한국산학기술학회논문지
Vol. 7, No. 6, pp. 1134-1141, 2006

자바 애플리케이션을 위한 역할기반 접근제어 패키지의 설계 및 구현

오세종[1*]

Design and Implementation of Role-Based Access Control Package for Java Applications

Se-Jong Oh[1*]

요 약 자바는 실행 플랫폼에 독립적이고 모바일 분야(J2ME)로부터 엔터프라이즈 환경(J2EE)에 이르는 다양하고도 일관된 솔루션을 제공하여 이기종 분산 애플리케이션 환경을 위해 적합한 개발 도구로 평가받고 있다. 본 연구에서는 자바 애플리케이션을 개발할 때 필요로 하는 접근제어 모듈을 자바 패키지 형태로 구현하여 제공함으로써 애플리케이션의 개발 기간을 단축시키고 시스템 관리자들이 보다 편리하게 접근제어 관리를 할 수 있도록 하였다. 구현된 모듈은 역할기반 접근제어(RBAC) 모델을 기초로 하고 자바 패키지 및 접근제어 관리도구를 포함한다.

Abstract Java is platform-independent and supports uniform solutions from mobile area (J2ME) to enterprise area (J2EE), so Java is a good development tool for the environment of heterogeneous machines and distributed applications. Java applications need access control module as a Java package. In this paper, we design and implement it. Therefore Java developers can reduce development time, and system managers easily do access control work. Proposed module is based on Role-Based Access Control (RBAC) model and includes a Java package and administration tool.

Key words : Java, Java package, Role-Based Access Control, Security

1. 서 론

오늘날 애플리케이션의 개발은 서로 다른 운영체제를 가지고 지역적으로 분산되어 있는 컴퓨팅 환경에서 이루어지는 경우가 많다. 또한 인터넷의 발전으로 웹 애플리케이션이 확산되어가는 추세이다. 자바는 실행 플랫폼에 독립적이고 모바일 분야(J2ME)로부터 엔터프라이즈 환경(J2EE)에 이르는 다양하고도 일관된 솔루션을 제공하며, 웹 애플리케이션 환경을 위해서도 적합한 개발 도구로 평가받고 있다. 자바에서 사용자 인터페이스를 위해서는 JSP가, 비즈니스 로직(business logic)을 위해서는 서블릿(servlet)이나 EJB가 이용될 수 있다.

기업 환경을 위한 정보시스템과 같이 그 특성상 다수의 사용자가 다수의 자원을 이용하는 자바 애플리케이션의 경우 사용자들에게 부여된 권한에 따라 접근을 제한하는 일이 필수적이다. 정보 시스템의 규모가 커지고 시스템을 이용하는 사용자 수가 급속도로 증가함에 따라 권한이 있는 사용자라 할지라도 그 부여 받은 권한의 내용과 정도에 따라 시스템의 자원에 대한 이용을 통제해야할 필요가 커지고 있다[1]. 사용자들은 시스템 환경에 따라 여러 그룹으로 분류가 될 수 있으며 각 그룹은 서로 다른 이용 권한을 가지게 된다. 예를 들면 사장이 볼 수 있는 정보와 부장, 과장, 일반 사원이 볼 수 있는 정보는 서로 다르게 관리된다. 그리고 웹 애플리케이션 환경에서 우수회원은 일반 회원에 비해 보다 많은 서비스를 이용할 수 있어야 한다. 이와 같이 사용자에게 부여된 권한에 따라 시스템에 대한 이용을 제한하는 행위를 접근제어(access control)라고 한다[2]. [그림 1]은 접근제어의 개념에 대한 설명이다.

이 연구는 2005학년도 단국대학교 대학연구비의 지원으로 연구되었음.

[1]단국대학교 컴퓨터과학전공
*교신저자: 오세종(sejongoh@dankook.ac.kr)

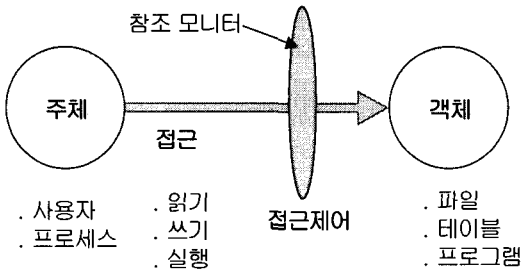


그림 1. 접근제어의 개념[2)]

자바 애플리케이션 개발자들은 애플리케이션을 개발할 때 [그림 1]과 같은 기능을 수행하기 위한 접근제어 모듈을 애플리케이션 내에 포함해서 개발하여 왔다. 즉, 애플리케이션은 사용자, 접근 대상(객체), 사용자의 접근 대상에 대한 권한 정보를 저장하고 관리하며, 접근제어를 수행하는 참조 모니터 부분을 포함하고 있다. 그 결과 접근 제어 모듈의 변경 및 향상이 쉽지 않고, 새로운 애플리케이션을 개발할 때 마다 매번 접근 제어 모듈을 구현해야 하는 번거로움이 있었다.

