기업간 비즈니스 프로세스 관리에서의 접근 권한 통제

RBAC for multi-organizational Business Process Management

허원창(Hur, Wonchang)*, 배혜림(Bae, Hyerim)**

초 록

최근의 경영환경의 발전 추세는 프로세스가 복잡해지고, 다수의 기업이 서로 상호작용하는 다조직 프로세스의 필요성이 증대되는 방향으로 전개되고 있다. 그리고, 이러한 문제에 대한 시스템적 접근법으로 BPM 시스템이 각광을 받고 있다. 그러나, 아직까지 BPM 시스템은 참여 주체들 사이에 발생하는 민감한 데이터나 정보에 대한 접근권한의 통제 기능을 제공하지 못하고 있다. 기존의 RBAC (Role-based Access Control) 모델은 승인을 얻지 못한 사용자가 객체에 접근하는 것을 방지하기 위한 논리적인 틀을 제공하기 위해 데이터베이스 및 다양한 어플리케이션에 도입되어 왔다. 그러나, B2B와 SCM 등과 같은 보다 동적인 환경에서 기존의 방법론으로는 권한의 동적인 구성과 적용이 어려운 문제점을 안고 있다. 본 논문에서는 RBAC방법론에 기반한 새로운 권한 템플릿을 제시하고 이를 통해 기업간 전자거래에서의 접근 권한 통제를 효과적으로 할 수 있는 방법론을 제시한다. 또한 이러한 방법론의 프로토타입 시스템의 의사코드를 제시한다.

ABSTRACT

As the number of users who are involved in a business process increases, it becomes imperative to effectively control their privileges of accessing sensitive data and information which are usually easily obtained by BPM system. Traditional RBAC (Role-based Access Control) model was first introduced to provide a logical framework to prevent unauthorized users from obtaining confidential, but in more dynamic environment such as B2B and SCM process, it usually lacks in capability of addressing such issues as configurability, customizability, or scalability of user privileges. In this study, we have proposed a privilege-template based RBAC model that can address such issues effectively. We also provided a design of the RBAC model along with illustrative examples and pseudo codes that can be used for implementing a prototype system.

키워드: 역할기반접근제어, 비즈니스 프로세스 관리, 비즈니스 프로세스 모니터링, 물류 프로세스
RBAC, Business Process Management, Business Process Monitoring, Logistics Process

이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임.(지방연구중심대학육성사업/차세대물류IT기술연구사업단)

* 인하대학교 경영학부, 교신저자
** 부산대학교 산업공학과

1. 서 론

비즈니스 프로세스 관리(Business Process Management, 이하 BPM)는 조직의 주요한 업무 프로세스의 실행을 정보시스템을 통해 자동화하여 그 효율성을 증대하려는 경영혁신 기법의 하나이다[1]. BPM을 지원하는 시스템들은 비즈니스 프로세스의 자동화는 물론 프로세스의 진행과정에 사용자가 개입하여 그 실행을 제어하고, 진행과정을 모니터링하며, 진행 성과를 분석할 수 있는 다양한 기능을 제공한다. 기업은 이를 통해 자사 업무 프로세스의 비효율을 탐지하고 이를 개선하여 보다 효율적인 업무 절차를 재구축하기 위해 적극적으로 BPM을 도입하여 왔다.

BPM의 활용도가 높아지면서, 기업 내부의 프로세스는 물론, 기업간 거래 프로세스의 효율화를 위해서도 BPM의 기능이 폭넓게 사용되고 있다. 특히, 공급 사슬상의 거래 프로세스는 하나의 비즈니스 프로세스에 여러 관련 업체들이 참여하게 됨으로써 기업 내부의 프로세스 관리를 위한 기능과 더불어 다양한 추가적인 기능이 요구된다. 대표적인 것이 역할에 따른 접근권한의 관리이다. 특히, 참여 조직의 규모가 커지고 프로세스의 수행에 관여되는 많은 중요한 데이터들이 BPM을 통해 접근할 수 있게 됨에 따라, 정보의 보안유지와 정보의 효율적 접근이라는 두 가지 목표를 동시에 달성할 수 있는 체계적인 접근권한의 관리 방법의 도입이 중요한 요소로 여겨지게 되었다.

본 연구에서는 기존의 정보시스템의 접근권한을 통제하는 방식인 역할기반 접근통제(Role-based Access Control, 이하 RBAC) 모형에 대하여 살펴보고, 이를 기업간 거래 프로세스의 실행에 적용할 수 있는 논리 모형을 제안하였다. 제안된 모형은 RBAC의 모형화 단위(building bock)인 권한(privilege)을 데이터 객체와 그에 대한 연산 수준으로 세분화하여 정의할 수 있도록 하였고, '템플릿(template)'의 개념을 도입하여 권한 관리의 효율성과 확장성(scalability)을 도모할 수 있는 시스템적 기반을 제공하였다. 제안된 모형의 유용성을 보이기 위해, 예제 모형과 프로토타입 구현을 위한 API(Application Programming Interface) 함수들을 설계하였고, 이를 바탕으로 구현될 권한 통제 시스템의 작동 절차를 제시하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 RBAC 모형에 대한 개념설명과, 비즈니스 프로세스 관리에 적용되기 위한 요구사항을 제시하였다. 3장에서는 본 연구에서 제안하는 RBAC의 논리적 모형을 제시하였으며, 4장에서는 이를 적용한 권한 통제 절차를 제시하였다. 마지막으로 5장에서는 결론과 추후 연구과제를 제시하였다.

