情報保護學會論文誌
第12券 第4號, 2002. 8

애드호크러시 조직의 특성을 고려한 역할기반 모델*

심 완 보**, 박 석***

A RBAC Model Considering the Characteristics of the Adhocracy Organization

Won Bo Shim**, Seog Park***

요 약

접근제어는 사용자가 자원에 접근시 해당자원에 대한 접근권한이 있는지를 검토해 접근을 허가하거나 거부하는 것을 말한다. 대표적인 접근제어 방법으로는 임의적 접근제어, 강제적 접근제어, 역할기반 접근제어가 있으며 현재는 역할기반 접근제어 방법이 좋은 평가를 받고 있다. 그러나 지금까지의 역할기반 접근제어 모델은 업무가 표준화 되어 있고 변화가 없는 안정적인 구조의 상하관계가 명백한 관료제의 조직 구조들을 지원하는 모델들이었다. 일부 Team Based Access Control Model 과 같은 팀 개념을 지원하는 접근제어 모델이 제안되긴 하였지만 기업의 태스크포스팀과 같은 유기적이며 임시적이고 업무가 표준화되어 있지 않고 환경변화가 많으며 상하관계가 분명치 않은 애드호크러시 조직의 특성을 충분히 반영하지는 못했다.

본 논문에서는 기존의 관료제 조직과 다른 애드호크러시 조직의 특성을 살펴보고 이러한 특성으로 인해 기존의 접근제어 모델들이 애드호크러시 조직의 접근제어 모델로서 사용시 발생되는 문제들을 살펴보았고 이러한 문제들을 해결하기 위한 개선된 역할기반 접근제어 모델을 제안하고 주요 평가기준에 따라서 분석하였다.

ABSTRACT

Role Based Access Control (RBAC), which is a method, using role as an access control, has been popular with users and it is recognized as an effective method to replace the Discretionary Access Control and the Mandatory Access Control. However, the existing Role Based Access Control Models have only been limited to the bureaucracy organization in which a distinctive hierarchy system was used, incorporating a stable structure and a standardized work system.

Only in some parts, some access control models have been used, which supports 'Team' concept, such as Team Based Access Control Model. However, it did not incorporate the characteristics of the adhocracy organization, which is similar to the company's task force team, whose characteristics are organic, temporary, no standardized operation procedures, and many frequent changes.In this study, we have discussed the characteristics of the adhocracy organization, which is different from the existing bureaucracy organization, and we have also discussed the problems related to when the existing access control models are used as the access control model for the adhocracy organization due to its characteristics.In addition, based on the problems, we have suggested an improved role based access control model for the adhocracy organization, and have come up with the solutions when any problems occur in the access control system.

Keyword : *Access Control, RBAC(Role Based Access Control), Adhocracy Organization*

---

* 본 논문은 2001년도 충청대학 교내연구비 지원으로 수행되었음.
** 서강대학교 컴퓨터학과 박사과정(cool96@chch.ac.kr)
*** 서강대학교 컴퓨터학과 교수(spark@dblab.sogang.ac.kr)

Ⅰ. 서 론

1.1 연구의 배경

오늘날의 조직내의 컴퓨터환경에서 구성원들이 조직내의 자원들을 접근하기 위해서는 적절한 접근정책의 적용을 받아야 한다. 주요 접근제어 방법으로 강제적 접근제어, 임의적 접근제어, 역할기반 접근제어 등의 접근제어 모델들이 제시되었다.

이러한 접근제어의 목적은 조직의 정보시스템내의 자원에 대한 보안과 안전한 공유에 있는데 대상이 되는 조직의 특성에 따라 서로 다른 사항을 고려하여야 한다. 여기서의 조직이란 일정한 환경 하에서 특정의 목표를 추구하며 이를 위해 일정한 구조를 형성하는 사회적 단위라고 정의 되며, 크게 두 가지로 분류할 수 있다.

하나는 M. Weber가 정의한 관료제(Bureaucracy) 조직이고 다른 하나는 W.G. Bennis가 정의한 애드호크러시(Adhocracy) 조직이다. 관료제 조직에서는 권한의 부여가 개인적인 힘이나 능력 때문에 이루어지는 것이 아니라, 조직내의 특정 직위에서 일정한 역할을 담당하기 때문에 직위에 따라 자동적으로 주어지게 된다. 관료제 조직은 명확한 권한계층과 고도의 공식화, 조직운영의 효율성 등의 특징으로 인해 산업혁명이후 우리 사회의 주류적인 조직구조로 인정받아 왔다.[1]

그러나 오늘날 사회는 급변하고 있고 복잡한 여러 문제를 해결하기 위하여 새로운 조직구조를 개발하고 있다. 그것이 바로 애드호크러시 조직이다.[2]

애드호크러시 조직은 급속히 변경할 수 있는 일시적 조직체계이며 그 구성원들은 직위나 신분에 따라 수직적으로 구분되는 것이 아니라 기술과 전문적 훈련에 따라 융통성 있게 기능적으로 구분되는 조직이고 복잡하고 예측하기 어려운 과업을 수행한다.

기존의 대부분의 접근제어 모델들은 관료제적인 조직을 대상으로 설계된 접근제어 모델이어서 이들을 애드호크러시 조직에 적용하는 데는 많은 문제점이 따르게 된다.

이에 애드호크러시 조직의 접근제어를 위한 문제점들을 분석하고 그 문제점들을 해결하는 접근제어 모델을 연구해 보고자 한다.

1.2 관련연구

1985년 미 국방성에 의해 규정된 TCSEC(Trusted Computer System Evaluation Criteria)는 강제적 접근제어(MAC : Mandatory Access Control)와 임의적 접근제어(DAC : Discretionary Access Control)에 대해 규정하고 있다.[3]

강제적 접근제어는 군사환경이나 매우 제한적인 환경에서 제한된 수의 보안 관리자들에 의해 일정한 규칙에 따라 사용자의 정보에 대한 접근을 통제한다. 반면에 임의적 접근제어 정책에서는 정보의 소유자들이 임의적으로 접근권한을 다른 사용자에게 위임할 수 있게 한다. 그러나 기업과 같은 상업적인 환경에서는 기업마다 서로 다른 보안 요구사항과 정책들을 반영해야 하기 때문에 강제적 접근제어나 임의적 접근제어만으로 이러한 요구를 만족시킬 수가 없다.[4]

R. Sandhu의 역할기반 접근제어 모델은 기업과 같은 조직의 구조를 자연스럽게 반영할 수 있는 역할구조를 지원하고 정책 중립적이기 때문에 기업마다의 서로 다른 보안 요구사항을 반영할 수 있고 권한관리의 비용을 줄여주어 상업적인 응용에서 강제적 접근제어나 임의적 접근제어를 대체할 수 있는 접근제어 방법으로 좋은 평가를 받고 있다.[5]

그러나 역할기반 접근제어 모델은 상하관계가 분명하고 업무처리가 표준화 되어있는 관료제조직을 대상으로 하기 때문에 애드호크러시 조직의 특성을 반영하기는 어려운 모델이다.

