韓國 OA學會 論文誌
第7卷 第2號, 2002. 6.

2002-7-2-1-2

SECOS의 접근제어를 위한 RBAC의 구현

박 동 규*, 황 유 동*, 안 현 수*

Implementation of RBAC for Access Control of SECOS(SoonChunHyang E-Commerce System)

Dong-Gue Park*, Yu-Dong Hwang*, Hyun-Su An*

요 약

SECOS(SoonChunHyang E-Commerce System)는 순천향대학교 전자상거래 소프트웨어 연구센터에서 개발된 전자상거래 시스템으로, 지불시스템, 상품검색시스템 및 이들을 통합하기 위한 프레임워크로 구성되어 있다. SECOS 내의 모든 내부 모듈들은 CBSE(Component Based Software Engineering) 기법에 기초하여 컴포넌트화 되어 있다.

본 논문에서는 SECOS의 효율적인 접근제어를 위해 RBAC를 적용하고, 분산된 검색 시스템에서의 RBAC을 효율적으로 구현하기 위하여 ACs를 적용하였으며, 또한, ACs을 제공하기 위해 Attribute Authorities(AAs)를 구현하였다. 본 논문에서 제안된 접근제어 시스템은 자바를 기반으로 한 EJB 컴포넌트로 구현하였다.

Abstract

SECOS(SoonChunHyang E-Commerce System) is the e-commerce system which was developed by e-commerce software research center in soonchunhyang univ. The system was composed of payment system, retrieving system and framework being used to combine these systems. The modules in the system was composed of components which was developed by CBSE(Component Based Software Engineering) method.

In this paper, we implement the Role Based Access Control(RBAC) component for access control of SECOS. We use Attribute Certificates(ACs) in order to implement RBAC in the distributed retrieving system, and implement Attribute Authorities(AAs) which can provide ACs. The proposed system is implemented by EJB component based JAVA.

* 순천향대학교 정보기술공학부
본 연구는 정보통신부의 ITRC 사업에 의해 수행된 것임

논문접수 : 2002. 4. 13
심사완료 : 2002. 6. 4

Ⅰ. 서론

SECOS(SoonChunHyang E-Commerce System)는, CBD(Component Based Development) 방식에 따라 개발된 컴포넌트들을 이용하여 새로운 응용 시스템을 개발하는 CBSD(Component Based Software Development)기법을 이용하여 구축된, 웹 기반 정보활용의 전형적 모델을 제공키 위한 개방형 프레임워크에 기반을 둔 시스템이다. SECOS의 전체적인 기능을 분류해 보면 첫째, 객체 웹상의 새로운 지불시스템(IC 카드형, 네트워크형)을 구현하였고 둘째, EJB 기반의 지능형 상품정보수집 및 사용자 기반의 새로운 상품검색 시스템을 구축하였으며, 셋째 상품검색을 위한 컴포넌트들을 개발하여 객체 웹기반의 지불시스템과 상품검색 시스템통합을 위한 컴포넌트 기반의 프레임워크를 개발하였다. 또한, 비즈니스 모델의 변화에 유연하게 대처할 수 있도록 CBSE(Component Based Software Engineering) 기법에 기초하여 연구가 진행되었다. 다음 그림 1은 SECOS 시스템의 전체적인 구성을 보여준다.

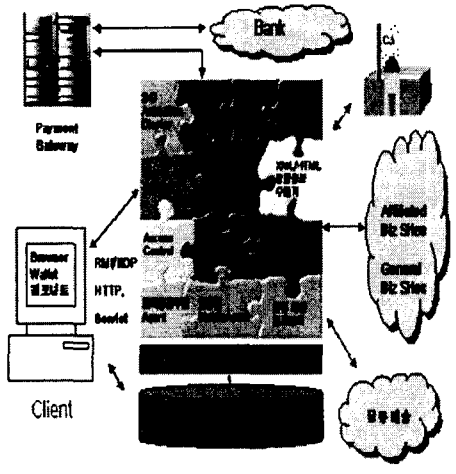

그림 2. SECOS의 구성도

본 논문에서는 분산되어 있는 검색 시스템의 접근제어 수단인 RBAC를 효율적으로 구현하기 위하여 ACs의 내용을 검토하고, ACs을 제공하는 Attribute Authorities(AAs)의 구현을 다룬다. 웹 서버의 접근 제어에 대한 현재의 연구는 주로 사용자에 기반 한다. 웹과 RBAC 기술의 성공적인 결합은 대규모 시스템에서의 효과적이고 광범위한 보안을 지원할 수 있다.