본 논문에서는 자바 개발자들이 필요로 하는 접근제어 모듈을 자바 패키지(package)의 형태로 구현하여 제공함으로써 개발자들이 접근제어 모듈을 별도로 구현할 필요가 없도록 하였고 사용자, 객체, 권한 등에 대한 정보를 관리할 수 있는 지원 도구를 제공하여 애플리케이션이 수행될 때 시스템 관리자가 보안관리를 용이하게 할 수 있도록 하였다. 제공되는 패키지는 완전한 API 문서를 포함하여 별도의 설명서가 없이도 개발자들이 패키지를 사용할 수 있도록 하였다. 본 연구의 패키지는 역할기반 접근제어(RBAC) 모델을 기초로 한다. RBAC 모델은 기업과 같은 비즈니스 환경에 적합한 모델로 알려져 있다.

본 논문의 구성은 다음과 같다. 2장에서는 자바의 보안 관련 사양 및 접근제어에 관련된 연구를 살펴본다. 또한 접근제어 모델에 대한 구현 사례를 살펴본다. 3장에서는 RBAC 모델에 기초한 접근제어 패키지의 설계 및 구현 내용을 소개한다. 먼저 설계 및 구현의 개요에 대해 설명하고 자바 패키지, API 문서, 접근제어 지원 도구에 대해 설명한 다음 4장에서 결론을 맺는다.

2) [그림 1]에서 참조 모니터(reference monitor)란 실제 접근제어를 수행하는 프로세스 혹은 시스템을 의미한다. 참조 모니터는 주체가 객체에 접근을 할 때 이를 탐지하여 자신이 가지고 있는 권한 정보와 비교한 뒤 접근에 대한 허용 여부를 결정한다.

2. 관련 연구

사용자가 많고 접근 대상이 되는 정보 객체가 많은 애플리케이션 환경일수록 접근제어가 중요한 보안 이슈가된다. 지금까지 여러 가지 접근제어 모델들이 개발되었는데 대표적인 것으로는 접근제어 리스트(ACL: Access Control List) 모델, 자율적 접근제어(DAC: Discretionary Access Control) 모델, 강제적 접근제어(MAC: Mandatory Access Control) 모델 등이 있다 [1,2]. 접근제어 리스트는 Unix나 Window 같은 운영체제에 사용되며, 사용자에게 부여된 권한 리스트에 따라 접근제어를 수행한다. 자율적 접근제어 모델에서는 정보 객체마다 소유자(owner)가 존재하며 소유자들이 자율적으로 사용 권한을 다른 사용자에게 부여하거나 회수할 수 있는 모델이다. 강제적 접근제어는 군사환경과 같이 엄격한 정보보호가 필요한 환경에서 사용된다. 역할 기반 접근제어(RBAC: Role-Based Access Control) 모델은 사용자(user)들에게 직접 권한을 할당(assign)하던 기존의 모델들과는 달리 [그림 2]와 같이 현실세계에서 수행하는 업무적 역할에 따라 인가권한(permission)을 역할(role)에 할당하고, 사용자들은 적당한 역할에 소속되도록 함으로써 사용자들의 권한 관리를 효율적으로 할 수 있도록 지원한다[3,4]. 또한 RBAC에서는 사용자-역할 할당(URA) 정보 및 인가권한-역할 할당(PRA) 정보를 관리하고 역할들 간의 계층(role hierarchy)을 정의하기 위한 역할-역할 할당(RRA) 정보를 관리한다. 본 연구에서는 기업 환경에 적합한 것으로 알려진 RBAC 모델을 구현 대상으로 삼았다.

