데이터베이스를 이용한 RBAC(역할기반 접근제어) 서버 API 구현

김진식○ 김민영 이상원○
성균관 대학교 VLDB 데이터베이스 연구실
{x1m2n3o4p○ , inline0208, swlee○}@skku.edu

An Implementation of Hierarchical RBAC(Role Based Access Control) API using Database

Kim-Jin Sik○ , Kim-Min Young, Sang-Won Lee○
VLDB Lab. , Sungkyunkwan University

요 약

RBAC(Role Based Access Control) 이란 특정 사용자가 어떤 대상에 특정 행동을 하는 데에 있어서 그 사용자가 가진 역할 (Role) 에 의해 접근 가능유무를 판정하게 하는 방법이다. 그 RBAC 에 역할간의 계층관계를 추가한 것이 계층적 RBAC (Hierarchical RBAC)이다. 본 논문에서는 그런 다른 어플리케이션에 쉽게 추가 되거나 아니면 독자적으로 인증 기능을 가지는 계층적 RBAC 서버에 사용될 수 있는 API와 그와 관련된 응용 어플리케이션을 자바와 데이터베이스를 이용하여 설계 및 구현하였다.

1. 서 론

1.1 RBAC의 필요성

최근 데이터베이스 및 네트워크 시스템 관리, 보안 관리에 있어서 역할기반 접근제어(Role-based Access Control)에 기반을 둔 응용 프로그램이나 RBAC 서버를 구현하는 기관들이 늘고 있다. 이것은 데이터베이스에 접근하는 사용자가 많아짐에 따라 관리자로 하여금 데이터베이스를 효과적으로 관리하게 하기 위한 하나의 수단이 되었다. RBAC을 사용하게 되면 각각의 사용자들을 역할에 따라 구분하고 그 역할에 특정 접근권한을 줌으로써 사용자를 관리할 수 있다. 사용자들은 자신이 가진 역할에 의한 권한만 가질 수 있기 때문에 보안 수준도 높아지게 된다.

1.2 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 RBAC 의 정확한 의미와 장점과 분류에 대해서 간략히 정리한다. 3장에서는 본 논문에서 개발한 RBAC 서버 API와 서버 API를 이용한 RBAC 관리자 툴, 웹에서의 간단한 사용 예를 설명한다. 마지막으로 4장에서는 결론 및 향후 계획에 대해서 논의한다.

2.관련 연구

2.1 RBAC의 정의[1]

접근 제어 (Access Control)란 말은, 어떤 능력을 물리적으로 또는 시스템 기반으로 활성화시키거나 제한시키는 방법을 뜻한다. 컴퓨터 기반 접근 제어는 누가, 또는 어떤 프로세스가 특정 시스템 리소스에 대하여 접근을 할 것인지 뿐만 아니라, 허가된 접근의 형태까지 규정할 수 있다.

RBAC (Role-based Access Control), 즉 역할 기반 접근 제어는 한 기관의 부분으로서 개인 사용자들이 가지는 역할을 기반으로 접근 결정을 내리는 제어 방법이다. 접근 권한은 역할 이름에 의해 그룹화 되고, 리소스의 사용은 그와 관련된 역할을 허가 받은 사용자들에게로 제한된다.

병원 시스템을 예로 들면, 의사의 역할에 환자를 진단하는 일, 약을 처방하는 일, 실험 결과를 요청하는 일 등을 포함시킬 수 있고, 연구원의 역할에는 연구를 위한 익명 치료 정보 수집만으로 제한할 수 있다.

접근 제어를 하기 위해 역할을 사용하는 것은 보다 변화에 유연한 보안 정책을 제공하고, 능률적인 보안 관리에도 도움이 된다.