다음 그림 2는 웹상의 RBAC의 개요를 보여준다. 역할 서버는 도메인을 위해 user-role 할당 정보를 가지고 있다. 성공적인 사용자 인증 후에, 그 사용자는 역할 서버로부터 도메인에서의 사용자에게 할당된 역할을 받는다. 그 후에, 사용자가 그 도메인에서 할당받은 역할로 웹 서버로의 접근을 요구할 때, 웹 서버는 사용자의 ID(identity) 대신에 사용자의 역할에 따라서 서버에서 처리를 실행하는 것을 허락한다. 웹 서버는 서버 자신의 정책에 기반 하는 역할 계층이나 제한을 가지고 있을 수 있다. 역할 서버의 관리는 웹 관리자에 의해 분산된 형태로 수행될 수 있다. 그러나 웹 서버는 사용자에 의해 보여진 역할 정보를 보호해야 한다. 예를 들면, 악의가 있는 사용자는 위조된 역할 정보를 사용함으로써 웹 서버로의 허가받지 않은 접근을 얻을 수 있기 때문이다.

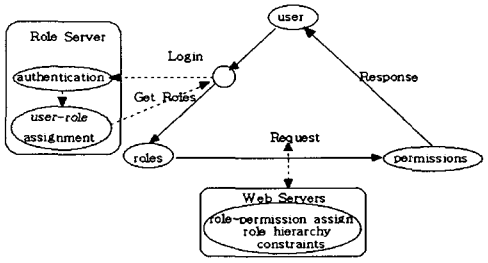

그림 2. 웹 상에서의 RBAC 개요

본 논문에서는 SECOS에서 효율적인 접근제어 시스템을 구현하기 위하여 AC를 적용한 RBAC 시스템을 이용한 효율적인 액세스제어방법을 제안하고 이를 구현하여 제안된 방법의 유효성을 검증한다.

Ⅱ. Attribute Certification

Attribute certification는 최근 생겨난 기술영역으로 신분확인의 편의를 제공하기 위하여 공개키 하부구조(PKI)에 관련한 인증을 확장하고 있다. 그것은 넓은 범위의 신분확인 결정 표준이 통합된 양식에서 관리되도록 한다. 특히 Attribute certification은 분산되고 상호 신뢰할 수 없는 환경에서 중계자의 불필요한 신임을 최소화하도록 역할 관련된 속성을 유용하고 효과적으로 관리하고 위임할 수 있게 한다. 이와 같이, Attribute certification의 정의와 채택은 역할기반 정책의 지원을 증가시키기 위한 기회를 제공한다.

다음 그림 3은 Attribute Certificate가 웹 상에서 어떻게 RBAC를 위해 사용되는지를 보여준다. 사용자가 RBAC이 사용된 도메인에서 웹 서버에서의 작업을 실행하는 것을 원한다면, 사용자는 우선 세션의 시작 시점에서 역할 서버에 연결한다. 역할 서버가 사용자를 인증한 후에, URA (User-Role Assignment) 데이터베이스에서 기존에 할당된 사용자의 역할을 발견하고 Attribute