2. 비즈니스 프로세스 관리에서의 접근 권한 통제 (RBAC)

RBAC 모형은 '역할(Role)'이라는 개념적 개체를 통해 다양한 리소스에 대한 다중 사용자 (Multi-users)의 접근권한을 선택적으로 규정하고 이를 통해 주요 데이터의 보안

〈표 1〉 RBAC의 개체가 의미

개체	의미
권한 (Privilege)	시스템에서 수행 가능한 특정한 행위가 수행될 수 있도록 허용하는 것을 의미한다. 이러한 행위에는 기본적인 시스템의 사용, 시스템의 실행의 제어, 혹은 시스템의 감독 및 성능 관찰 등이 포함된다.
역할(Role)	시스템에서 수행이 허용된 행위의 특정한 조합을 의미한다. 즉, 하나의 역할은 다수의 특정한 권한의 집합으로 정의된다.
사용자(User)	역할에 따라 정의된 권한이 허용하는 행위를 수행하는 주체를 의미한다. 사용자는 다수의 역할을 맡을 수 있으며, 이는 일반적으로 그가 속한 조직의 구조에 의해 결정된다.
세션(Session)	사용자에게 할당된 역할이 유효한 시간적 간격을 의미한다. 일반적으로 사용자가 시스템에 접속하면, 세션이 동적으로 생성되고 세션이 유지되는 동안 규정된 역할에 따라 권한을 행사할 수 있다. 세션이 종료되면 사용자의 권한도 사라진다.

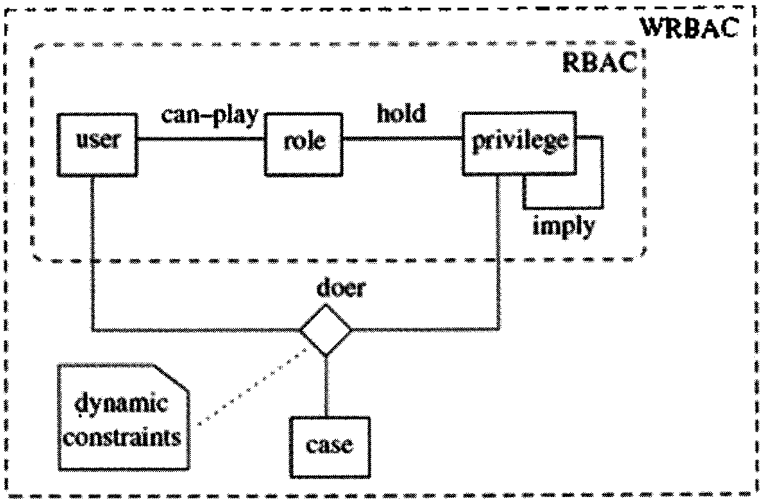
을 통제하는 가장 보편적인 보안통제(Security Administration) 모형의 하나이다[3, 4, 7]. 가장 간단한 형태의 RBAC는 (1) 사용자 (users), (2) 역할 (roles), 그리고 (3) 권한 (privileges) 의 세 가지 개체(entity)의 논리적 조합을 통해 모형화 될 수 있다. 일반적으로 하나의 역할은 복수의 권한을 수행할 수 있으며, 사용자들은 다시 복수의 역할에 할당되는 방식으로 모형화 된다. RBAC를 구성하는 각 개체의 개념은 다음의 〈표 1〉과 같이 정의된다.

RBAC 모형은 대상 응용영역에 맞게 여러 가지 형태로 변형되고 확장되어 왔다. 비즈니스 프로세스 관리에 있어서도 업무 절차의 통제 및 감시를 위해 다양한 역할과 권한을 가진 사용자나 관리자들이 관련 정보에 접근할 필요성이 생기면서, 체계적인 접근권한의 통제모형이 필요하게 되었다. 실제로 몇몇 연구에서는 비즈니스 프로세스 통제를 위한 확장된 RBAC 모형을 제안하였다 [2, 8].

〈그림 1〉은 프로세스 인스턴스를 의미하는 'case' 개체를 통해 RBAC 모형을 비즈니스 프로세스 통제에 적용할 수 있도록 확장한 모형이다. 이 모형에서는 'user', 'case', 그리고 'privilege' 세 가지 개체의 동적인 조합을 통해 업무 프로세스에 따라 사용자의 통제권한 및 정보 접근 권한이 달라질 수 있도록 하였다.

기존 연구를 통해 제시된 RBAC기반 모델은 비즈니스 프로세스에서의 체계적인 접근권한 관리의 필요성에 입각하여 출발하였고, 본 연구의 기반적 연구를 제시하였으나, 몇 가지 관점에서 여전히 추가적 연구와 모델 확장을 필요로 한다.

먼저, 대부분의 RBAC에 기반한 확장 모



〈그림 1〉 W-RBAC-RBAC for Workflow Management

형들은 사용자가 가지는 권한의 대상에 대한 구체적인 모형을 제시하지 않고 있다. 비즈니스 프로세스를 통제하거나 모니터링 하는 행위는 프로세스의 진행과정에서 생성, 삭제, 변경, 저장 되는 다양한 데이터 객체들을 대상으로 한다. 뿐만 아니라, 이에 대한 다양한 조작 및 분석을 필요로 하기도 한다. 그러나, 기존의 RBAC는 사용자가 가지는 권한이 구체적으로 어떠한 데이터 모형상의 어떠한 데이터에 대해서 어떠한 분석과 조작을 수행할 수 있는가를 체계적으로 정의하지 않고 있다.

두번째로, 기존의 RBAC는 이를 실제로 시스템에 구현하고 운용하는데 있어서 모호성이나 비효율성이 야기될 가능성이 있다. 예를 들어, 기존의 RBAC모형을 사용하여 프로세스 P_i의 수행시간의 평균치를 볼 수 있는 권한과, P_i의 한 단위업무를 수행하는 작업자 U_i의 업무완료시간의 표준편차를 볼 수 있는 권한을 구분하여 모형화 하기 위해서는, RBAC 모형과 별도로 시스템 내부에 모니터링 대상 데이터와 분석 방법이 조합되는 가짓수만큼의 세분화된 권한이 추가로 정의되어야 함을 의미한다. 따라서 권한 통제의 체계와 논리성은 RBAC 모형 자체에서 보장되지 못하고 시스템의 구현 방식에 의존적일 수 밖에 없다.