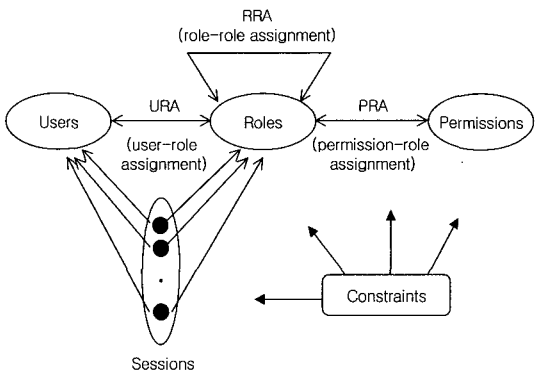


그림 2. 역할기반 접근제어(RBAC) 모델 개요

자바는 구조적으로 애플리케이션이 안전하게 실행될 수 있도록 하는 여러 단계의 메카니즘을 가지고

있다[5,6]. 또한 API를 통해 [표 1]에 있는 바와 같이 애플리케이션 보안을 위한 여러 서비스들을 제공한다 [7-9]. 접근제어와 관련해서는 java.security.acl 패키지를 통해 앞에서 설명한 접근제어 리스트 (ACL) 모델을 지원한다. 그러나 java.security.acl 패키지에는 접근제어에 필요한 사용자, 사용자의 인가권한(permission) 정보만 관리할 수 있도록 하였기 때문에 이 정보를 이용하여 실제 접근 제어를 구현하는 것은 개발자의 몫이다. 또한 서블릿을 이용하는 환경에 한해 제한적으로 RBAC 모델을 지원하고 있는데, 사용자-역할 할당(URA) 정보 및 역할-인가권한 할당(PRA) 정보를 XML 형태로 저장할 수 있도록 하고 tomcat과 같은 서블릿 컨테이너가 접근제어를 수행하도록 하였다. 그러나 역할계층(role hierarchy)과 같은 RBAC 모델의 핵심 기능을 지원하지 않고 서블릿 환경에서만 실행이 된다는 점에서 본 연구에서 구현한 패키지와는 차이가 있다.

표 1. 자바에서 제공하는 주요 보안 기능

서비스명	설 명
암호화 (Encryption)	패키지를 통해 비대칭 암호화, RSA 암호화, 공개키 암호화 등을 제공한다 (JCA, JCE)
인증 (Authenticati on)	메시지 혹은 사용자의 신뢰성 확인을 위해 사용되며 message digest, message authenti cation codes, digital certificates 등을 지원한다. (java.security, java.security .cert)
SSL	웹브라우저와 웹서버간의 안전한 커뮤니케이션을 위해 사용됨 (JSSE: java secure sockets extension)
접근제어	사용자 인증 및 권한 확인에 사용 (JAAS, Servlet, EJB 보안 모델)

앞에서 설명한 자바의 보안 기능을 이용하여 안전한 시스템을 구현 사례들이 많이 있다 [10-14]. 접근제어를 구현한 사례로는 jGuard 프로젝트가 대표적인데 웹 애플리케이션 환경을 대상으로 하고 있으며 JAAS와 RBAC 접근제어 기능을 통합하려고 시도 하였다 [15]. Giuri은 유연한 접근제어 정책을 자바에 구현하는 방법을 보였다[16,17]. Hauswirth는 웹 환경에서 접근제어 개발 도구를 구현 하였다[18].

앞에서 살펴본 자바 환경에서 RBAC에 기초하여 접근제어를 구현한 사례들은 대부분 자체 시스템의 일부로서 구현 하였거나 웹과 같은 특정 환경을 위한 것이다. 또한 접근제어의 기초가 되는 사용자, 역할, 인가권한 등의 데이터를 쉽게 입력하고 관리할 수 있는 방법을 제공하지 않고 RBAC의 핵심적인 개념들을 충분히 지원하지 못하고 있다. jGuard의 경우에도 역할을 단순한 그룹(group)의 개념으로 다루고 있다. 따라서 RBAC 모델을 충실히 지원하면서도 개발자들이 쉽게 가져다 적용할 수 있는 접근제어 모듈이 필요하다. 이에 따라 본 연구에서는 RBAC 모델을 구현한 자바 패키지를 제공하여 개발자들이 쉽게 접근제어 모듈을 구현할 수 있도록 하였다.

3. 접근제어를 위한 자바 패키지의 설계 및 구현

3.1 개요

본 연구를 위한 개발 환경은 다음과 같다.