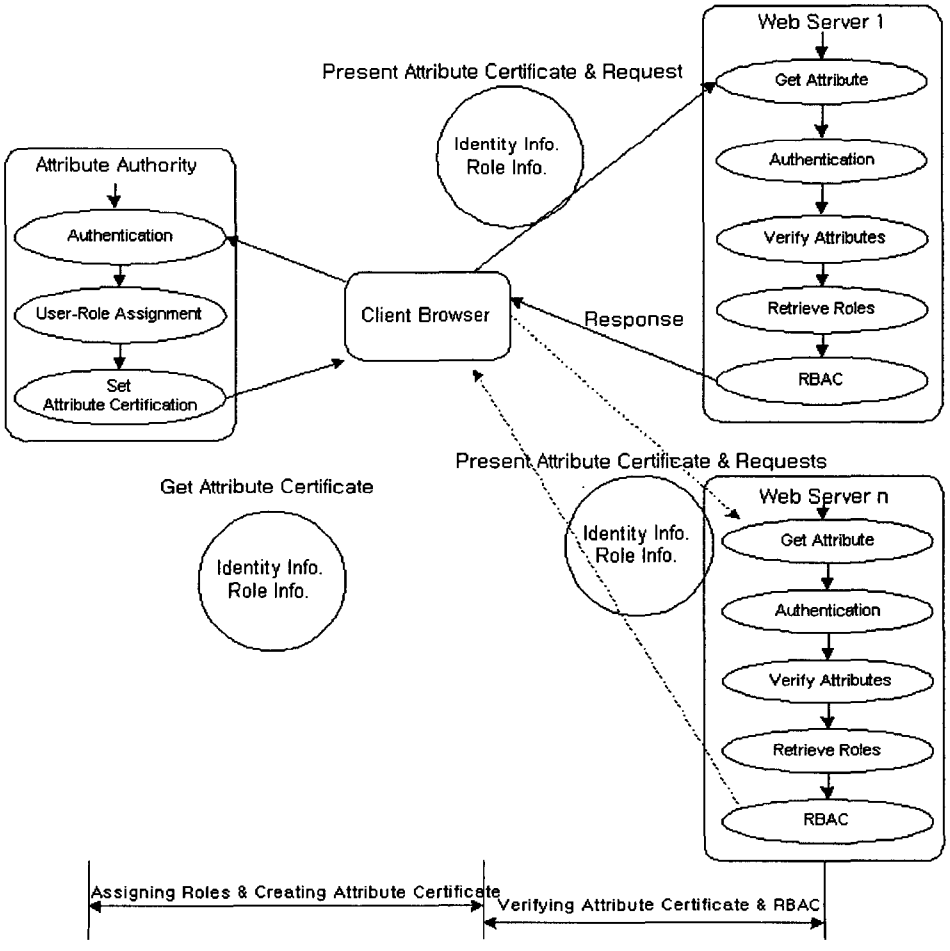

그림 3. Attribute Certificate를 사용한 분산 검색 시스템에서의 RBAC

Certificate를 만든다.

그리고 나서 생성된 Attribute Certificate는 Attribute Certificate와 일치하는 사용자의 개인키를 가지고 있는 사용자의 시스템으로 보내져서 저장되고 그 결과, 사용자는 Attribute Certificate의 수명이 다 끝날 때까지 자신의 할당된 역할을 얻기 위해 역할 서버로 되돌아 갈 필요가 없다. 결과적으로, 그 증명서가 유효한 한 사용자는 RBAC이 사용된 도메인에서 자신의 Attribute Certificate내에 표시된 역할을 사용할 수 있다.

사용자는 자신의 시스템에서 많은 Attribute Certificate를 가지고 있을 수 있다. 사용자가 클라이언트의 증명서를 요청하고 PRA (Permission-Role Assignment)를 갖는 웹 서버로의 접근을 하려고 할 때, 자신의 브라우저에서 서버의 URL을 입력함으로써, 브라우저와 웹 서버는 서로 SSL을 통해서 서로 인증한다. 브라우저가 서버의 X.509 증명서를 받아서, 검증한 후에 사용자는 자신의 역할 정보가 있는 고유의 Attribute Certificate를 선택하여, 웹 서버에게 그것을 보낸다. 웹 서버는 Attribute Certificate를 검증함으로써 사용자를 인증한다. Attribute Certificate가 유효하고 성공적으로 검증됐다면, 웹 서버는 그 증명서에 있는 역할 정보를 신뢰하고, RBAC를 위해 역할 계층조직과 Web 서버에 있는 permission-role 할당 정보를 사용한다.

속성 증명서는 역할, 그룹과 같은 사용자의 속성 정보를 포함한다. 그러나 속성정보가 사용되기 전에 속성 증명서의 사용자가 인증되기 때문에, 속성증명서 만으로는 사용될 수 없다. 속성 증명서는 속성 증명서가 하나 이상의 유효한 ID 증명서와 함께 연결되는 방법을 사용하여야 한다. 본 논문에서는 사용자의 공개키를 포함한 X.509 증명서의 사용자 이름과 일련 번호에 의해 X.509 증명서와 속성증명서를 연결하는 다음 그림 4와 같은 방법을 사용하였다. 이 연결은 ID 증명서와 속성 증명서의 다른 수명을 제공한다. 그림 4와 같은 속성 증명서를 발행하는 순서를 설명하면 다음과 같다.

1)사용자는 다른 ID 인증기관에 의해 디지털로 서명하고, 하나 이상의 ID 증명서를 발행 받는다. 그 증명서는 사용자의 속성 정보를 가지고 있지 않다. 후에, 속성 인증기관은 사용자를 위해 사용자의 ID 증명서의 일련 번호를 사용하여 속성증명서와 연결(이외에도 사용자의 공개키, hased 공개키, 암호화 또는 hased 암호, 응용기반과 도인 정책과 같은 다른 정보를 사용하여 연결하는 것이 가능하다)한다. 2) ID 인증기관은 사용자를 위해 ID 증명서를 발행하고, 그것을 서명한다. 3) ID 증명서 발생 뒤에, 속성 인증기관은 사용자의 속성, ID 증명서와의 연결정보, 그리고 속성의 수명과 발행자의 정보와 같은 다른 관계된 정보를 포함한 속성 증명서를 발행한다. 4) 속성 인증기관은 ID 증명서에서 일치된 연결 정보를 포함하는 속성 증명서를 서명한다.