또한, 기존 접근법들은 기업내부 프로세스에서의 접근권한 모델을 다루고 있어서 사용자 수준의 역할만을 정의할 뿐, 기업간 거래에서 발생하는 프로세스에서 차지하는 개별 주체들의 역할을 정의할 수 없다는 단점을 지닌다. 이러한 한계는 접근권한의 관리가 단순히 프로세스의 실행관리에만 초점을 두도록 제한하는 부정적 효과를 가진다. 본 연구에서는 기업간 전자거래를 가정하며, 프

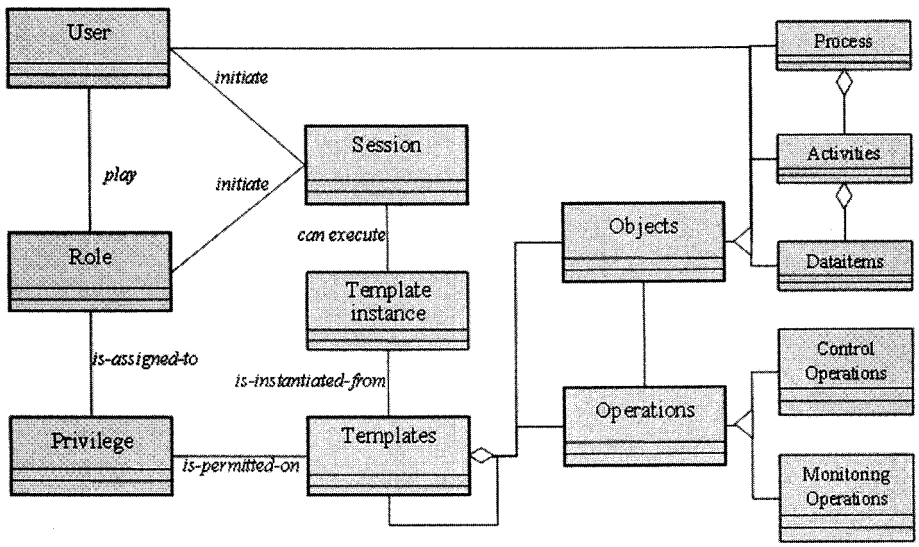
로세스 실행제어뿐 아니라, 프로세스의 모니터링 관점에서도 권한관리가 가능하도록 모델을 확장한다. 기업간 전자거래의 필요성은 관련 표준에서 이미 중요하게 인식되고 있는 문제이며, RBAC에 기반한 기업간 전자거래 프로세스의 실행 및 모니터링에 관한 접근제어는 B2Bi, SCM과 같은 영역에서 기업간 협업을 촉진하는 방법론이 될 수 있을 것으로 기대한다.

3. 템플릿(template)을 사용한 RBAC 모형

1. 개념

본 연구에서 제시하는 권한 통제 모형은 역할과 권한을 모형화 하기 위하여 연산(operation)과 연산의 대상 객체(object)까지 고려한 모형화 방식을 사용한다. 연산(operation)은 비즈니스 프로세스 관리에 필요한 사용자의 행위를 추상화 한 것이고, 객체(object)는 그 행위의 대상을 나타낸다. <그림 2>는 이와 같은 RBAC 모형을 시스템에 구현하기 위한 메타 모형을 나타낸다. 그림에서 보는 바와 같이 객체와 연산의 조합은 템플릿으로 정의되고, 권한은 이러한 템플릿에 할당된다. 사용자가 시스템의 통제행위나 모니터링 행위를 위해 세션을 생성하게 되면, 세션이 지속되는 동안 사용자의 역할에 따른 권한은 할당된 템플릿의 인스턴스를 획득하고, 이를 통해 지정된 행위를 수행하도록 통제된다.

구체적으로, 객체는 BPM 시스템의 관리



<그림 2> 프로세스 모니터링을 위한 RBAC 메타모형

대상이 되는 데이터 객체들을 의미하는 것으로, 비즈니스 프로세스(process), 단위업무(activities), 관련 문서 등을 포괄하는 데이터 아이템(dataitems), 작업자(users) 등이 포함된다[6]. 각 데이터 객체들은 그림에서 보는 바와 같은 포함관계를 가진다. 하나의 프로세스는 다수의 업무로 구성되며, 하나의 업무는 다수의 데이터 아이템을 포함한다. 대상객체에 대한 모형화는 BPM시스템에 따라 상이하나 WfMC에서 제시된 표준화를 위한 참조모형을 바탕으로 표준화된 모형을 제시할 수 있다. 대부분의 상용 BPM 시스템은 관계형 데이터베이스를 통해 위와 같은 데이터를 관리하므로 기술적으로 하나의 객체는 뷰(View)의 형태로 표현될 수 있다.

한편, 객체를 대상으로 한 연산에는 크게 프로세스의 실행을 제어하는 것과 관련되는 제어연산(control operation)과 실행 과정을 관찰하는 모니터링 연산(monitoring operation)으로 구분하였다. 제어연산은 비즈니스 프로세스의 실행과정을 통제하기 위한 행위를 의미하는 것으로 프로세스의 시작(initiate), 완료(complete), 중지(abort), 일시중지(suspend), 재개(resume), 포기(fail) 등의 연산을 포함한다. 모니터링 연산은 비즈니스 프로세스의 실행 과정과 그 결과를 관찰하는데 필요한 행위를 의미하는 것으로 비즈니스 프로세스의 실행기록, 작업자의 업무성과, 문서의 로그(log) 등을 분석하는 통계(statistical)연산, 집합(record-set) 연산, 또는 보다 복잡한 데이터 분석을 처리하는 분석적(analytical)연산 등을 포함한다[5]. 특히 템플릿을 이용하면 다양한 형태의 모니터링 연산에 개별적인 권한을 부여할 수 있다. 통제연산과 달리 모니터링 연산은 사용자에 따라 다양한 형태가 존재할 수 있는데, 예를 들어, 최대/최소값, 평균, 분산 등과 같은 간단한 기술 통계치(descriptive statistics)의 계산이나, 관계형(relational) 데이터베이스에 사용되는 필터링(filter), 그루핑(group-by), 프로젝션(project), 유니온(union) 등의 레코드 처리 연산, 그리고 보다 복잡한 분산분석(ANOVA), 회귀분석(regression), 상관분석(correlation) 등 전통적인 통계 분석 알고리즘과, PERT (Program Evaluation and Review Technique), 병목(bottleneck) 분석, 업무 부하(workload) 분석 등, 프로젝트 관리 기법에서 사용되는 알고리즘들이 모니터링 연산으로 사용될 수 있다. 본 RBAC 모형이 가지는 확장성은 이와 같은 새로운 모니터링 연산이 정의되더라도 이에 대한 권한을 새롭게 설정할 수 있도록 해 준다.