R.K. Thomas가 제안한 C-TMAC모델은 사용자로 구성된 팀에 의해 행해지는 협업활동을 위한 접근제어 방법을 제공한다. C-TMAC모델에서 사용자는 팀에 할당되고 팀의 구성원으로서 팀의 자원을 접근할 수 있게 된다. 그러나 각각의 사용자들은 자신이 사용할 수 있는 퍼미션들이 자신의 역할과 현재 팀의 활동에 따라 결정되어진다. 즉, 사용자의 권한은 사용자의 역할에 따른 권한과 팀의 권한의 합으로 이루어지고 이는 다시 팀의 문맥 정보에 따라 제한을 받음으로써 결정된다.[6]

C-TMAC모델은 팀원간의 협업을 지원하는 모델이긴 하나 팀을 임시적인 조직이라기 보다는 어느 정도 영구성을 갖는 팀 조직을 대상으로 한 모델이다. 또한 C-TMAC 모델에서는 사용자가 복합적인 권한을 부여받음으로 해서 발생할 수 있는 권한충돌에 대한 고려가 없으며 팀 자체적인 권한관리가 어려워 이로 인해 표준화되지 않은 업무를 처리해야 하는 애드호크러시 조직의 특성을 반영하긴 어렵다. 지금까지 살펴본 바와 같이 지금까지 연구된 기존의

접근제어 모델들은 애드호크러시 특성을 고려하지 못했거나 일부 정도만 반영하고 이러한 특성들을 모두 만족 시켜 애드호크러시 조직의 접근제어시 문제점을 모두 해결해 주는 모델은 없다고 할 수 있다.

이에 본 논문에서는 애드호크러시 조직의 특성을 고려한 접근제어 모델을 개발하고자 한다.

논문의 나머지는 다음과 같이 구성되어 있다.

2장에서는 예제를 통해 애드호크러시 조직을 위한 접근제어시 문제점들을 도출해 내고 각각의 문제점들이 해결될 수 있는 방법들을 제시할 것이다.

3장에서는 애드호크러시 조직의 특성을 반영한 개선된 RBAC모델을 제시하고 이 모델을 예제에 적용해 타당성을 검토해 볼 것이다. 4장에서는 제안된 모델을 다른 모델들과 비교하여 평가를 하고 5장에서 결론을 맺고자 한다

Ⅱ. 애드호크러시 조직을 위한 기존의 접근제어의 문제점 분석

2.1 예제를 통한 문제점 도출

다음은 애드호크러시 조직에서 일어날 수 있는 다음과 같은 예제 상황을 바탕으로 접근제어 관점에서 발생될 수 있는 문제점들을 도출해 보고자 한다.

[예제]

『최근 기업환경이 어려워져 회사의 경영이익 감소와 부채의 증가로 자금 압박을 받고 있고 회사의 사활마저 위협받고 있는 절대절명의 위기 상황에서 최고경영자는 이대로는 회사가 더 이상 버틸 수 없다는 상황판단 하에 전사적인 문제점을 찾아내고 이를 해결할 수 있는 방안을 찾고자 사내에 향후 6개월 이내에 회사의 구조조정을 통해 이 문제를 해결할 임무를 갖는 태스크포스팀을 만들었다.

구조조정 작업을 성공하기 위해서는 어느 한 분야의 문제를 해결한다고 되는 일은 아니고 시간적인 여유가 있는 것은 더욱 아니다.

구조조정에 관계된 해당분야의 전문 인력들이 모여 빠른 시간 안에 문제점을 파악해 내고 그에 대한 해결방안을 찾아내 구조조정을 마무리해야 한다.

이때 최고경영자는 자신의 기업에 사활이 걸린 문제이기 때문에 이 태스크포스팀의 팀장에게 팀 운영의 전권을 주고 수시로 진행되는 상황을 중간의 계선 조직 없이 팀장으로부터 직접 보고를 받는다.

팀장은 경영관리실의 이사인 Tom이 맡았고 그는 팀장을 맡으면서 먼저 팀 내에서 수행해야 할 일들을 정의해 보았다. 기업 구조조정을 위해서는 재무구조개선, 수출전략 수립, 인력조정, 기업매각, 투자조정의 업무가 진행되어야 한다는 결정을 내리고 이 업무들을 수행할 사내의 전문가들을 발탁했다.

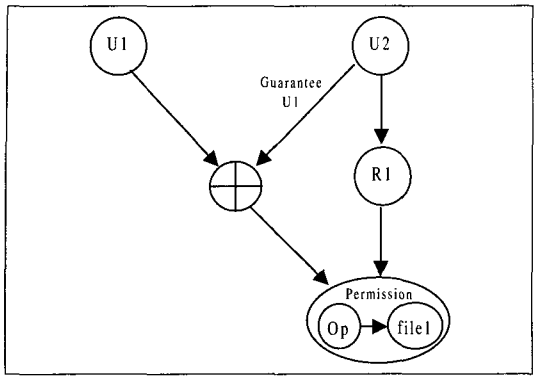
태스크포스팀에 구성되는 팀원들은 그들이 수행해야 할 업무의 중요성으로 인해 최고경영자의 재가를 받아 평소 사내 각 부서에서 업무수행 능력을 인정받아 온 전문가들로 파견을 받아 구성을 했다. 사내 경제연구소에서 근무하던 재무분야의 박사학위를 갖고 있는 재무 전문가인 선임연구원 Smith와 연구원 Ann은 이 태스크포스팀에 파견이 되었고 이들은 팀장 Tom에 의해 Smith는 재무구조개선 업무와 기업매각 업무를 동시에 맡게 되고 Ann은 재무구조개선 업무를 맡게 되었다. 경제연구소 내에서 Smith는 Manager라는 역할을 갖고 있었고 Ann은 Advisor라는 역할을 갖고 시스템에서 자원을 접근하며 업무를 보고 있었다. 팀장은 이들이 업무를 진행하는데 있어 기존의 역할 권한만으로는 그들이 맡은 업무를 수행하는데 부족함이 있어 팀 내에서 업무 수행 중 필요한 데이터에 접근할 수 있도록 내부 보안관리자(SSO: System Security Officer)인 John에게 지시를 해 시스템 상에서 Finance Director와 Finance Advisor라는 역할을 만들어 Smith와 Ann에게 각각 주었다. 이때 Finance Director 역할은 역할 구조상에서 Finance Advisor 보다는 상위의 역할이다.』

다음은 이러한 상황에서 접근제어 관점에서 일어날 수 있는 문제점들을 알아보고 각각의 문제에 대한 해결 방법을 제시하고자 한다.