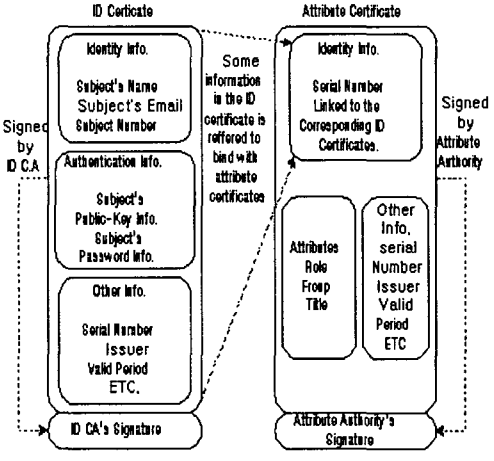

그림 5. 인증서와 속성증명서의 결합

Ⅲ. EJB를 사용한 검색 시스템에서의 RBAC 구현

1. 전체 시스템 구성

SECOS의 접근제어 시스템의 구성은 다음 그림5와 같다.

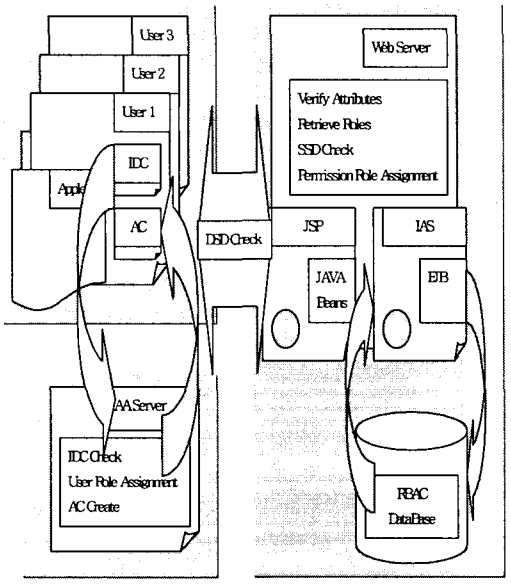

그림 5. SECOS의 접근제어 시스템 구성도

사용자가 쇼핑몰에 접근하면 제일 먼저 애플릿을 사용하여 속성증명서를 확인한다. 애플릿은 검색시스템이 있는 서버에 존재하게 되며 사용자의 접근이 이루어지면 사용자의 컴퓨터에 다운로드 되어 사용자가 가지고 있는 ID 증명서 와 속성증명서를 검사하여 검색시스템내의 EJB를 호출하게 된다.

```
-Binder Info
[ID CA's Signature]
0011162800003188005879950869456100938372
-Attributes Certificate
[Role]
E
[Group]
shoppingmall
[Title];
test Attribute Certificate
-Other Info
[Serial Number]
1142827
[Issuer]
황유동(Hwang Yu Dong)0021IB0000298522
[Valid Period]
2002-08-18 23:59:59
[ETC]
[Attribute CA's Signature]
e2ec6d2ce57d9bc09eac015379ba9a8f9a85d90b
```

그림 6. 속성증명서의 예

Applet 은 별도의 UI를 가지고 있지는 않지만 JSP page 내에 존재하면서 사용자의 인증서 정보를 확인하여 EJB를 호출하는 역할을 하게된다. 사용자의 인증서 정보가 EJB를 통하여 유효성을 확인 받게 되면 EJB는 속성증명서 내에 있는 역할 정보를 Web Server 내의 Session 정보에 이를 저장하게 된다. 그 후 사용자가 각 검색시스템의 EJB Component를 호출할 때마다 Session 에 저장된 역할 정보를 참조하여 접근허가를 결정하게 된다.

위 그림 6은 AAs로부터 사용자에 발급된 속성증명서의 예이다.

2. 사용자 접근 과정

사용자 접근과정을 정리하면 다음 그림 7과 같다.