템플릿을 사용한 권한은 이와 같은 객체와 연산으로 구성된 하나의 논리적 구성(configuration)을 나타내며, 이는 다양한 RBAC 모형을 구성하기 위한 모형화의 단위이자 관리의 단위가 된다. 이러한 모형화는 조합방식에 따라 매우 세분화된 권한을 정의할 수 있게 한다. 이렇게 세분화된 권한들은 매우 동적인 환경에서 자유롭게 생성되거나 변경될 수 있다. 이를 통해 본 모형은 템플릿의 동적인 구성(configurable)과 자유로운 변경(customizable)을 가능하게 하며, 이에 따른 권한 통제모형의 확장성(scalable)을 효과적으로 실현할 수 있다.

Definition 1 역할기반접근제어 템플릿 (RBAC-T)는 [E, R, C]로 표현된다. E 는 개체(entity)의 집합, R은 관계(relations)의 집합, 그리고 C는 개체 및 관계에 적용되는 제약조건을 의미한다.

- E = {U(Users), P(Privileges), R(Roles), S(Sessions), Op(Operations), O(Objects)},
- R = {PT($\subseteq$O$\times$Op), RA($\subseteq$U$\times$R), PA($\subseteq$P$\times$R)}, where PT = privilege template, RA = role assignment, PA = privilege assignment
- C = {e | e $\in$ {consist-of(o_1, o_2), manage(u_1, u_2), own (u_1, o_2)}}, where $o_i \in$ Operations, $r_i \in$ Roles

2. 정의

전술한 RBAC 논리모형은 구체적으로 다음 'Definition 1'과 같이 표현될 수 있다.

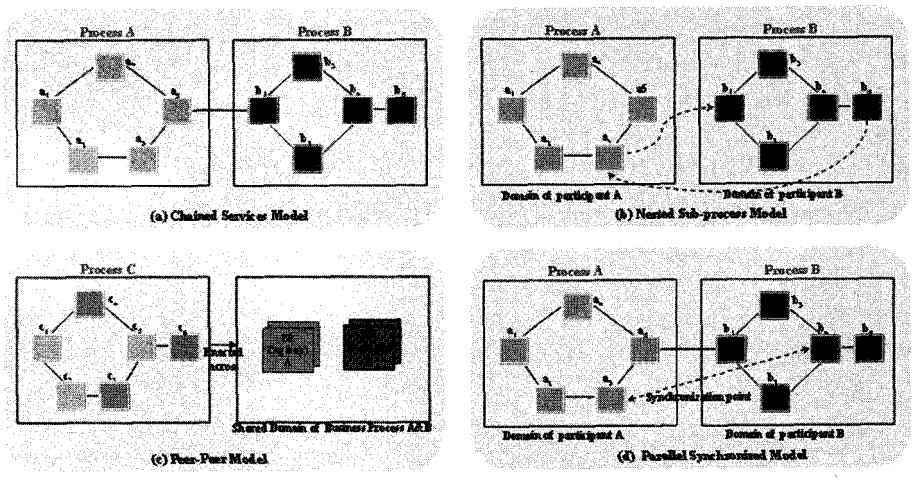
하나의 RBAC 모형은 사용자(Users), 권한(Privilege), 역할(Roles), 세션(Sessions), 연산(Operations), 그리고 객체(Objects)의 총 5개의 개체(entity)들로 구성된 집합을 기본으로 정의된다. 개체들간의 관계(relations)는 PT(Privilege Template), RA(role assignment), 그리고 PA(privilege assignment)의 세 개의 다중관계(many-to-many relationship)를 이용하여 정의된다. 여기서 주목해야 할 점은 권한이 개체로 정의되지 않고 객체와 연산의 관계로 정의된다는 점이다. 권한관계의 한 원소 $pt_i = (o_i, op_i)$은 데이터 객체 o_i에 대한 연산 op_i의 수행 권한을 나타낸다. RA는 사용자 개체와 역할 개체의 원소들 간의 할당관계를 나타내는 집합으로, 그 원소 (u_i, r_j)는 사용자 u_i가 r_j를 수행할 수 있음을 의미한다. 마찬가지로, PA는 역할 개체와 권한 개체의 원소들 간의 할당관계를 나타내는 집합으로, 그 한 원소인 (pt_i, r_j)는 권한 pt_i가 역할 r_j에 할당되어 있음을 의미한다. 제약조건은 역할의 상하관계(manage), 데이터 객체의 포함관계(consist-of)나 소유관계(owns) 등 개체간에 존재하는 종속관계를 어떻게 권한의 허용 규칙에 반영할 것인가를 결정한다. 일반적으로 조직구조에 의해 사용자들 간에는 역할의 상하관계가 존재하며, 데이터 객체간에는 데이터 구조의 특성에 의해 포함관계가 존재한다. 또한 특정한 데이터는 사용자와 소유관계를 가지기도 한다(예를 들면 중요한 문서를 작성한 하위 작업자). 이러한 제약조건은, 권한 통제과정에서 참고되어야 할 승인 규칙이나 예외사항으로 작용한다. 예를 들어, 어떠한 조직에서는 특정한 프로세스를 종료할 권한을 가진 관리자가 해당 프로세스와 포함관계를 가지는 하위프로세스를 종료시킬 수 있는 권한을 자동적으로 상속받도록 할 수 있지만, 다른 조직에서는 이를 허용하지 않을 수도 있다. 또는, 어떤 조직에서는 역할의 상속관계를 허용하지 않아서, 조직구조상 높은 권한을 가진 관리자 낮은 권한의 사용자가 참조할 수 있는 특정한 업무의 결과문서를 볼 수 없도록 할 수도 있을 것이다. 제약조건은 이러한 규칙을 표현하고, 권한 통제 시스템은 이를 통해 다른 형태의 통제 방식을 적용하게 된다.