- 운영체제 : Windows XP
- Java : JDK 1.5.0
- 개발 도구 : Eclipse SDK 3.5.0

본 연구의 개발 범위는 [그림 3]과 같다. [그림 3]에서 짙은색 부분이 본 연구에서 개발한 부분이다. 본 연구는 역할기반 접근제어를 위한 모듈을 자바 패키지의 형태로 지원하는 것을 1차 목표로 하고 있다. 패키지의 이름은 sec.ac.rbac이며 JAR 파일로 압축하여 배포한다. sec.ac.rbac 패키지는 RBAC 접근제어를 위해 필요한 기본 데이터를 관리하기 위한 클래스들 및 사용자 로그인, 접근 제어 수행을 위한 클래스를 포함하고 있다. 접근제어에 필요한 기본 데이터(사용자, 역할, 인가권한 정보 등)은 파일 또는 데이터베이스의 테이블에 저장되며, 접근제어 모듈이 실행되면 주기억장치로 로드된다. 본 연구에서는 또한 구현된 sec.ac.rbac 패키지를 이용하여 시스템 관리자가 접근제어에 관련된 정보를 용이하게 관리하고 로그인한 사용자를 모니터링할 수 있도록 관리 도구를 구현하였다. 관리도구 역시 자바를 이용하여 구현하였으며 sec.ac.rbac 패키지가 어떻게 활용될 수 있는지를 보여주는 사례이다. sec.ac.rbac 패키지는 각각 다른 실행환경을 지원하기 위해 설정(setup) 파일을 참조하는데, 설정 파일의 이름은 'Res_rbac.properties' 이고 자바의 클래스경로(classpath)로 지정된 디렉토리에 위치해야 패키지에서 참조할 수 있다. 본 연구에서는 개발자들이 sec.ac.rbac 패키지를 쉽게 이해하고 사용할 수 있도록 표준 API 문서를 작성하였으며, 패키지와 함께 jar 파일 형태로 압축하여 배포한다.

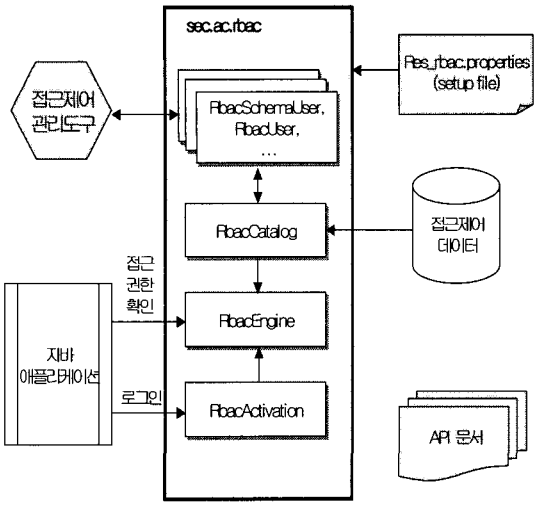


그림 3. 연구의 개발 범위

3.2 RBAC 접근제어 패키지의 구성

[표 2]는 sec.ac.rbac 패키지에 포함된 클래스들에 대한 목록이다. [표 2]에서 그룹 A의 클래스들은 RBAC 접근제어 모델을 구성하는 각 요소들에 대한 속성 정보(자료 구조)를 정의하기 위한 것이다. [그림 4]는 첫 번째 그룹의 클래스 중 RbacSchemaUser의 내용이다. 이 클래스는 사용자에 대해 관리하는 속성을 정의하고 있으며, 정의에 따르면 본 패키지에서는 사용자에 대해 사용자ID(userID)와 비밀번호(passwd) 속성을 관리한다.

```
package sec.ac.rbac;
public class RbacSchemaUser {
        public String userID;        // User ID
        public String passwd ;       // Password
}
```

그림 4. RbacSchemaUser 클래스의 내용

표 2. sec.ac.rbac 패키지의 클래스 요약

그룹	클래스명	설 명
A	RbacSchemaUser	사용자에 대한 속성(property) 정보
	RbacSchemaRole	역할에 대한 속성 정보
	RbacSchemaObject	접근대상에 대한 속성 정보
	RbacSchemaPerm	인가권한에 대한 속성 정보
	RbacSchemaUra	사용자-역할 할당에 대한 속성 정보
	RbacSchemaPra	인가권한-역할 할당에 대한 속성 정보
	RbacSchemaRra	역할-역할 할당에 대한 속성 정보
	RbacSchemaActivation	로그인 한 사용자에 대한 속성 정보
B	RbacUser	사용자 객체를 위한 클래스
	RbacRole	역할 객체를 위한 클래스
	RbacObject	접근대상 객체를 위한 클래스
	RbacUra	사용자-역할 할당 객체를 위한 클래스
	RbacPra	인가권한-역할 할당 객체를 위한 클래스
	RbacRra	역할-역할 할당 객체를 위한 클래스
C	RbacActivation	로그인 한 사용자 객체를 위한 클래스
	RbacCatalog	객체들에 대한 정보를 관리하기 위한 클래스
	RbacEngine	접근제어 수행 결과를 알려주는 클래스