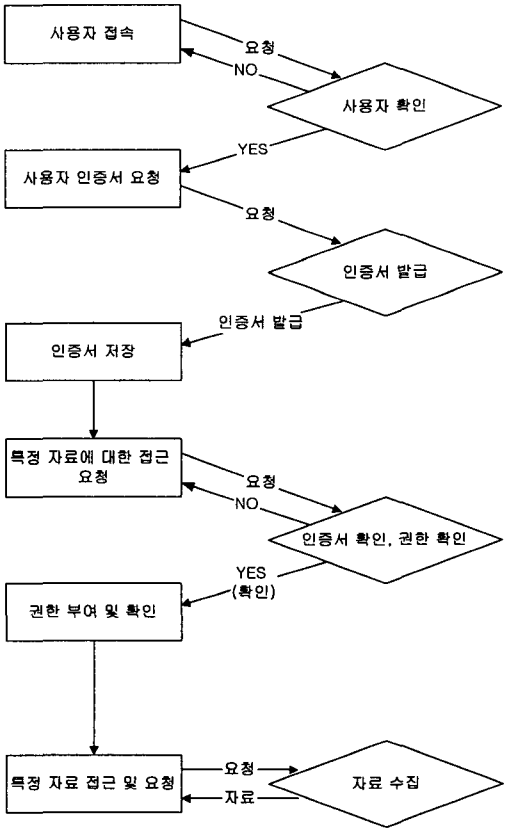

그림 7. 사용자 접근 과정

검색시스템으로 클라이언트의 사용자 접근은 검색시스템에서 클라이언트의 사용자가 인가된 사용자임을 확인한 다음 진행된다. 위 그림 7은 검색시스템 클라이언트의 접근과정을 보여주고, 접근 과정을 설명하면 다음과 같다.

① 클라이언트의 사용자가 인가된 사용자임을 인증 후 서버에서 사용자의 역할을 확인하여 사용자에 맞는 역할을 부여하게된다. ② 역할을 부여받은 사용자가 특정 자료에 대하여 접근하려 할 때 POS 서버에서는 사용자의 권한을 확인한다. ③ 사용자의 역할에 맞는 권한을 부여하고 특정 자료에 대한 접근을 허용하게 된다.

검색시스템의 데이터를 액세스하기 위하여 위와 같은 접근 과정을 거침으로써 사용자의 업무와 관련 없는 중요한 데이터를 보호할 수 있다.

각 시스템에서 사용자가 가진 역할에 대한 권한을 확인하는 과정이다. 이때 각 시스템에서는 getPermission() 컴포넌트를 이용하여 사용자의 역할 정보를 이용하여 사용자가 액세스하려는 객체에 대하여 접근 권한이 있는지를 판정하여 접근허용 또는 접근 불허를 하게 된다.

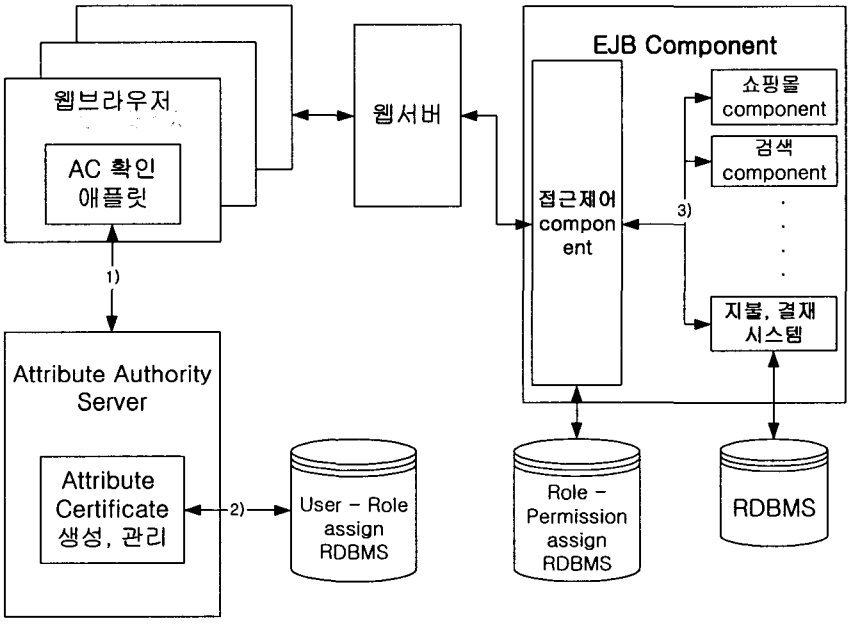

그림 9. 사용자 접근과정 인터페이스 예

3. 접근제어 인터페이스

SECOS 시스템에서의 접근제어는 위 그림 8과 같은 인터페이스에 의해 이루어진다.

1)의 과정에서 사용자가 쇼핑몰에 접근을 하여 로그인을 하면 사용자에게 AC가 있는지를 확인하여 사용자에게 발급된 AC가 없다면 AAS(Attribute Authority Serve 부터 AC를 발급 받는다. 이때 사용되는 ejb컴포넌트는 setDefault- Role()와 getUserRole()이다.

2)의 과정은 사용자의 애플릿에서 AAS에 AC를 요청하였을 때 AAS에서 데이터베이스에 접근하여 사용자에 맞는 역할 정보를 검색하는 과정이고, 3)의 과정은 사용자가 쇼핑몰, 검색 서버, 지불,결재 시스템에의 접근시

4. 역할 설정

각 역할은 쇼핑몰에서 필요한 역할들을 설정하여 정의한 것이다. 필요에 따라 관리자가 추가 또는 설정을 변경할 수 있다. 변경은 관리자 프로그램을 사용하여 관리하게 된다.