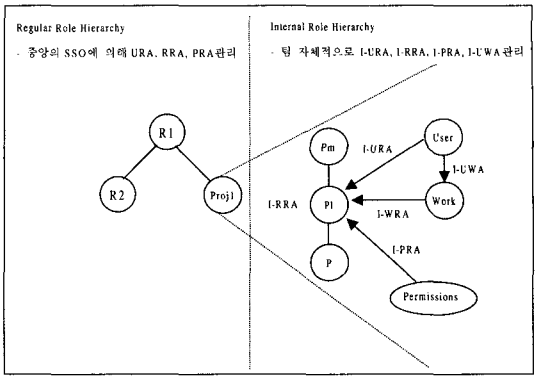
2.2 기존의 접근제어 관점에서의 문제점 및 해결

[문제 1] 수행하는 업무의 비표준화로 인한 업무 수행상 접근해야 할 자원범위 예측의 어려움이 있다.

애드호크러시 조직에서 다루어지는 업무가 일반적으로 이전에는 다루어지지 않았던 처리과정이 표준화되지 않은 일들이다. 이러한 특성으로 구성원 등에 대한 시스템 자원의 접근범위와 각 자원에 대한 접근권한을 초기부터 완벽하게 정의하는 것은 불가능하게 된다. 그러나 팀원들이 업무를 수행하는 중



〔그림 1〕 Guarantee 개념도



〔그림 2〕 이중화된 역할 모델

에는 미리 예측하지 못했던 자원에 대한 접근이 필요해질 때가 많이 있다.

예를 들면 〔그림 1〕 에서와 같이 사용자 U1이 자신의 업무 수행 중에 file1에 대한 접근이 꼭 필요하지만 현재는 file1에 대한 접근 권한이 없다고 한다면 사용자 U1은 자신의 업무를 완수할 수 없게 될 것이다. 이때 같은 팀 내의 동료인 사용자 U2가 역할 R1을 통해 file1에 대한 접근권한이 있다고 할 때 사용자 U1은 동료인 사용자 U2의 Guarantee로 file1에 대한 일시적인 접근을 허용 받을 수 있다.

이때 사용자 U1은 애초에 접근이 금지된 file1에 접근을 하였으므로 보안상의 문제가 될 수 있으나 같은 업무를 수행중인 동료 U2의 책임 하에 일시적인 접근이 허용된 것이기 때문에 이러한 접근사실에 대한 Audit정보가 관리된다면 보안상의 문제를 보완하면서 팀원들의 업무 수행을 위한 자원접근을 신속하게 해결할 수 있을 것이다.

[문제 2] 팀 내의 자치적인 권한 관리가 필요하다.

업무수행 중에 변화가 많고 업무의 중요도로 인해 자체내의 인원들에 의해서 보안이 관리될 필요가 있는 애드호크러시 조직은 중앙의 보안관리자(SSO)의 개입이나 도움 없이도 팀 내에서 자치적으로 자원에 대한 권한관리가 이루어질 필요가 있다.

우리는 〔그림 2〕 에서와 같이 역할 구조를 중앙의 SSO에 의해 관리되는 Regular Role구조와 애드호크러시 조직 내부에서 자치적으로 관리하는 Internal Role 구조로 나누어 관리되도록 함으로 해서 이러한 문제를 해결할 수 있다. 자치적인 권한관리에는 Work의 생성/삭제, 역할 생성/삭제, I-RRA에 의한 역할구조 관리, I-PRA에 의한 자원에 대한 권한조정, 업무에 따른 역할권한 View관리, I-URA에 의한 역할에 대한 사용자 할당/회수, I-UWA에 의한 Work에 대한 사용자 할당/회수, I-WRA에 의한 Work에 대한 필요한 역할 할당/회수 등이 있다.

[문제 3] 상충되는 권한충돌의 문제 해결

애드호크러시 조직의 구성원들은 이전 조직에서 가졌던 역할 권한과 애드호크러시 조직으로 파견되어 갖게 되는 역할권한 등으로 해서 복합적인 권한을 갖게 된다. 이때 같은 객체에 대해 서로 상충되는 권한이 부여 될 수가 있는데 이를 권한 충돌이라 한다.

예를 들면 어떤 사용자 U1이 객체 o에 대해 접근이 허가된 (o, +a)와 접근이 불허된 (o, -a)를 각각 다른 역할을 통해 갖게 되었을 때 사용자 U1이 객체 o 를 접근 하고자 할 때 이를 허용할 것이지 불허할 것인지 둘 중의 하나를 결정해 주어야 한다.

이 문제를 해결하기 위해 여러 연구들이 진행되어 왔다. Rabitti는 어떤 역할이 한 객체에 대해 권한을 갖게 되면 그 상위의 모든 역할은 묵시적으로 같은 권한을 갖게 되고 반대로 권한을 갖지 못하도록 하면 모든 하위의 역할은 거부된 권한을 갖지 못한다고 했다. 그러나 많은 경우에 있어 하위역할의 권한이 상위역할로 상속되지 않아야 하는 경우도 있고 마찬가지로 상위역할에서 권한을 갖지 못하게 하였다 하더라도 하위역할에서는 가질 필요가 있는 경우가 많이 있다.[14]

Bertino는 권한을 Strong과 Weak로 나누고 Strong과 Strong은 부여가 불가능하게 하고 Strong과 Weak의 충돌 시는 Strong이 Weak를 우선하도록 했다. Weak와 Weak의 충돌이 일어났을 때는 사용자에게 직접 (o,+a)나 (o,-a)를 주어 해결하거나 어느 한쪽을 삭제하거나 Strong을 주어 해결하게 했다.