[표 2]에서 그룹 B의 클래스들은 RBAC 접근제어 모델을 구성하는 각 요소들에 대해 사용 가능한 메소드(method)들을 정의하기 위한 것이다. [표 3]은 두 번째 그룹의 클래스 중 RbacUser가 제공하는 메소드들의 목록을 보여준다. 자바 개발자들은 이러한 메소드들을 이용하여 RBAC 접근제어에 필요한 정보를 관리(입력, 수정, 삭제)할 수 있다.

[표 2]에서 그룹 C의 클래스들은 RBAC 모델에 근거하여 실제 접근제어를 수행할 때 필요하다. RbacActivation 클래스는 사용자가 로그인 했을 때 생성되는 세션을 관리하기 위한 것으로 로그인하면서 활성화한 역할 정보를 담고 있다. RbacCatalog 클래스는 접근제어 모듈이 실행될 때 접근제어 데이터에 대한 디스크 I/O를 줄이기 위해 접근제어 데이터를 주기억 장치로 로드하여 저장하는 역할을 한다. 이러한 방법은 접근제어를 빠르게 수행하기 위해 필수적이다. RbacEngine 클래스는 실질적으로 접근제어를 담당하며 자바 애플리케이션에서 (사용자, 역할, 인가권한) 정보를 주고 접근을 요청하면 RbacCatalog 및 RbacActivation를 참조하여 허가 여부를 판단하여 알려 주는 역할을 한다. 세 번째 그룹의 클래스들은 접근제어 모듈 실행시 단 하나의 인스턴스(instance)만 생성되도록 설계 하였다.

표 3. RbacUser 클래스의 메소드 요약

메소드명	설 명
addUser()	새로운 사용자 객체를 추가한다
updateUser()	사용자 객체의 비밀번호를 수정 한다
deleteUser()	사용자 객체를 삭제한다
getUserList()	등록된 사용자들의 목록을 가져온다
getRowCount()	현재 등록된 사용자의 인원수를 가져온다
isExistUser()	사용자가 등록되어 있는지를 확인한다
isValidUser()	사용자의 비밀번호가 올바른지를 확인한다.

3.3 API 문서의 작성

자바 패키지가 개발자들에 의해 사용될 수 있기 위해서는 패키지의 내용에 대한 자세한 API 문서가 제공되어야 한다.

본 연구에서는 자바의 표준 API 문서 양식에 따라 각 클래스, 속성, 메소드들에 대한 API 문서를 작성하여 제공한다. [그림 5]는 API 문서의 일부를 보여준다.