다음 그림 9는 쇼핑몰의 정보를 액세스할 수 있는 권한을 가진 역할을 나타내는 검색시스템의 역할 계층도 이고 다음 그림 10은 그림 9에서 보여준 역할의 권한과 사용자와의 매핑을 관리하는 관리자 역할 계층도이다.

다음 그림 9와 10의 역할과 권한을 설명하면 다음 표2, 3과 같다. 그림 9의 역할 중 MEMBER와 EMPLOYEE는 SSD 관계이고 SALE MANAGER와 ITEM

MANAGER는 DSD 관계이다.

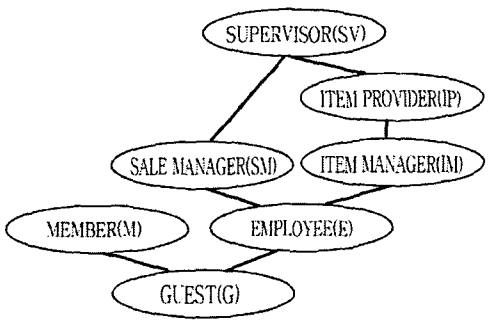

그림 9. 역할 계층도

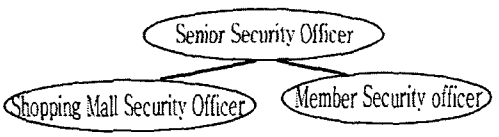

그림 10. 관리 역할 계층도

표 2. 역할 구성

역할 구분		
역할명	역할 설명	권한
Supervisor	쇼핑몰관리자	등록 및 판매 물품관리
Item_Provider	물품 등록	물품 등록 및 관리
Item_Manager	물품 관리	물품 관리
Sale_Manager	판매 관리	판매 관리
Employee	쇼핑몰 직원	물품 조회
Member	쇼핑몰 회원	물품 구매
Guest	비회원	

표 3. 관리자 역할 구성

역할명	역할 설명	권한
보안 관리자	Administrate Roles	Administrate Roles
쇼핑몰 관리자	Administrate Shopping Mall Roles	Administrate Item_Provider Role, Item_Manager role, Sale_Manager Role, Employee Role
회원 관리자	Administrate Member Roles	Administrate Member Role, Guest Role.

5. Database Table 설계

다음의 표 4부터 7까지는 역할-권한을 할당하고 취소할 수 있는 관리역할과 각 관리역할이 역할-권한 할당 데이터를 관리할 수 있도록 설계된 데이터베이스 테이블을 보여준다.

표 4 역할-권한 할당 관리역할 테이블
Table Name : Admin_Role_AssignP_Tbl

Field Name	Field Type	Length	내용
Admin_Role_ID	char	2	Admin Role ID
Pre_Condition	char	2	Prerequisite Condition
Min_Int	char	1	Role의 포함여부 구분자 (: min role 포함, (: min role 포함하지 않음
Min_Role_ID	char	5	Role ID
Max_Role_ID	char	5	Role ID
Maz_Int	char	1	Role의 포함여부 구분자 (: max role 포함, (: max role 포함하지 않음
Mobile_ImMobile	char	1	InMobile 일경우만 체크

위 표 4는 각 관리역할을 정의한 테이블로 정의된 관리역할이 역할-권한 관계를 할당할 수 있는 권한의 범위를 설정, 저장하는 테이블이다. 다음 표 5는 관리역할이 역할-권한을 할당할 때 관리역할의 전제조건을 설정, 저장하는 테이블로 표 6의 AND SET 테이블, 표 7의 NOT SET 테이블과 조인하여 전제조건의 범위로 사용한다. 관리역할의 역할-권한 할당 취소 또한 표 4, 5, 6, 7과 동일한 형태의 테이블을 만들어 설정, 저장하여 사용한다.

표 5 역할-권한 할당 시 관리역할의 전제조건 테이블
Table Name : Pre_Condition_SetP_Tbl

Field Name	Field Type	Length	내용
Pre_Condition	char	2	Prerequisite Condition
And_Set_Name	char	5	And_Set_Name
Not_Set_Name	char	5	End_Set_Name

표 6 역할-권한 할당 시 관리역할의 전제조건의 AND SET 테이블
Table Name : And_SetP_Tbl

Field Name	Field Type	Length	내용
And_Set_Name	char	5	And_Set_Name
And_Set_Role	char	5	And_Set_Role_ID

표 7 역할-권한 할당 시 관리역할의 전제조건의 NOT SET 테이블
, Table Name : Not_SetP_Tbl

Field Name	Field Type	Length	내용
Not_Set_Name	char	5	Not_Set_Name
Not_Set_Role	char	5	Not_Set_Role_ID

6. 관리자용 프로그램 구현

다음 그림 11, 12, 13은 저장 프로시저를 호출해서 사용할 수 있는 관리도구의 GUI이다.