4. SCM (Supply Chain Management) 프로세스에서 RBAC 절차

비즈니스 프로세스 관리를 통해 기대할 수 있는 보다 궁극적인 목적은, 기업의 내부 프로세스를 효율화 하는 것뿐만 아니라, 기업간의 거래 절차를 효율화하여 비용을 절감하고 운영의 민첩성을 증대하는 것이다. SCM 프로세스의 일반적인 특징은 내부 프로세스를 소유한 여러 참여업체들이 상호작용에 의해 연계된 통합 프로세스를 수행한다는 점이다. 프로세스 관점에서 이러한 상호작용의 유형을 구분하면, 다음의 〈그림 3〉에서 보는 바와 같이, 표준화기구인 WfMC (Workflow Management Coalition)에서 규정한 네 가지로 구분될 수 있다.

기존의 RBAC 모형에서 권한은 수행할 수 있는 행위와 그 대상이 자체에 내재된 개념으로써 행위의 대상에 따라 차별화 되지 않는다. 그러나 BPM을 통한 비즈니스 프로세스 관리환경, 특히 기업간 거래 프로세스를 통제하는 환경에서는, 대부분의 프로세스가 조직간, 기업간의 경계를 넘나들며 진행되며, 이에 따라 프로세스 상에서 다양한 데이터와 정보가 흘러 다닌다. 이러한 환경에서 보다 효과적인 권한 통제를 위해서는 대상 객체에 대한 체계적인 모형화와 각 데이터 객체에 대한 차별화된 기밀성의 설정, 그리고 이에 따른 차별화된 접근 권한을 규정할 수 있어야 한다.

특히, SCM상에서의 물자의 이동에 관련된 프로세스의 경우는 물자의 이송에 대한 권한을 가지고 있는 주체, 즉 예를 들면 화주가 물자의 이동에 대한 모니터링에 대한 요구사항을 가지게 되며, 향후 시스템은 이러한 모니터링을 가능하도록 해야 한다. 현재의 물류 추적 시스템은 배송을 담당하는



〈그림 3〉 Inter-process relationship

3PL등의 물류업체가 일방적으로 제공하는 기능에 의존적이며, 향후에는 화주와 물류를 담당하는 업체간의 역할 기반 권한에 대한 상호 계약에 기반한 모니터링이 필요하다. 또한, 생산업체와 부품을 제공하는 공급업체 간에도 이러한 역할 기반의 모니터링 모델이 필요하며, 본 연구와 같은 차별화된 접근 권한의 정의와 사용에 대한 필요성이 증진될 것으로 기대한다.

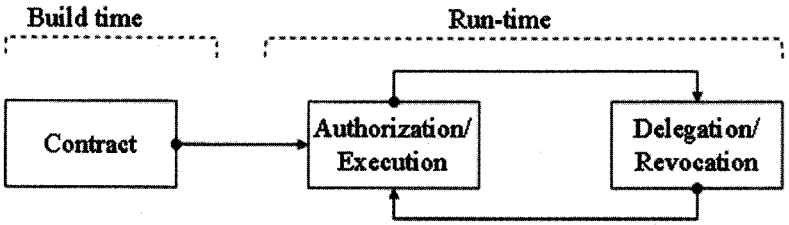
SCM 프로세스에 참여하는 각 업체가 통합 프로세스 상에서 수행할 수 있는 권한과 역할은 각 참여업체들의 프로세스 간의 관계에 따라 달라질 것이다. 따라서, SCM프로세스에 참여하는 각 업체는 별도의 계약 과정을 통하여 상호간의 역할에 대하여 명확히 정의할 필요가 있으며, RBAC 모형은 이를 효과적으로 지원할 수 있어야 한다. 다음의 〈표 2〉는 SCM 프로세스 유형에 따라 정의 가능한 역할들을 나열하였다.

최근 새로운 어플리케이션 프레임워크의 표준으로 자리잡고 있는 웹 서비스 기반의 SOA (Service Oriented Architecture) 환경에서도, 서비스를 제공하는 측과 서비스를 사용하는 측이 상호계약에 기반하여 어플리케이션이 수행된다. SCM 프로세스의 실행과정에서도 한 기업이 자신의 내부 프로세스와 관련 정보가 외부에 공개되는 것은 민감한 사안일 것이다. 따라서 미리 이루어진 프로세스의 관리 권한 구조에 대한 계약을 바탕으로 체계적인 권한의 통제 과정이 이루어져야 한다. 이를 위해서는 권한의 모형화뿐만 아니라, 권한의 제약, 권한의 승인(authorization), 권한의 이양(delegation), 권한의 회수(revocation) 등 권한에 대한 체계적인 운영 기능 역시 필요하게 된다. B2B 환경의 SCM 프로세스 상의 권한 통제는 위와 같은 모형 기능과 운영 기능을 동시에 지원할 수 있는 형태로 제공되어야 하며, 이를 위한 효율적인 기능(function)들을 갖춘 시스템이 구현되어야 한다.

RBAC에 기반한 SCM 프로세스 관리의 권한 통제 과정은 〈그림 4〉에서 보는 바와 같이 정의단계(build-time)와 실행단계(run-time)로 구분된다. 이는 BPM의 비즈니스 프로세스의 관리가 정의단계와 실행단계로 구분되는 것과 동일하다. 정의단계에서는 프로

〈표 2〉 프로세스간 관계에 따른 역할

	Company role (A)	Company role (B)
Chained Model	*Precedent.*	*Successor*
Nested Process	*Caller*	*Callee*
	Main-process	*Sub-process*
	Customer	*Service provider*
Peer-Peer	*Participant*	*Participant*
Parallel Synchronized	*Participant*	*Participant*