2.2 NIST에서 제안한 RBAC 모델

NIST[1]에서는 기존의 RBAC 모델들을 정리하고 표준 참조 모델을 제안하였다. 그 이유는 현재까지 기준이 되는 표준 모델의 부재로 역할기반 접근통제를 구성하는 특성들에 대한 일반적인 합의 없이 RBAC 모델 개발이 계속되고, RBAC 용어 자체도 사용자 및 개발자들 사이에서 다양한 방식으로 사용되고 있으며 역할에 대해서도 다양한 방식으로 구현이 이루어지고 있는데, 이는 RBAC 서버를 API를 구현할 때 혼란을 야기시킬 여지가 많기 때문이다.[2]

NIST에서 제안한 RBAC 표준 참조 모델은 네 가지 모델로 정의된다.[3] 본 논문에서 소개할 RBAC API는 Core RBAC과 Limited Hierarchical RBAC을 기반으로 구현을 하였기 때문에 이 두 RBAC 모델을 설명하겠다.

1) National Institute of Standards and Technology

2.2.1 Core RBAC

Core RBAC은 RBAC 모델의 기본이 되는 모델이다. 사용자가 어떤 객체에 대한 기능 즉, 권한을 직접 요구하는 것이 아니라 사용자가 원하는 권한을 가지고 있는 역할에 할당됨으로써 그 역할의 권한을 갖게 되는 것이다. 역할은 사용자와 권한 사이에서 중재역할을 하면서 인증되지 않은 사용자가 특정 객체에 접근하는 것을 통제한다.

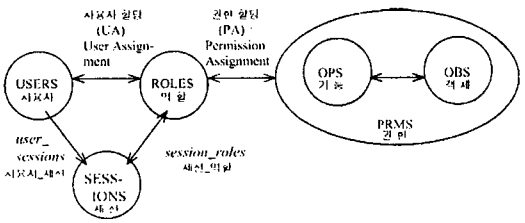

그림 1. Core RBAC

그림 1은 Core RBAC을 도식화한 것이다. Core RBAC은 사용자, 역할, 세션, 기능, 객체 (대상),권한 등의 구성요소로 이루어져 있다. 각각의 구성요소의 관계를 살펴보면, 한 명의 사용자가 다수의 역할을 가질 수 있고, 반대로 여러 명의 사용자가 하나의 역할을 공유할 수도 있으므로 사용자와 역할은 다 대 다 관계이다. 마찬가지로 하나의 역할이 다수의 권한을 가지거나 다수의 역할이 하나의 권한을 공유할 수 있으므로 역할과 권한 사이에도 다대다 관계가 성립한다. 마지막으로 역할과 세션 또한 한 명의 사용자가 접속을 하더라도 여러 개의 역할이 활성화될 수 있기 때문에 다대다 관계이다.

2.2.2 Limited Hierarchical RBAC

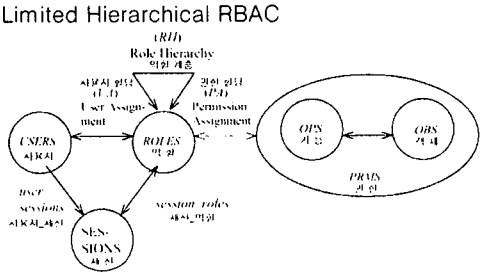

그림 2. Hierarchical RBAC

Hierarchical RBAC은 Core RBAC에 역할 계층 구조 개념을 추가한 모델이다. 상위 역할은 자신이 가지고 있는 하위 역할을 포함한다. 즉, 하위 역할이 가진 모든 권한을 가진다. 그림 3은 그러한 예를 보여준다.

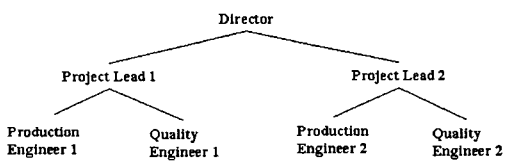

그림 3. 역할 계층 예제

Hierarchical RBAC에서 다중 상속을 제한한 것, 즉 하위 역할이 하나의 상위 역할에만 상속되도록 한 것을 Limited Hierarchical RBAC이라 한다. 따라서 이 구조를 도식화하면 트리로 나타난다.

3.API 설계 및 구현

3.1 RBAC API

본 논문의 RBAC API 는 자바와 데이터베이스의 프로시저로 구성되어 있으며 저장매체로 관계형 데이터베이스를 사용한다. NIST에서 제안한 표준안 중에서 활용 가능성이 높은 트리형태의 Limited RBAC 모델을 선정하여 함수명세나 제약조건을 그대로 반영하였다.