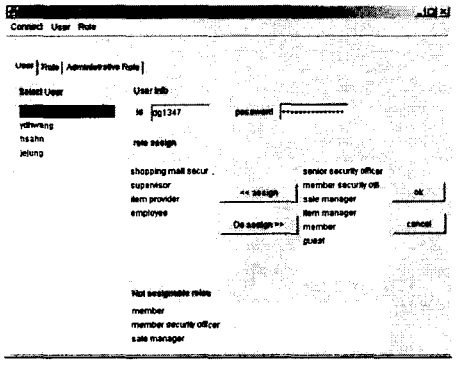

그림 11. 관리도구에서 사용자-역할 할당

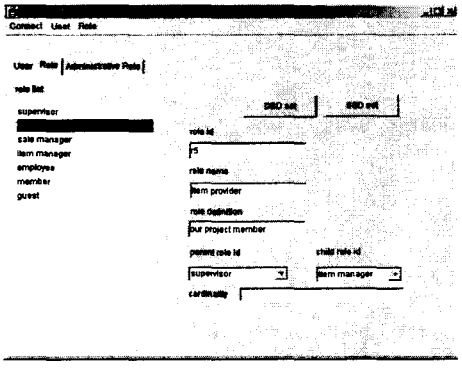

그림 12. 관리도구에서 역할 정의

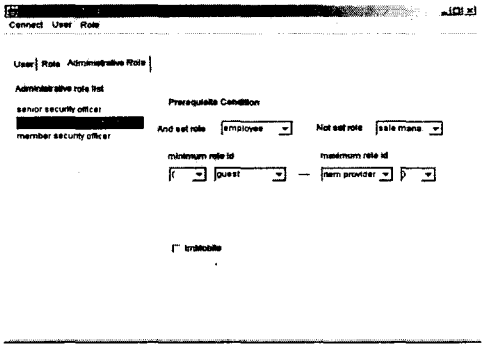

그림 13. 관리도구에서 관리 역할 정의

위 그림 11은 관리도구에서 사용자-역할을 할당하는 메뉴의 화면이고, 그림 12는 관리도구에서 역할을 정의하는 화면이다. 역할을 정의할 때 역할의 SSD와 DSD 관계 또한 같이 정의한다. 위 그림 13은 역할을 관리하는 관리역할을 정의하는 화면이다.

위 그림 14는 사용자의 역할에 대한 권한을 확인하기 위하여 사용하는 EJB 컴포넌트로서 상품 검색, 등록 등의 작업을 하는 다른 EJB 컴포넌트들이 작업을 하기 전 이 컴포넌트를 호출하여 권한이 있는지를 확인한다.

위 그림 15는 사용자에게 할당된 역할에 어떠한 권한이 있는지를 검색하는 EJB 컴포넌트이다.

6. 구현환경

표 8. 구현환경

시스템 구성 환경	
OS	Windows 2000
Database	Oracle 8.1.5
Application Server	Borland AppServer4.5
EJB 개발툴	JBuilder4
JDK	Jdk1.3
인증서관련	SDKJava4.0

EJB를 개발하고 실행하기 위해서는 여러 가지 프로그램이 필요하게 된다. 구현환경은 위 표 1과 같다.

주로 JBuilder4.0을 사용하여 개발하였고 OS는 Windows 2000 professional 버전을 사용하였다. EJB를 개발하였더라도 서비스하기 위해서는 Application Server가 필요하다. Application Server는 Borland 사

```
public void AccessControl(String role, String ejb_id) throws AccessDeniedException {
  try {
    Context ctx = new InitialContext();
    Object ref = ctx.lookupLink("RolePermissionTbl");
    RolePermissionTblHome praHome = (RolePermissionTblHome)PortableRemoteObject.narrow(ref,RolePermissionTblHome.class);
    Collection returnValue = praHome.findAll();
    if(returnValue.size()<=0) throw new AccessDeniedException("권한이 없습니다");
    Iterator iterator = returnValue.iterator();
    boolean findpra = false;
    while(iterator.hasNext()) {
      Object object = iterator.next();
      RolePermissionTbl praTbl = (RolePermissionTbl)PortableRemoteObject.narrow(object,RolePermissionTbl.class);
      if(role.equals(praTbl.getRoleId())&&(ejb_id.equals(praTbl.getObjectId()))) findpra = true;
    }
    if(!findpra) throw new AccessDeniedException("권한이 없습니다");
  }catch(Exception e) {
    throw new AccessDeniedException("권한을 확인하는 과정에서 문제가 발생했습니다.");
  }
}
```