그러나 이는 SSO가 현재의 Authorization State에 권한을 추가 할 때 발생하는 충돌 문제를 해결하고자 하는 것이다. 사용자가 접근하는 순간에 접근에 대한 허용 여부를 실시간으로 결정해야 하는 상황에서는 적절한 방법이 아니다.[13]

Jajodia의 논문에서도 Conflict Resolution문제를 언급하고 있는데 여기서는 사용자가 하나의 역할구조에서 권한전파 규칙에 의거해 최종적으로 갖게되는 권한의 충돌을 다루고 있다.[12]

그는 ASL(Authorization Specification Language)을 이용해 다양한 접근제어 정책을 기술할 수 있게 하여 충돌문제를 해결하고자 했다. 예를 들면 충돌을 허용치 않거나 (s,o,-a)를 우선으로 하거나 (s,o,+a)를 우선으로 하거나 혹은 결정을 유보하여 open policy나 closed policy와 같은 Default Decision 정책으로 넘기는 것이다. 그러나 권한 충돌 문제를 해결함에 있어 위의 정책들 중 어느 한가지를 적용하여 해결하는 단순한 상황보다는 몇 가지 정책이 정해진 우선 순위에 따라 차례대로 적용되어야 결정되는 상황이 많다. 우리는 관료제와 애드호크러시 조직과 같이 서로 다른 특성을 갖는 조직에서 갖게 되는 권한의 충돌을 포함하여 권한충돌 문제를 해결하고 있다.

Shen은 공동작업환경에서 사용자가 복수의 역할을 갖음으로 해서 발생할 수 있는 잠재적인 충돌 문제를 해결하는 몇 가지 규칙을 제안했다.[15]

그는 좀 더 구체적인 역할이 우선 적용되게 하였다. 그러나 역할들이 서로 상하관계가 없는 독립적인 상황에서는 어느 역할이 더 구체적인지 판단하기 어려운 상황이 많이 있다. 다음은 우리가 제안하는 권한부여 방법과 권한충돌 해결규칙을 보이고자 한다.

우리는 권한부여에 있어서 public authorization과 private authorization을 제안하였다. 이 개념을 이용해 우리는 하위 역할의 모든 권한이 상위 역할로 상속되는 것을 Sandhu RBAC 모델의 private role hierarchy를 사용하지 않고도 가능하게 했다.[5]

하위 역할에 대한 public authorization은 상위 역할로 상속되지만 하위 역할에 대한 private authorization은 상위 역할로 상속되지 않는다.
우리는 다음과 같이 접근권한을 정의하였다.

Access Authorization: (s, o, a, at)
s : subject, ex) ProjectLeader
o : object, ex) host/dir/file1
a : access mode, ex) +write, -read
at : authorization type, ex) pub, priv

권한충돌 해결 규칙
(Step 1)
충돌된 권한들 중에 어느 하나가 애드호크러시 역할에서 나온 권한이 있으면 그 권한을 우선 적용한다.

(Step 2)
권한이 같은 애드호크러시 역할 혹은 일반 역할에서 나왔을 경우

(Step 2-1)
Subject간에 상하의 관계가 있으면 상하간에 주어질 수 있는 권한의 종류는 16가지의 조합이 가능하고 이때 부호가 같은 경우는 충돌상황이 아니므로 이를 제외하고 하위역할이 private authorization이면 상위역할로 상속이 되지 않으므로 충돌상황에서 제외시킬 수 있다. 그러면 다음과 같은 4가지의 조합이 가능하게 된다.

〔표 1〕 가능한 권한충돌 조합

상위역할	(s,o, +a,pub)	(s,o, +a,priv)	(s,o, -a,pub)	(s,o, -a,priv)
하위역할	(s,o, -a,pub)	(s,o, -a,pub)	(s,o, +a,pub)	(s,o, +a,pub)

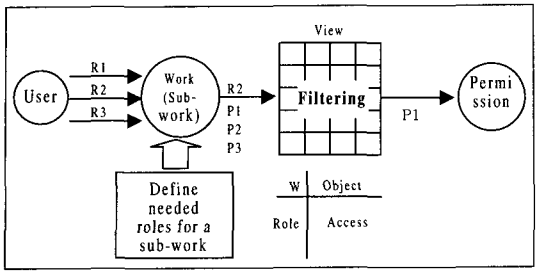
이들 충돌 가능한 조합에 대해 SSO는 어느 것에 우선권을 줄 것인지 정의할 수 있다. 이때의 권한충돌은 이때 정해진 규칙에 따라 해결한다.

(Step 2-2)
Subject간에 상하관계가 없이 독립적일 때는 명시적으로 선언된 권한에 우선권을 준다. 이는 권한부여자의 의지를 반영하기 위한 것이다.

(Step 2-3)
같은 명시적인 권한이거나 묵시적인 권한일 경우는 Negative권한에 우선권을 준다.

[문제 4] 업무(Work)에 따른 역할에 할당된 자원접근 권한의 제한

사용자가 다수의 역할을 갖고 있다고 할 때 이 사용자의 역할을 통한 권한이 업무에 관련되어서만 활성화 될 수 있도록 할 필요가 있다.



〔그림 3〕 Work개념을 이용한 권한 Filtering

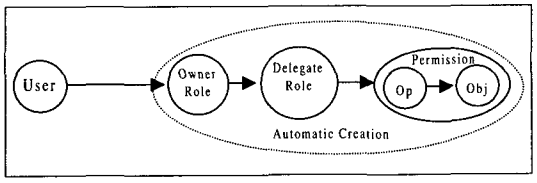
〔그림 3〕 에서 사용자가 역할 R1, R2, R3를 갖고 어떠한 Work을 수행한다고 할 때 이 Work을 수행하는데 필요한 역할은 R2뿐이라면 사용자가 비록 R1, R2, R3 역할을 갖고 있다 할지라도 이 Work을 수행 중에는 R2만이 활성화 될 수 있다.

또한 활성화된 R2도 R2에 부여된 모든 권한이 가능한 것이 아니라 주어진 Work에서 역할에 따른 객체에 대한 접근 허용 매트릭스에 정의된 접근만이 가능해 좀더 정밀한 객체접근을 하게 할 수 있다.