〈그림 4〉 RBAC 기반 프로세스 통제 및 모니터링

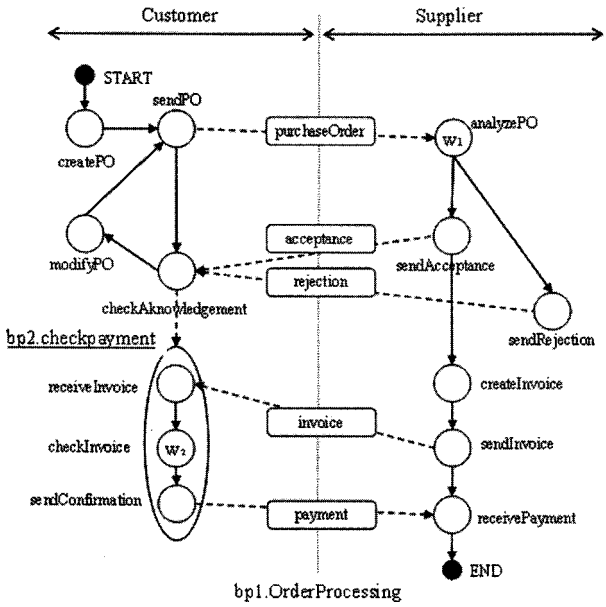
세스에 참여하는 주체들 간에 프로세스의 통제 및 모니터링에 필요한 주요 기능과 역할, 그리고 관리 대상 객체들에 대한 접근권한에 대해 합의(contract)하고 이에 대한 기술적 명세(specification)를 시스템에 구현하게 된다. 실행단계에서는 정의된 명세에 따라 권한의 통제가 이루어진다. 사용자의 특정한 연산의 수행 요구에 대한 승인(authorization) 과정과 필요에 따라 권한이 위임(delegation), 회수(revocation)되는 과정이 진행된다.

4.1 정의단계(Build-time)

정의단계에서는 프로세스의 통제와 모니터링에 관여할 사용자들(관리자나 업무수행자)을 결정하고 그들의 권한을 합의하여 시스템에 구현하는 작업이 진행된다. 이를 위해서 우선 프로세스의 진행단계에서 관리되어야 할 주요한 데이터 객체와 이에 적용되는 주요한 제어 및 모니터링 행위들을 파악하는 것이 필요하다.

〈그림 5〉는 구매자(customer)와 공급자간의 전형적인 주문처리 프로세스(bp_1)를 나타낸다. 그림에서 bp_1은 공급자와 구매자간을 연결하는 통합된 기업간 프로세스이고, 이의 한 하위 프로세스(sub process)인 bp_2는 구매자의 지불승인과 관련된 내부프로세스를 담고 있다. 특히 bp_2의 한 단위업무인 w_2는 상위 관리자의 지불승인 업무이며 그 단계에서 구매자 기업의 중요한 재무정보를 포함하고 있는 구매승인서(d_2)가 처리된다고 가정하자.

다음 〈표 3〉의 예제는 〈그림5〉의 예제 프로세스 상에 두 사용자인 구매자(u_1)과 공급자(u_2)의 역할을 정의하고 있는 RBAC 모형이다. 모형에 의하면 n 역할의 구매자 u1은 두 개의 권한 p_1과 p_4를 수행할 수 있다. 따라서 구매자 u_1은 주문처리 프로세스 bp_1을 시작시킬 수 있는 권한을 가지고 동시에 지불승인 프로세스 bp_2의 업무 w_2에서 작성된 문서인 구매승인서(d_2)의 승인을 거부할 수 있는 권한을 가진다. 한편, r_2 역할의 공급자 u_2는 권한 p_2와 p_3를 가진다. 따라서, 비즈니스 프로세스 bp_1을 구성하는 주문요청서 처리업무 w_1에서 작성된 문서 d_1의 내용을 참조할 수 있으며, 비즈니스 프로세스 bp_2의 수행성과에 대한 통계분석('stats')을 수행할 수 있다.



〈그림 5〉 전형적인 주문처리 프로세스

〈표 3〉 간단한 RBAC 모형

Entities & Assignments	Privileges
U(sers) = {u1, u2} R(oles) = {r1, r2} O(peration) = { 'initiate', 'read', 'stats', 'abort' } O(bjects) = {d1, d2, bp1, bp2, u1, u2} P(riviliges) = {p1, p2, p3, p4}, RA = {(u1, r1), (u2, r2)},PA = {(r1, p1), (r1, p4), (r2, p2), (r2, p3)}	p1 = ('initiate', bp1), p2 = ('read', bp1.w1.d1), p3 = ('stats',bp2), p4 = ('abort',bp2.w2.d2) ……

〈표 3〉의 RBCA 모형에 의하면 공급자(u_2)는 비즈니스 프로세스 bp_2의 전체적인 수행성과에 대한 모니터링(p_3)을 할 수 있지만, 구매자가 지불을 승인하는 내부 프로세스인 bp_2에 대한 세부 정보를 보거나 해당 업무를 종료시킬 수 있는 권한은 허용되지 않는다. 이와 같이 조합 가능한 권한 통제 모형을 사용하면 서로 다른 관리 체계를 가지는 부서간, 혹은 기업간의 업무 수행 환경에서 통합적으로 진행되는 비즈니스 프로세스의 실행 및 모니터링과 관련된 다양한 통제 권한을 조합하고 모형화하는 데 있어서 매우 유용하다.

4.2. 실행단계(Run-time)

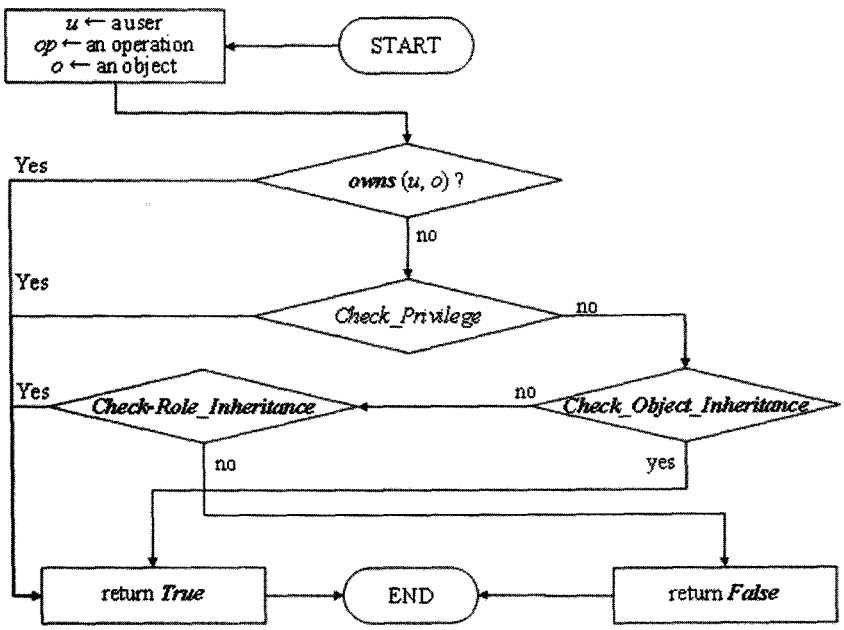
실행단계에서는 정의단계에서 설정된 권한 모형의 인스턴스를 바탕으로 프로세스 통제 및 모니터링 연산의 권한 승인 및 위임 등의 권한 통제 과정이 진행된다. 기술적으로 연산의 통제 과정은 해당 프로세스에 대한 특정한 연산을 요구한 사용자가 그 연산의 수행 권한을 가지고 있는지를 확인하는 후 해당 연산이 실행될 수 있도록 허용하는 과정이다. 이러한 과정은 BPM 시스템에 구현된 주요한 API 함수들을 통해서 수행된다. 본 연구의 권한 통제 모형에 사용되는 주요한 API 함수들은 다음의 〈표 4〉에