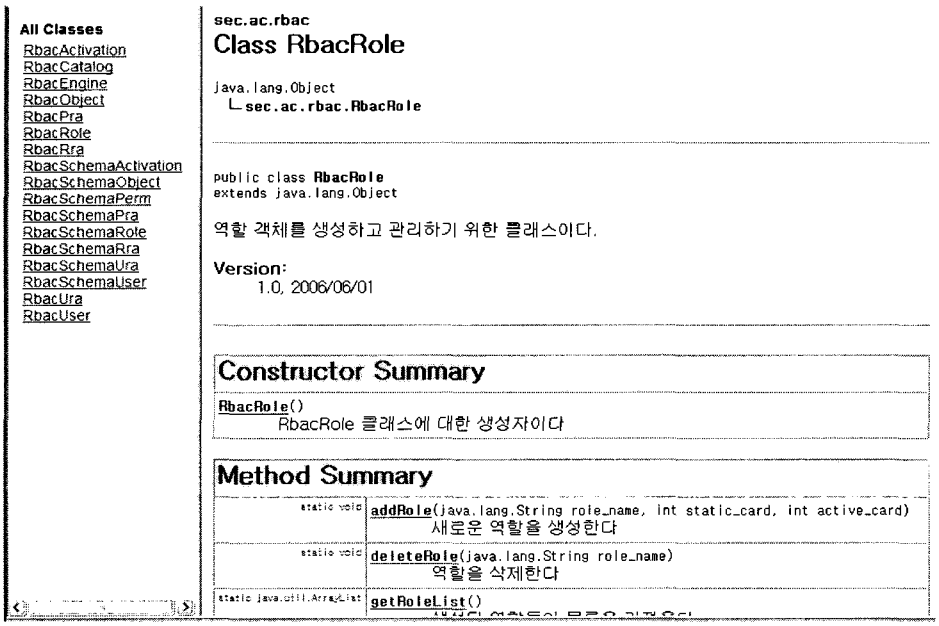


그림 5. API 문서의 예

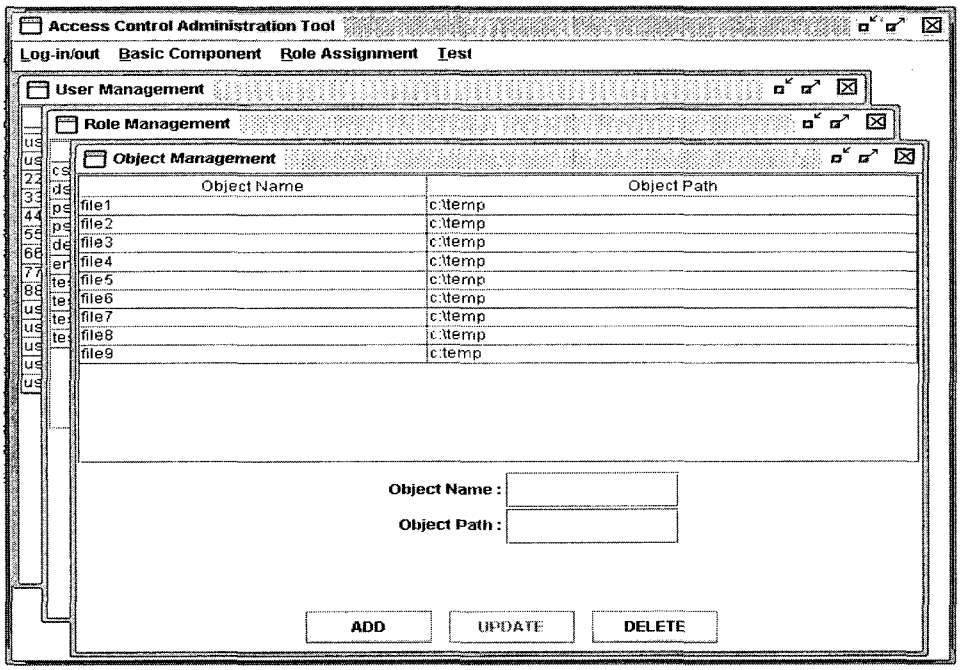


그림 6. 지원 도구 화면

3.4 지원 도구의 개발

sec.ac.rbac 패키지는 RBAC 접근제어를 수행하기 위한 최소한의 기능들로 구성되어 있다. 자바 애플리케이션 개발자들은 이 패키지를 이용하여 접근제어를 위한 데이터를 입력하고 관리하기 위한 도구를 구현하여야 한다. 본 연구에서는 사전에 이러한 도구를 구현하여 sec.ac.rbac 패키지와 함께 번들(bundle)로 제공함으로써 자바 애플리케이션 개발자들이 바로 프로젝트에 사용할 수 있도록 하였다. [그림 6]은 지원 도구의 실행 화면의 일부를 보여준다. 지원 도구는 기본적으로 사용자, 역할, 접근대상을 포함한 RBAC의 구성요소들에 대해 데이터를 입력, 수정, 삭제할 수 있는 기능을 제공한다. 이러한 기능들은 sec.ac.rbac 패키지를 이용하여 구현하였다. [그림 6]은 접근대상(object)를 관리하는 화면에서 사용자가 신규로 접근 대상을 등록했을 때 이를 처리하는 부분의 일부이다. [그림 7]의 코드는 sec.ac.rbac 패키지에 포함된 RbacObject 클래스의 isExistObject() 메소드와 addObject() 메소드를 사용하는 예를 보여준다.

```
if(!RbacObject.isExistObject(object_path + "\\" + object_name)) {
   RbacObject.addObject(object_path + "\\" + object_name);
   reloadData();
}
```