[문제 5] 사용자 자원에 대한 임의적 접근제어의 허용

애드호크러시 조직 내에서 사용자들은 리포트 파일과 같은 자신의 객체를 생성할 수 있다. 자신이 생성한 객체에 대한 권한은 사용자 자신이 모든 권한을 가질 수 있다. 또한 자신이 생성한 객체에 대해서는 다른 사용자가 접근할 수 있도록 자율적으로 허용할 수도 있다. 〔그림 4〕 에서 사용자가 객체 Obj를 생성하면 Obj에 관련된 Permission이 자동으로 생성되고 이를 관리하기 위한 Owner Role이 생성되어 사용자는 Owner Role에 자동으로 할당된다. 이때 다른 사용자에게 Obj에 대한 Permission을 위임해 주기 위한 Delegate Role이 Owner Role의 Sub Role로서 생성되어 최종적으로 〔그림 4〕와 같은 구조를 갖는다.

Owner Role에는 Delegate Role에 다른 사용자들을 할당하고 할당을 취소할 수 있는 Permission이 부여되어 있으며 Delegate Role에 할당된 사용자는 Obj에 부여된 Permission을 사용할 수 있게 된다.



〔그림 4〕 사용자 자원에 대한 임의적 접근제어

이때 Delegate는 보안상 1단계만 허용한다.

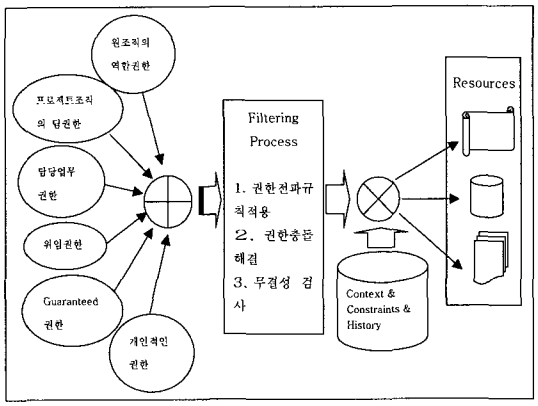
이러한 방법으로 사용자는 역할개념을 유지하며 자신의 자원에 대한 임의적 접근제어를 가능하게 할 수 있다. 여기서 우리는 사용자 자신의 자원에 대한 권한을 사용자가 보안관리자를 통하지 않고 직접 다른 사용자에게 주는 방법을 제안하는 것이며 RBAC에서는 사용자가 직접 다른 사용자를 해당 파일의 퍼미션을 갖는 역할에 할당하는 것을 허용하지 않는다. 그러므로 우리는 Delegate Role 개념을 이용해 사용자 수준에서 다른 사용자에게 자신이 만든 파일에 대한 접근을 허용하는 방법을 제안하는 것이다.

Ⅲ. 애드호크러시 조직의 특성을 반영한 개선된 RBAC 모델

3.1 애드호크러시 조직을 위한 접근제어 모델의 시스템 개요

앞에서 살펴본 바와 같이 애드호크러시 조직내의 구성원들은 다양하고 변화되는 권한을 갖고 조직내의 자원을 접근하게 되고 이에는 또한 적절한 상황에 맞는 접근 통제도 이루어 져야 한다.

〔그림 5〕는 제안하는 접근제어 모델의 전체적인 개념을 나타내는 개요도 이다. 사용자의 다양한 역할 권한들은 다음과 같은 권한 필터링 과정을 통해 자원에 대한 접근허가 여부가 결정된다. 먼저 사용자에게 부여된 역할들을 역할 구조에 따른 상속의 개념을 적용해 사용자가 사용할 수 있는 역할들의 집합이 결정되고 이에 따른 자원에 대한 다양한 접근 권한들이 결정된다. 이때 결정된 자원에 대한 다양한 권한들의 충돌은 권한충돌 해결정책을 통해 접



〔그림 5〕 애드호크러시 특성을 고려한 접근제어 모델 개요도

근허가 여부를 결정한다. 다음으로 무결성 유지를 위해 정의된 제한조건(Constraint)들에 위배되지 않는지를 검토해 최종적인 접근허가 여부가 결정된다.

3.2 Modeling을 위한 가정

(가정1)
애드호크러시 조직은 특수목적을 위한 임시적 조직이다.

(가정2)
구성원은 내부의 해당분야 전문가를 파견 받아 구성된다.

(가정3)
외부의 전문가가 구성원으로 들어올 수도 있다.

(가정4)
중앙의 보안관리자는 애드호크러시 조직에서 접근 가능한 자원의 범위를 PRA(Permission Role Assignment)에 의해 지정해 주고 애드호크러시 조직은 이후에는 지정 범위 내에서는 자치적으로 운영할 수 있다.

(가정5)
기존의 관료제(Bureaucracy)조직은 Sandhu의 RBAC 모델을 사용한다.

(가정6)
애드호크러시 조직을 만들어 이에 대한 접근제어를 하고자 할 때 제안하는 개선된 RBAC 모델을 사용한다.

(가정7)
외부의 역할관리는 중앙의 보안관리자에 의해 이루어지고 내부의 역할관리는 내부의 보안관리자 혹은 팀장에 의해 관리된다.

3.3 제안하는 모델

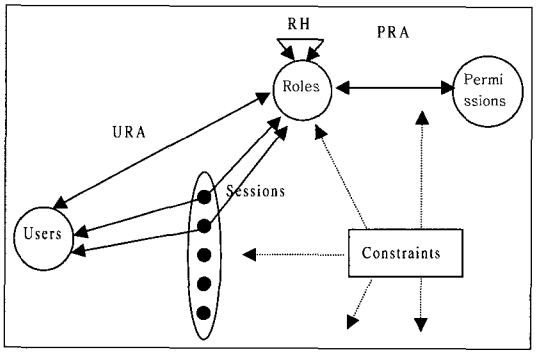
제안하는 모델에서는 기존의 관료제 조직과 애드호크러시 조직의 특성을 통합하여 반영한 접근제어 모델을 만들기 위해 이중의 역할 구조를 사용했다.

기존의 관료제 조직을 위한 접근제어 모델로는 Sandhu의 RBAC 모델을 사용하고 이를 외부역할 모델이라고 할 것이다. 애드호크러시 조직을 위한 접근제어 모델로는 제안하는 내부역할 모델을 사용할 것이다.

3.3.1 외부 역할 모델

〔그림 6〕은 Sandhu의 기본 RBAC 모델이다.

우리는 외부역할 모델로서 Sandhu의 RBAC 기본모델을 사용한다.