3.1.1 데이터베이스/프로시저

그림 4는 데이터베이스 스키마를 모델링 한 ER 다이어그램이다. 역할 계층 구조를 위해 Role 간의 부모자식 관계를 표현하는 Role_Role 테이블이 존재한다.

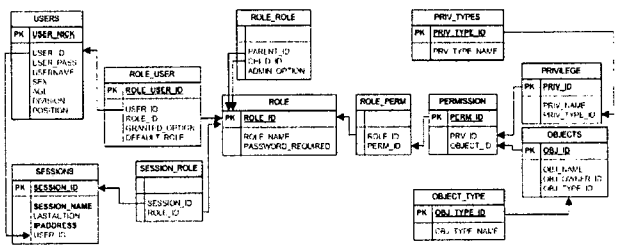

그림 4. RBAC ER Diagram

3.1.2 자바

그림 5는 실제 RBAC API에서 사용되는 대표적인 클래스들을 간략화한 클래스 다이어그램이다. 앞의 그림2에서 표현되고 있는 RBAC의 구성요소들을 각 클래스로 표현하였으며 구성요소들 간의 관계도 아래와 같이 표현하였다.

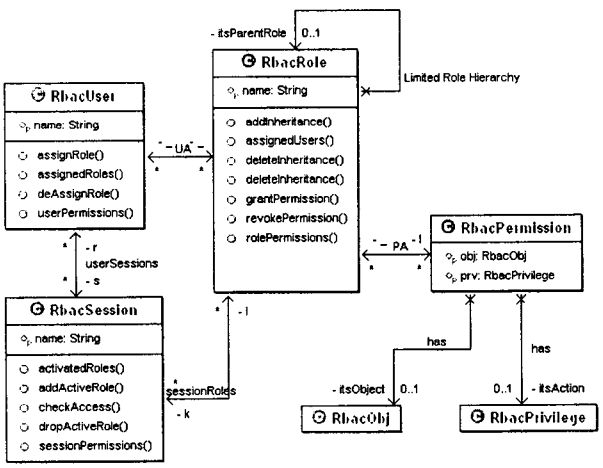

그림 5. RBAC API 기본 클래스 다이어그램

위의 클래스 다이어그램에서, Permission은 한 쌍의

Obj 와 Privilege로 구성된다. 여기서 Obj는 관리될 수 있는 대상을 뜻하고 Privilege는 그 대상에 정의될 수 있는 어떤 행동(Action) 을 뜻한다. 또한 역할과 역할 간의 연관 관계도 계층 관계로 존재함을 알 수 있다.

3.1.3 계층 질의문

```
select p.perm_id,p.prv_id,p.object_id
from role r,role_perm rp,permission p
where r.role_id = rp.role_id
and   p.perm_id = rp.perm_id
and  r.role_id in (select child_id
                   from role_role
                   start with parent_id = ?
                   connect by prior child_id = parent_id
                  )
```

위의 질의문은 RbacSession 객체의 checkAccess 메서드 내부에서 사용 하는 질의문이다. 제한된 계층적 RBAC으로 구성되어 있기 때문에 기존의 상용 데이터베이스에 지원하는 계층형 질의문으로 요청을 하면 좀 더 효과적으로 한 역할의 자식이 되는 역할들을 수집할 수 있다.

3.2 관리자 도구

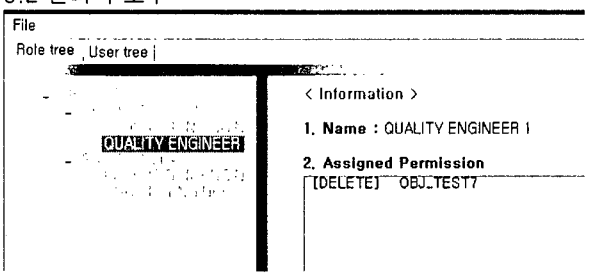

그림 6. 역할 계층 관계 표현

그림 6은 관련연구 부분의 그림 3을 관리자도구를 사용하여 생성한 모습이다. 이 작업창에서 직접 하위 역할을 추가 하거나 한 역할을 다른 역할에 끌어다 놓음으로서 역할 계층관계를 동적으로 바꿀 수 있다. 오른쪽 부분은 간단히 그 역할에 대한 정보를 표시하고 각각의 역할에 여러 개의 허가를 포함시킬 수 있는 기능을 가지고 있다. (PA: Permission Assignment 표현)