그림 14. 역할-권한의 할당관계를 확인하는 EJB 컴포넌트의 예

```
public void findPRA(String rID) {
    jList2.removeAll();
  try {
    aPRA.removeAllElements();
    Collection returnValue = praHome.findAll();
    Iterator iterator = returnValue.iterator();
    while(iterator.hasNext()) {
      Object object = iterator.next();
      RolePermissionTbl praTbl = (RolePermissionTbl)PortableRemoteObject.narrow(object, RolePermissionTbl.class);
      if(rID.equals(praTbl.getRoleId())) aPRA.addElement(praTbl);
      else nPRA.addElement(praTbl);
    }
    int pn = aPRA.size();
    if(pn>0) {
      for(int i=0;i<pn;i++) {
        jList2.add(((RolePermissionTbl)aPRA.elementAt(i)).getObjectId());
      }
    }
  }catch(Exception e){
  }
}
```

그림 15. 역할에 해당하는 권한 정보를 검색하는 EJB 컴포넌트의 예

의 AppServer4.5를 사용하였다. 추가로, 서명된 애플릿을 만들기 위해 Microsoft 사의 SDK for Java4.0을 사용하였다. 이는 Internet Explorer에서 서명된 애플릿을 사용하기 위해서였다. OS를 제외한 프로그램들은 모두 회사의 홈페이지에서 다운받을 수 있다.

Ⅳ. 결론

본 논문에서는 분산되어 있는 검색 시스템의 접근제어

수단으로 RBAC를 효율적으로 구현하기 위해 ACs의 내용을 검토하여, ACs을 제공하는 Attribute Authorities(AAs)를 구현하였다. 연구결과로 Attribute Certificate의 내용을 검토 및 분석하였으며, 이 분석 결과를 토대로 Attribute Certificate의 검색시스템 적용을 모델링 하였고, Attribute Certificate를 발급하는 Attribute Authority의 내용을 검토 및 분석하였으며, Attribute Authority를 구현하였다. Attribute Authority의 구현으로 EJB를 사용한 검색시스템에 RBAC의 적용 및 구현, 분산 환경에서의 검색 시스템에 Attribute Certificate를 사용한 RBAC의 적용 및 구현을 하였다.

향후 계속될 연구로는 서로 다른 검색시스템에서 서로 다른 AA로부터 AC를 발급 받았을 경우 액세스 제어를 효율적으로 수행 할 수 있는 방법을 연구해야할 것으로 사료된다.

참고문헌

[1] Ravi Sandhu, Venkata Bhamidipati, "Role-Based Administration of User-Role Assignment : The URA97 Model and its Oracle Implementation", Journal of Computer Security, Volume 7,1999

[2] Ravi Sandhu , Qamar Munawer, "The ARBAC99 Model for Administration of Roles", ACSAC, 1999

[3] Ravi Sandhu , Venkata Bhamidipadi, "An Oracle Implementation of the PRA97 Model for Permission-Role Assignment ", ACM RBAC, 1998

[4] Ravi Sandhu, Joon S. Park, "Decentralized User-Role Assignment for Web-based Intranets ", ACM RBAC, 1998

[5] Dong G. Park, Yu D. Hwang "RBAC in Distributed Retrieving Systems by Attribute Certificates", IC2001, 2001

[6] J. S. Park and R. Sandhu. " Binding Identities and Attributes Using Digitally Signed Certificates" ACSAC 2000.

[7] J. S. Park and R. Sandhu. "Smart Certificates: Extending X.509 for Secure Attribute Services on the Web." In Proceedings of 22nd National Information Systems Security Conference, Crystal City, Virginia, October 1999.

[8] 박동규, 황유동, 안현수, "사용자-역할 할당을 위한 URA99 모델의 구현", 한국멀티미디어학회 춘계학술발표논문집, (2001.6.2)

[9] 박동규, 황유동 "권한-역할 할당을 위한 PRA99 모델의 구현", 한국전자공학회 하계학술발표논문집, (2001.6.28)

저자소개

박 동 규

한양대학교 대학원 전자공학과 공학박사

1992 - 1995 순천향대학교 정보통신공학과 전임강사

1995 - 1999 순천향대학교 전기전자공학부 조교수

2000년 - 현 순천향대학교 정보기술공학부 부교수

황 유 동

2000년 순천향대학교 대학원 공학석사

2001년 - 현 순천향대학교 대학원 박사과정 재학중

안 현 수

2002년 2월 - 순천향대학교 대학원 공학석사

2002년 3월 - 현 (주)휴노 테크놀로지 재직중