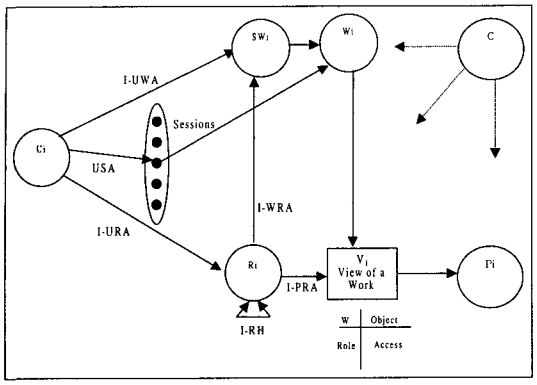


〔그림 6〕 외부역할 모델

Sandhu의 RBAC 기본모델은 다른 논문들에서 많이 다루어 졌기 때문에 여기서는 자세한 설명은 생략하기로 한다.

3.3.2 내부 역할 모델

〔그림 7〕은 애드호크러시 조직을 위한 내부역할 모델이다. 애드호크러시 조직에 파견된 사용자는 내부 보안관리자에 의해 I-URA에 따라 내부 역할에 할당되고 또한 I-UWA에 따라 Work에 할당된다. 이 때 Work은 수행하는 업무 단위이다.[17] Work에는 I-WRA에 의해 해당 Work을 수행 중에 활성 가능한 Role들이 정해져 있어 사용자가 가진 역할 중에서 현재 수행하고 있는 Work에 관련된 Role 권한만을 활성화 할 수 있도록 역할 필터링(Role Filtering) 기능을 해준다. 여기서 Work을 통한 역할 필터링은 SOD(Separation Of Duty)와는 다른 개념이다. SOD는 서로 이해관계에 있는 두 개의 역할을 한 사용자가 동시에 갖지 못하게 해 부정을 미연에 방지하고자 하는 목적인 반면에 여기서 Work 개념은 사용자가 여러 역할을 갖고 있다 하더라도 이를 현재 수행하는 업무에 관련된 역할만을 활성화가 시킬 수 있게 하는 Need-To-Know를 실현하기 위한 것이라고 할 수 있다. 이 개념을 이용하면 비록 SOD



〔그림 7〕 내부 역할 모델

관계에 있지 않은 역할이라 하더라도 업무에 관계없이 필요 이상의 역할권한을 사용할 수 없게 해 최소권한의 원칙(Least Privilege Principle)을 더욱 충실히 지원할 수 있다. 또한 업무를 선택하는 것만으로도 업무를 처리하기 위해 필요한 역할들을 모두 활성화시킬 수 있어 사용자 측면에서의 편이성을 제공받을 수 있다. 각 역할은 I-PRA에 의해 자원에 대한 Permission을 갖지만 여기서도 해당 Work에 따라 역할이 객체에 대해 갖는 접근권한을 매트릭스 형태의 뷰 기능을 제공해 세밀한 접근관리가 가능하도록 한다.

3.4 제안한 모델의 정의 및 형식적 해석

3.4.1 제안된 모델의 형식적 정의

다음의 〔표 2〕는 제안된 모델을 형식적으로 정의하고 있다.

표기법

U	: User
U_i	: a set of users in the inner role hierarchy
R	: Role
R_i	: Internal Role
R_e	: External Role
P	: Permission.
P_i	: Internal Permission
P_e	: External Permission
W_i	: a set of work in the inner role hierarchy
SW_i	: a set of sub-work in the inner role hierarchy
S	: Sessions.
C	: Contexts
I-RH	: Inner Role Hierarchy
E-RH	: External Role Hierarchy
V_i	: a set of view in the inner role hierarchy

〔표 2〕 제안된 모델의 형식적 정의

정의	형식적 정의	비고
1	$P \subseteq U \times (R_e \cup R_i) \times C$	
2	$E\text{-}RH \subseteq R_e \times R_e$	External Role Hierarchy
3	$I\text{-}RH \subseteq R_i \times R_i$	Internal Role Hierarchy
4	$RH \subseteq \{E\text{-}RH \cup I\text{-}RH\}$	
5	$E\text{-}URA \subseteq U \times R_e$	External URA
6	$I\text{-}URA \subseteq U \times R_i$	Internal URA
7	$URA \subseteq \{E\text{-}URA \cup I\text{-}URA\}$	
8	$E\text{-}PRA \subseteq P_e \times R_e$	External PRA
9	$I\text{-}PRA \subseteq P_i \times R_i$	Internal PRA
10	$PRA \subseteq \{E\text{-}PRA \cup I\text{-}PRA\}$	
11	$SW_i \subseteq W_i$	
12	$I\text{-}UWA \subseteq U \times SW_i$	Internal UWA
13	$I\text{-}WRA \subseteq R_i \times SW_i$	Internal WRA
14	$USA \subseteq U \times S$	
15	$V_i \subseteq W_i \times R_i \times P_i$	View of a work

정의1은 사용자의 접근하고자 하는 자원의 접근권한은 사용자가 할당받은 외부역할과 내부역할의 합으로 이루어지고 이때 문맥(Context) 정보가 필터링 역할을 하게 됨을 나타낸다.

정의11은 업무(Work)가 다수의 하위업무(sub-work)로 이루어짐을 나타낸다. 이 업무(Work)의 개념을 통해 애드호크러시 조직내의 팀원간의 수평관계를 나타낼 수 있게 된다. 사용자는 정의12에 의해 하위업무(sub-work)단위로 할당된다. 정의13은 하위업무(Sub-work)를 수행함에 있어 필요로 하는 역할을 정의한다. 이를 통해 사용자는 업무를 수행함에 있어 필요한 역할을 활성화시킬 수 있게 된다. 정의15는 주어진 업무(Work)에서도 뷰의 맵정보에 따라 역할이 가지고 있는 퍼미션을 필터링할 수 있게 된다.

3.4.2 제안된 모델을 위한 함수

〔표 3〕은 제안된 접근제어 모델에서 필요한 정보를 얻기 위해 사용되는 함수들이다.

1) WorkFromUser 는 사용자 ui가 관계하는 업무(Work)를 알아내는 함수이다.
2) RolesFromUser 함수는 사용자 ui에게 할당되

〔표 3〕 제안된 모델의 함수

함수명	형식적 표현
WorkFromUser function : W ← U	WorkFromUser(u_i) ⊆ {w | (u_i,w) ∈ I-UWA }
RolesFromUser function : R ← U	RolesFromUser(u_i) ⊆ { r | (u_i,r) ∈ { E-URA ∪ I-URA } }
UserFromSession function : U ← S	UserFromSession(s_i) ⊆ { u | (u, s_i) ∈ USA }
RolesFromWork function : R ← W_i	RolesFromWork(u_i, w_i) ⊆ { r | (u_i,w) ∈ I-UWA ∧ (u_i,r) ∈ I-URA }
RolesFromSession function : R ← S	RolesFromSession(s_i) ⊆ {r | (∃r′≥r) [(UserFromSession(s_i), r′) ∈{ URA ∪ I-URA }] }

어 있는 역할을 얻기 위한 함수이다.