위의 그림에서 예로 "Quality Engineer"라는 역할은 "OBJ_TEST7" 이라는 대상에 대해서 "DELETE"라는 권한을 가지고 있음을 알 수 있다. 또한 "DIRECTOR" 와 "PROJECT LEAD1" 이라는 권한은 "QUALITY ENGINEER" 의 부모이기 때문에 자동으로 이 권한을 가지게 된다.

두 번째 탭인 User Tree 에서는 간단히 존재하는 사용자의 목록을 나열하여, 각각 사용자가 현재 가지고 있는 역할(Roles)과 허가 (Permissions)를 찾아 볼 수 있는 기능을 제공한다. (UA : User Assignment 관계 표현)

3.3 웹 환경에서 이용 예시

RBAC 은 웹 환경에서 권한을 조정하는 데에 유용하게 쓰일 수 있다. 기존의 웹에서는 권한관리에 대해서 대부분 정적으로 구성되어 있으며 약간의 변화에 대한 수정을 하는 데에도 많은 시간이 든다. RBAC 서버를 사용하여 인증을 하려면 보안이 필요한 페이지에 그 대상과 행동을 적어서 RBAC 서버에 요청을 하면 된다. 이렇게 웹을 구성하게 되면 역할을 구성하는 허가의 범주가 변하거나 어떤 역할이 그 부분과 관계되는지에 대해 웹 작성자는 신경 쓰지 않아도 되므로 편리하다.

```
RbacPrivilege priv = new RbacPrivilege(권한이름)
RbacObject obj =    new RbacObj(대상이름)
WebSecurityManager.isUserAuthenticated(user,rp)
```

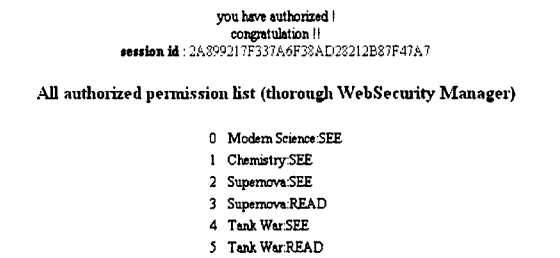

그림 7 사용자의 현재 세션의 가능 허가 목록

4. 결론 및 향후 계획

지금까지 RBAC 의 정의와 왜 필요한지에 대해 알아보았고, RBAC 의 인증방법을 독자적으로 처리할 수 있는 향후 RBAC 인증 서버에서 쓰일 수 있는 자바 API Package와 그 API에 관련되어 있는 데이터베이스 구조 등의 구현 상황을 보였다. 추가로 이 API를 이용한 예로서 RBAC 서버 관리자가 사용할 수 있는 관리자 도구와 웹에서의 간단한 사용 예를 보였다. 그러나 이것은 현재로서는 단순한 API 일 뿐이고, 실용적인 서버의 역할을 하기 위해서는 이 API가 분산처리 객체형식으로 구성하여야 할 것이다. 또한 다수의 사용자가 이 서버를 사용한다면 많은 처리를 할 수 있도록 하기 위해 많은 테스트와 최적화 과정을 거쳐야 할 것이다.

5. 참고 문헌

[1] An Introduction to Role-Based Access Control, NIST/ITL Bulletin, December, 1995 (http://csrc.nist.gov/rbac/NIST-ITL-RBAC-bulletin.html)

[2] 김학범, 김동규 "RBAC 표준 참조 모델 연구동향" (情報保護學會誌, Vol.10 No.2, [2000]). p.2

[3] D.F.Ferraiolo, Ravi Sandhu, Serban Gavrila, D.Richard Kuhn and Ramaswamy Chandram ouli, "Proposed NIST Standard for Role-Based Access Control"