그림 7. 지원 도구에서 접근대상(object)을 등록하는 부분

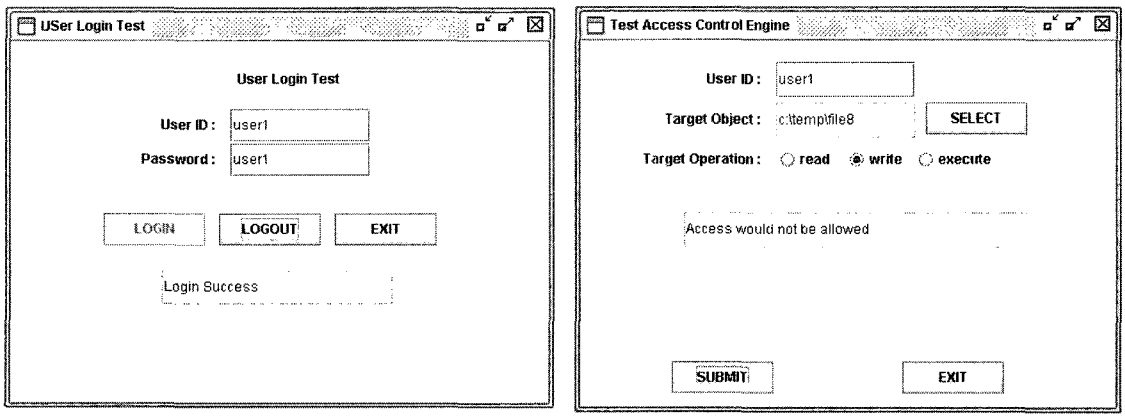
본 연구에서 구현한 지원 도구는 접근제어가 제대로 동작하는지를 테스트해 볼 수 있는 기능도 제공한다. 테스트를 위해서는 먼저 등록된 사용자중의 하나로 로그인을 하고 특정 접근대상에 대해 권한이 있는지를 테스트 해 볼 수 있다. [그림 8]의 (a)는 사용자 로그인 화면이고 (b)는 로그인한 사용자 'user1'이 'c:\tmp\file8'에 대해 쓰기(write) 권한이 있는지를 테스트하는 화면이다. 테스트의 결과 'user1'은 'c:\tmp\file8'에 대해 쓰기 권한이 없는 것으로 판별 되었다.

자바 애플리케이션 개발자는 본 연구에서 번들로 제공하는 지원 도구 대신에 자체적인 관리 도구를 구현하여 사용할 수도 있다. 이 경우에도 sec.ac.rbac 패키지가 사용된다.

3.5 개발에 대한 검토

다음은 본 연구에서 개발한 내용에 대한 기타 검토 사항이다.

- sec.ac.rbac 패키지에 포함된 모든 메소드들은 정적 메소드(static method)의 형태로 구현하여 개발자가 클래스 객체의 생성 없이 바로 메소드를 사용할 수 있도록 하였다.
- 웹 애플리케이션 환경에서 본 패키지를 이용하고자 하는 경우는 자바의 HTTP 서블릿(servlet)에 RbacActivation, RbacCatalog, RbacEngine를 포함시킴으로써 HTTP에 의한 접근 요구시 접근제어를 수행할 수 있다.
- RbacActivation, RbacCatalog, RbacEngine 클래스는 외부에서 생성할 수 없도록 하고 인스턴스만을 얻을 수 있도록 하였다. 또한 접근제어 모듈 실행시 단 하나의 인스턴스(instance)만이 존재한다.
- 본 연구에서 개발한 패키지는 의무분리(separation of duty), 위임(delegation)과 같은 정책을 지원하지 못한다. 이와 같은 정책들은 적용 환경에 따



(a) 로그인 화면 (b) 접근제어 테스트 화면

그림 8. 접근제어에 대한 테스트 기능

라 다양하게 변형될 수 있기 때문이다. 다만 추상 클래스 (abstract class)의 형태로 기본 골격만 제공하고 나머지는 개발자들이 구현(implement)하여 사용하도록 하는 방법을 고려하고 있다.

- 본 연구에서 제공하는 관리 도구는 단일 보안관리자 환경을 기본으로 하였다. Administrative RBAC 모델과 같은 다수의 보안 관리자에 의한 분산 보안 관리 환경을 지원하기 위해서는 보다 많은 연구가 필요하다.

본 논문은 새로운 접근제어 모델 (개념적 모델)을 제시하는 논문은 아니며, 알려진 개념적 모델을 실제 프로젝트 환경에서 쉽게 이용할 수 있도록 실용성을 높이는데 초점을 두었다. 그동안 많은 접근제어 모델들이 제시되었지만 현실의 문제를 직접 해결하는데 도움을 줄 수 있도록 구현된 모델은 많지 않으며, 본 논문에서와 같이 자바 환경이라면 어떤 프로젝트에서나 사용할 수 있도록 일반성을 가지는 개발결과는 찾아보기 어려웠다. 따라서 본 논문에서 제시된 RBAC 접근제어 패키지는 실제 개발 프로젝트에 많은 도움을 줄 것으로 판단된다.