3) UserFromSession 함수는 세션 si에 연결된 사용자를 알아내는 함수 이다.
4) RolesFromWork 함수는 주어진 업무(Work)에 관계된 역할을 알아내는 함수이다.
5) RolesFromSession 함수는 세션 si에 관계하는 모든 역할을 알아내는 함수이다.

3.5 모델의 적용 예

여기서는 앞에서의 다루었던 예제를 제안한 모델에 적용해 문제점이 없는지를 검토해 보도록 하겠다. 기존 조직의 관료제적인 부서에 대한 접근제어는 Sandhu의 모델을 사용했기 때문에 여기서는 생략을 하기로 하겠다.

다만 애드호크러시 조직을 위한 제안한 모델과 유기적으로 동작하는데 문제점이 없는 지만을 검토하기로 한다.

[1단계] 외부역할의 정의

중앙의 보안관리자(SSO)는 조직의 전체에서 적용할 권한관리를 위해 Sandhu의 RBAC 모델을 사용하고 이에 필요한 Role, User, Permission 등을 정의한다.

이때 구조조정 태스크포스팀을 위한 TF1이라는 역할을 생성한다.

중앙의 SSO는 TF1역할에 PRA에 의해 태스크포스팀에서 필요한 모든 자원에 대한 권한을 할당해 주고 URA에 의해 TF1 역할 내에서 업무를 수행할 요원을 할당한다.

[2단계] 내부역할의 정의

중앙의 SSO에 의해 TF1 역할이 만들어지고 TF1 역할 내에 속한 자원과 내부 역할을 관리할 내부 SSO가 정해지면 이후 내부 역할에 관한 관리는 내부 SSO에 의해 이루어진다.

내부 SSO는 다음과 같은 일 한다.

1) 내부에서 사용할 역할을 생성한다.
2) 역할간의 상속개념을 지원하기 위한 상하관계를 정의 한다. : I-RRA
3) 정의된 역할들에게 필요한 자원에 대한 권한을 할당한다. : I-PRA
4) 태스크포스팀 내에서 업무를 수행할 구성원에 대한 역할을 할당한다. : I-URA
5) 태스크포스 팀내에서 수행해야 할 업무인 Work을 정의한다. 이 Work은 수평적인 개념이며 서로간에 상하관계를 갖지 않는다.
6) 각각의 Work들은 sub-work으로 다시 나뉜다.
7) 각각의 sub-work에 각각의 sub-work을 수행하는데 필요한 역할을 할당한다. : I-WRA
8) 다음은 각각의 sub-work에 각각의 sub-work을 담당할 사용자를 할당한다. : I-UWA

[3단계] 시스템을 이용한 권한 사용

Smith는 재무구조개선업무를 처리하기 위해 자신의 자리에 있는 PC를 통해 시스템에 접속을 하고 사용자ID와 암호를 입력해 시스템 사용에 대한 인가를 받는다. 접속화면에서는 조직의 전체적인 역할 구조도가 나타나고 일반 역할과 TF1 역할은 다른 모양으로 구별된다. Smith는 자신이 파견되어 속해 있는 구조조정 태스크포스팀을 위한 외부 역할 TF1을 선택한다. TF1 역할을 선택하면 화면은 태스크포스팀에서 정의된 업무 리스트가 보여지는 화면으로 전환된다. 이 업무들은 상호간 독립적인 업무들이며 상하관계를 갖지 않는다. 다음은 이러한 상황에서 일어날 수 있는 문제점들을 예를 통해 살펴보고 우리가 제안하는 모델과 해결 방법을 통해 이러한 문제점들이 해결될 수 있음을 보이고자 한다.

[문제점1] 업무의 비표준화로 인한 자원범위 예측의 어려움.

Ann이 자신의 업무를 완수하기 위해서는 file1에 대한 접근이 필요하나 자신이 부여받은 역할에서는 접근권한이 없다고 가정할 때 file1의 접근 권한을 갖고 있는 팀 동료인 Smith에게 일시적인 접근을 할 수 있게 Guarantee를 요청할 수 있다.

Smith의 Guarantee를 받은 Ann은 일시적으로 file1에 접근해 자신의 업무를 완수할 수 있게 된다.

[문제점2] 팀내의 자치적인 관리.

팀내의 자치적인 관리는 역할 구조를 외부역할 구조와 내부역할 구조로 분리해 내부역할의 관리는 팀내에게 하게 함으로 해서 가능하게 할 수 있다.

[문제점3] 상충되는 권한충돌의 문제.

Smith가 경제연구소에서 Manager 역할로서는 file1에 대한 접근이 불가능하였으나 태스크포스팀에서는 Finance Director 역할을 부여받아 file1에 대한 접근이 가능하다고 했을 경우 우리는 file1에 대해 상충되는 접근 권한을 갖게된 Smith의 file1에 대한 접근 가능 여부를 결정해 주어야 한다. 이때 우리는 권한충돌 해결정책을 통해 해결할 수 있다.

[문제점4] 수행중인 업무에 따라 활성 가능한 역할의 제한.

Smith는 팀내에서 내부역할 Finance Director, M&A Advisor를 할당받고 있고, 재무구조개선업무, 기업매각업무를 수행하고 있다고 가정하자.

재무구조개선업무는 다음과 같은 sub-work으로 나뉘어져 있고 각각의 sub-work에는 업무수행에 필요한 역할이 다음과 같이 할당되어 있다.

1) 회계업무 : Finance Director
2) 부채관련업무 : Debt Manager, Finance Director
3) 자산관리업무 : Assets Manager
4) 증권업무 : Stock Advisor, Assets Manager

기업매각업무에는 다음과 같은 sub-work으로 나뉘어져 있고 각각의 sub-work에는 업무수행에 필요한 역할이 다음과 같이 할당되어 있다.

1) 매수업무 : Purchase Manager, M&A Advisor
2) 매각업무 : Sale Manager, Finance Director
3) 섭외업무 : Relation Manager
4) 소송업무 : Law Advisor, Assets Manager

사용자는 I-UWA에 의해 sub-work에 할당된다.

Smith가 I-UWA에 의해 재무구조개선업무의 sub-work인 회계업무와 기업매각업무의 sub-work인 매수업무에 할당되어 있다고 가정하자.

