Features in commercial Database Management Systems", NISSC, 1998.

[6] D. Ferraio, J. Cugini and R. Kuhn, "RBAC: Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.12.

[7] Matunda Nyanchama, "Commercial Integrity, Roles and Object Orientation", Univ. of Western Ontario, Phd thesis, Sep, 1994.

[8] 주광로, 박우근, "데이터베이스 시스템의 보안기술", 정보처리학회논문지, 제4권, 제2호, pp.33-43, 1997.

[9] 노봉남, 김용성, 장옥배, "다단계 객체지향 계층구조에서 상속의 보안 성질", 정보과학회논문지, 제20권, 제9호, 1998.