4. 결 론

다수의 사용자가 많은 수의 정보객체에 접근하는 환경에서는 사용자의 권한에 따른 접근제어가 필수적이다. 본 연구에서는 개발자들이 자바 애플리케이션을 구축할 때 사용할 수 있는 자바 패키지 형태의 접근제어 모듈을 설계하고 구현하였다. RBAC 모델을 접근제어에 대한 기본 모델로 하였고, 시스템 관리자 또는 보안 관리자가 접근제어 업무를 용이하게 할 수 있도록 도와주는 관리도구를 포함하고 있다. 관리도구는 연구 기간동안 기 개발된 접근제어 패키지가 정상적으로 실행되는지를 검증하는 용도로도 사용하였다.

자바 애플리케이션 사용자들은 본 연구에서 제공하는 접근제어 패키지 및 관리도구를 사용함으로써 접근제어 모듈을 구현하는데 필요로 하는 시간을 절약할 수 있고, 여러 프로젝트에서 반복적으로 사용할 수 있다. 또한 제안한 패키지는 자바의 장점인 확장성을 가지고 있어서 웹 애플리케이션을 포함한 다양한 자바 환경에서 그대로 활용 가능하다. 현재는 기 개발된 패키지를 기반으로 다수의 접근제어 모델을 지원할 수 있는 패키지를 구현중에 있다.

참고문헌

[1] Charles P. Pfleeger and Shari L. Pfleeger, Security in Compting, Prentice Hall, 3rd edition, 2003.

[2] Matt Bishop, Computer Security, Addison Wesley, 2003.

[3] G. Saunders, M. Hitchens, V. Varadharajan, "Role-based access control and the access control matrix", ACM SIGOPS Operating Systems Review, Vol. 35 Issue 4, 2001.

[4] Ravi Sandhu, Venkata Bhamidipati and Qamar Munawer, "The ARBAC97 Model for Role-Based Administration of Roles", ACM Transactions on Information and System Security, 1999.

[5] 이완석,김홍근, "자바 보안 모델", 정보과학회지 제16권 제4호, 1998.

[6] Java Secure Socket Extension (JSSE), http://java.sun.com/products/jsse/

[7] Java Cryptography Extension (JCE), http://java.sun.com/products/jce/

[8] Java Authentication and Authorization Service (JAAS), http://java.sun.com/products /jaas/

[9] Java Secure Socket Extension (JSSE), http://java.sun.com/products/jsse/

[10] 유양우, 문남두, 이명준, "SecureJMoblet : Jini2.0 기반의 안전한 이동에이전트 시스템", 한국정보처리학회 논문지, Vol.11, No.6, 2004

[11] 김수형,장철수,노명찬,김성훈,김중배, "응용서버를 위한 보안 프레임워크 설계 및 구현", 정보처리학회 2003 추계학술대회 논문집, Vol.10 No.2, 2003

[12] S.Gritzalis, G.Aggelis, "Security issues surrounding programming languages for mobile code: JAVA vs. Safe-Tcl", ACM SIGOPS Operating Systems Review, Vol.32 Issue 2, 1998.

[13] D.S.Wallach, D.Balfanz, D.Dean, E.W.Felten, "Extensible security architectures for Java", ACM SIGOPS Operating Systems Review, Vol.31 Issue 5, 1997.

[14] M.Hauswirth, C.Kerer, R.Kurmanowytsch, "A secure execution framework for Java", Proc. of the 7th ACM conference on Computer and communications security, 2000.

[15] jGuard Project, http://sourceforge.net/projects/jguard

[16] L.Giuri, "Role-based access control in Java", Proc. of the third ACM workshop on Role-based access control, 1999.

[17] L.Giuri, "Role-based access control on the Web using Java", Proc. of the third ACM workshop on Role-based access control, 1998.

[18] S.I.Gavrila, John F. Barkley, "Formal specification for role based access control user/role and role/role relationship management", Proc. of the third ACM workshop on Role-based access control, 1998.

오 세 종(Se-Jong Oh) **[정회원]**

- 1989년 2월 : 서강대학교 컴퓨터학과 (공학사)
- 1991년 2월 : 서강대학교 컴퓨터학과 (공학석사)
- 2001년 8월 : 서강대학교 컴퓨터학과 (공학박사)
- 2003년 9월 ~ 현재 : 단국대학교 컴퓨터과학 전공 조교수

<관심분야>
정보시스템 보안, 임베디드 시스템, 유비쿼터스 컴퓨팅