〈표 4〉 SCM 프로세스 모니터링을 위한 주요 API

Functions	Descriptions
is_permitted_on p, op, o)	$(P \times O_p \times O) \rightarrow$ {True, False}:해당 권한 p가 객체 o에 대한 연산 op에 대하여 정의된 권한인지를 체크
has role (u, r)	$(U \times R) \rightarrow$ {True, False}:해당 사용자가 주어진 역할에 할당되어 있는지를 체크
is_assigned_to (p, r)	$(P \times R) \rightarrow$ {True, False}:해당 권한이 주어진 역할에 포함되어 있는지를 체크
is_stronger_than (u_1, u_2)	$U^2 \rightarrow$ {True, False}:u1 사용자가 u2 사용자보다 더 높은 직위에 있거나 더 강력한 사용자인지를 체크
is_stronger_than (o_1, o_2)	$O^2 \rightarrow$ {True, False}: o1이 o2보다 상위의 객체인지를 체크.
owns (u, o)	$(U \times O) \rightarrow$ {True, False}: 해당 객체의 소유자가 u인지를 체크
has privilege (u, p, o)	$(U \times P \times O) \rightarrow$ {True, False}:해당 사용자가 o에 대한 권한 p를 가지고 있는지를 체크
users (r)	$R \rightarrow 2^U$:해당 역할을 수행할 수 있는 모든 사용자를 반환
users (p, o)	$(P \times O) \rightarrow 2^U$:o에 대한 p한을 가진 모든 사용자를 반환
roles (p)	$P \rightarrow 2^R$:p권한을 가진 모든 역할을 반환
roles (p, o)	$(P \times O) \rightarrow 2^R$:o에 대한 p권한을 가진 모든 역할들을 반환
delegate (u_G, u_D, p, o):	객체 o에 대한 권한을 가진 u_G 가 상대편 사용자 u_D 에게 o에 대한 권한 p를 부여함. 부여된 역할을 철회할 때에는 REVOKE (u_G, u_D, p, o)를 호출함
revoke (u_G, u_D, p, o)	객체 o에 대한 권한을 가진 u_G 가 상대편 사용자 u_D 에게 o에 대한 권한 p를 철회함.

나열되었다.

이 API들을 활용하여 권한 통제를 위한 핵심 절차인 권한 승인 과정을 그림 6과 같은 순서에 따라 구현하게 된다. 승인과정은 사용자가 특정 객체에 대한 연산을 수행할 수 있는 지를 확인하는 절차로써, 사용자(u)와 연산(op), 그리고 객체(o)가 입력으로 제공된다. 승인과정은 우선 사용자 u가 객체 o의 소유자인지를 확인한다. 소유자일 경우 자동적으로 연산에 대한 수행 권한이 승인된다. 소유자가 아닐 경우 승인과정은 권한을 확인하는 단계(check_privilege), 객체의 상속관계(check-object-inheritance)를 확인하는 단계, 그리고 역할의 상속관계(check-role-inheritance)를 확인하는 단계의 총 3단계를 거치게 된다. 권한의 확인에서는, 사용자 u가 가진 모든 역할과 권한을 검색하여, 연산 op와 객체 o를 포함하는 권한 템플릿이 설정되어 있는지 확인하여 권한을 승인된다. 이 단계에서 승인이 되지 않을 경우, 두 번째 단계에서는 사용자 u가 객체 o의 상위객체에 대하여 연산 op를 수행할 권한을 가지고 있는지 확인한다. 즉, 사용자가 객체 o의 상위 객체 중 해당 연산을 수행할 수 있는 권한을 가지고 있다면 객체 o에 대한 연산도 승인하게 된다. 두 번째 단계에서도 승인이 되지 않을 경우에는 마지막으로, 역할의 상속관계를 확인하는 단계에 들어간다. 이 단계에서는 사용자 u보다 하위의 역할을 가진 사용자들 중에 객체 o에 대해 연산 op를



〈그림 6〉 권한의 승인 절차

수행할 권한이 있는 사용자가 있는지를 확인한다. 그러한 사용자가 있을 경우 권한은 승인된다. 이와 같은 과정은 앞서 기술한 RBAC 모형의 제약조건에 영향을 받는다. 즉, 제약조건에서 객체의 포함관계나 역할의 상하관계에 예외조항을 설정함으로써 상위 객체에 권한을 가지더라도 특정한 하위 객체에 접근할 수 없도록 설정하거나, 상위의 역할을 가진 사용자라도 특정한 하위 역할에 규정된 권한을 수행할 수 없도록 통제할 수 있다.

다음의 〈표 5〉는 이와 같은 승인 단계를 〈표 4〉의 API를 이용하여 구현한 의사코드이다.

마지막으로, 권한 승인 절차 이외에 권한의 위임 및 회수를 처리하는 과정도 필요하다. SCM 프로세스는 전체적으로 하나의 프로세스이지만 내부 프로세스가 중첩, 연결 등의 관계로 각 참여주체가 가지는 프로세스의 조합으로 구성된다. 따라서, 계약 단계에서 공개된 내부프로세스에 대해서는 필요에 따라 다른 업체로 권한을 이양하는 단계가 필요하며, 프로세스의 종료 후에는 이를 다시 회수하는 것도 필요하게 된다. 구현단계에서 권한의 위임과 회수는 기본적으로 계약 내용에 따르며, 〈표 4〉에 정의된 위임함수(delegate)와 회수함수(revoke)를 통하여 승인과정과 같은 방식으로 구현할 수 있다.

5. 결 론

본 연구에서는 사용자의 권한에 따라 비즈니스 프로세스의 실행제어 및 모니터링 행위를 통제할 수 있는 RBAC 모형을 제시하였다. 제시된 RBAC 모형은 사용자의 권한을 행위와 대상의 조합으로 구성할 수 있도록 하였다. 이와 같은 방식은 다음의 두 가지의 장점을 가진다. 첫째, 사용자에 따라 개인화(personalized)된 다양한 권한을 정의할 수 있다. 일반적으로 대부분의 상용 BPM시스템은 시스템에 미리 구현된 표준화된 몇 개의 권한을 사용하여 사용자의 접근통제를 하도록 되어있다. 이러한 방식은 관리해야 할 비즈니스 프로세스가 많아지고, 기업간 거래와 같이 다양한 조직의 참여자가 관여되는 복잡한 비즈니스 프로세스에 대해 다양한 형태의 접근권한이 필요하게 될 경우

〈표 5〉 API를 이용한 Psuedo-code

Steps	Pseudo-code
Check_Object_ Inheritance	o_s exists such that ***is_stronger_than*** (os, o) & ***is_permitted_on***(p, op, os)?
Check_Role_ Inheritance	For any u_s (***is_stronger_than***(u, u_s)), p, r exists such that ***has_role***(u_s, r) & ***is_assigned_to***(p, r) & ***is_permitted_on***(p, op, o)?
Check_Privilege	(p, r)exists such that ***has_role***(u, r) & ***is_assigned_to***(p, r) & ***is_permitted_on***(p, op, o)?

이에 대한 요구를 만족시킬 수 없다. 또 하나의 장점은, 확장성에 있다. 권한 템플릿은 시스템을 통해서 관리할 할 수 있기 때문에 대상객체와 수행연산의 자유로운 조합을 통해 언제든지 새로운 권한을 정의하거나 수정, 갱신하는 것이 가능하다. 무엇보다도, 권한의 설정을 템플릿 단위로 함으로써, 복잡한 제약조건을 통해 규정하는 기존 방식에 비해 권한 통제 운영의 효율화를 기대할 수 있고, 다양한 접근 권한을 체계적으로 관리할 수 있을 것이다.

본 연구는 기업간 전자거래를 위한 RBAC 기반의 접근제어 모델을 제시하였다는 점에서는 의의를 찾을 수 있지만, 추후 연구과제로 먼저, 접근제어의 권한 위임을 하는 주체를 명시하고 이를 통해 모델의 적용가능성을 높이는 것을 생각할 수 있다. 전자거래에서도 권한을 위임하는 주체에 대한 다양한 시나리오를 검토하고 이를 모델에 반영하는 연구가 필요할 것이다. 다음으로, 권한을 위임하는 구체적 방법론에 있어서 세부화하여 적용할 필요가 있다. 객체들 간의 관계에 의해 권한위임이 전파되기로 하고 위임된 권한이 철회되기도 하는데, 이러한 위임과 철회의 과정이 객체들 간의 관계에 의해 정의되므로 이러한 관계에 따른 구체적인 위임/철회 방법론을 세분화하여 제시할 필요가 있겠다.

참 고 문 헌

[1] Blurton R. T., BPM-Profiting process management. SAMS, USA. 2001.

[2] Castano, S., Casati, F., and Fugini, M., "Managing workflow authorization constraints through active database technology," Information Systems Frontiers, Vol. 3, No. 3, pp. 319-338. 2001.

[3] Cheng, E. C., "An object-oriented organizational model to support dynamic role-based access control in electronic commerce," Decision Support System, Vol. 29, No. 4, pp. 357-369. 2000.

[4] Ferraiolo, D. et al., "Role-based access control (RBAC): Features motivations," Proceedings of 11th Annual Computer Security Application Conference, New Orleans, LA, pp. 241-248. 1995.

[5] Hur, W. et al., "Customizable Workflow Monitoring," Concurrent Engineering: Research and Application, Vol. 11, No. 4, pp. 313-325. 2003.

[6] Hur, W., "Design of a Distributed Enactment Model for Business Process Management," Journal of the Korean Institute of Industrial Engineers, Vol. 32, No. 3, pp. 191-199. 2006.

[7] Sandhu, R. et al., "Role-based access control models," IEEE computer, Vol. 29, No. 2, pp. 38-47. 1996.

[8] Wainer, J. et al., "WRBAC - a workflow security model incorporating controlled overriding of constraints," International Journal of Cooperative Systems, Vol. 12, No. 4, pp. 455-486. 2003.

[9] Wainer, J. et al., "DW-RBAC: A formal security model of delegation and revocation in workflow systems," Information Systems, Vol. 32, No. 3, pp. 365-384, 2007.

저 자 소 개

허원창	(E-mail: wchur@inha.ac.kr)
1997.2.	서울대학교 산업공학과 (학사)
1999.2.	서울대학교 산업공학과 (석사)
2004.2.	서울대학교 산업공학과 (박사)
2005.9.	(주)CyberMed 연구소장 역임
2005.9 ~ 현재	인하대학교 경영학부(MIS) 전임강사
관심분야	BPM, Intelligent system, knowledge management

배혜림	(E-mail: hrbae@pusan.ac.kr)
1996.2	서울대학교 산업공학과 (학사)
1998.2	서울대학교 산업공학과 (석사)
2002.8.	서울대학교 산업공학과 (박사)
2002 ~ 2003.	삼성카드, 정보기획팀
2003 ~ 2004.	동의대학교 인터넷비즈니스학과 전임강사
2004.9 ~ 현재	부산대학교 산업공학과 조교수
관심분야	BPM, eAI, 웹서비스, 물류 프로세스 관리