시스템은 업무리스트에 sub-work단위가 아닌 work 단위로 보여줄 것이다. 이때 업무리스트에는 여러 개의 work이 보여지게 된다. 그러나 Smith는 회계업무 sub-work을 포함하는 재무구조개선업무와 매수업무 sub-work을 포함하는 기업매각업무만이 선택 가능하다. 두 가지 업무 중에서 Smith는 재무구조개선업무를 선택한다. Smith가 재무구조개선업무를 선택하면 Smith에게 할당된 역할 중에서 재무구조개선업무에 필요한 Finance Director 역할만이 자동으로 활성화되고 M&A Advisor 역할은 재무구조 개선업무에는 관계가 없는 역할이므로 활성화되지 않는다.

이를 통해 Smith가 할당받은 역할의 권한을 업무에 관련되어서만 사용할 수 있게 해준다.

[문제점5] 사용자 자원에 대한 임의적 접근 권한의 허용.

태스크포스팀 내에서는 주어진 자원의 공유뿐만이 아니라 사용자 자신이 생성한 자원에 대해서는 조직의 상하 구조에 관계없이 자신의 임의대로 타인에게 접근권한을 허용할 필요가 있다.

예를 들어 Ann이 팀장인 Tom에게 보고할 Report1 file을 만들어 바로 위의 상급자인 Smith에게는 Report1에 대한 접근권한을 주지 않고 팀장 Tom만이 Report1을 볼 수 있게 하길 원한다고 하자.

이를 위해 Ann이 Report1을 만들면 Report1에 대한 접근권한을 관리할 위임역할을 자동으로 만들어 준다. 그리고 Ann이 Tom을 이 위임역할에 할당할 수 있게 함으로써 역할의 개념을 유지하면서 Ann이 자신이 만든 Report1에 대한 접근권한을 Tom에게 줄 수 있도록 했다.

우리는 앞에서의 예제를 제안된 모델에 적용해 봄으로 해서 예제에서 문제점으로 제기 되었던 업무의 비표준화로 인한 문제는 Guarantee 개념을 도입해 해결했고, 자치적 관리문제는 이중의 역할구조를

사용해 해결했다. 권한충돌의 문제는 권한충돌의 해결 기법을 통해 해결했으며 업무에 따른 권한제한은 워크(Work)개념을 이용해 해결했다. 마지막으로 사용자가 생성한 자원에 대한 문제는 위임역할을 사용해 해결했다. 우리가 제안한 모델과 방법은 기존의 관료제 조직을 위한 외부역할 모델과도 유기적으로 사용되어질 수 있음을 보았다.

Ⅳ. 모델의 평가

제안된 모델이 애드호크러시 조직을 위한 접근제어 모델로서 타당하다는 것을 다른 접근제어 모델과의 비교를 통해 평가를 하고자 한다. 모델을 비교함에 있어 역할 개념이 없는 다른 모델과의 비교는 큰 의미가 없어 여기서는 지금까지 연구된 역할개념을 포함하는 접근제어 모델 중에서 팀의 특성을 고려한 TMAC 모델과 C-TMAC 모델과 비교를 할 것이다.

비교의 범위는 각 모델들을 이용해 앞에서 살펴본 애드호크러시 조직에 대한 접근제어 적용시 각각의 문제점들을 얼마나 잘 해결할 수 있는 지로 평가했다. 우리는 [표 4]를 통해 2.2절에서 살펴본 애드호크러시 조직을 위한 접근제어에서 발생되는 문제점들을 각각의 접근제어 모델들이 얼마나 잘 해결하고 있는지를 비교해 보았다.

[표 4]에서와 같이 우리는 제안한 애드호크러시 조직을 위한 접근제어 모델을 통해 기존의 다른 접근제어 모델에서 해결할 수 없었던 문제점들을 해결할 수 있음을 볼 수 있다.

Ⅴ. 결 론

공공기관이나 금융기관과 같이 비교적 안정되어 있어 집단간의 통합이 그리 중요하지 않을 경우 구조적인 관료제 조직이 효과적일 수 있다.

그러나 업무 수행이 표준화되어 있지 않고 업무수행 중 환경변화가 많으며 다양한 부서의 전문가들에 의해 업무가 이루어지는 기업의 구조조정 팀과 같은 경우 태스크포스와 같은 애드호크러시 조직의 특성을 갖게 된다.

그러나 지금까지의 컴퓨팅 자원에 대한 접근제어를 위해 연구된 RBAC 모델은 이러한 애드호크러시 조직의 특성을 충분히 반영하지 못하고 있다. 이에 새로운 조직의 모델인 애드호크러시 조직의 특성을 충분히 반영하는 접근제어 모델 개발과 구현이 필요하게 된다. 우리는 본 논문에서 애드호크러시 조직의 특성을 분석하고 이러한 특성들로 인해 접근제어시 문제가 되는 사항들을 도출해 보았다.

이를 바탕으로 우리는 도출된 문제들을 해결하기 위해 Guarantee 개념을 도입해 Trusted Group 내의 팀원간의 유연한 권한 사용이 가능하게 했으며, 팀의 자치적인 권한관리를 위해 이중화된 역할 구조를 사용해 가능하게 했다. 팀원이 갖게되는 복합적인 권한으로 인해 발생되는 권한의 충돌을 해결하기 위해 권한충돌 해결정책을 사용하여 해결하였으며 팀원간의 수평적인 개념을 반영하기 위해 워크(Work) 개념을 도입했다. 또한 팀내에서 사용자 자신이 생성하는 객체에 대한 관리를 위임(Delegation)역할을 사용해 역할개념을 유지하는 임의적 접근제어를 가능하게 했다. 우리가 제안한 접근제어 모델은 기존의 관료제(Bureaucracy) 조직도 포함되는 모델이기 때문에 이들 두 조직을 통합적으로 연계해서 사용할 수 있는 접근제어 모델이다. 또한 예를 통해 이러한 문제들이 제안된 모델의 적용과 해결방법을 통해 해결될 수 있음을 보였다.

우리가 제시한 모델의 부족한 점은 워크플로우에 대한 개념이 아직 부족하다는 것이다. 실제 접근제어에서 워크플로우에 대한 개념 수용이 필요하며 우리의 향후 연구는 이러한 워크플로우 개념을 보완하는 데 있다. 특히 인터넷상에서 복수의 웹서버를 이용하는 환경에서 우리의 모델을 이용하여 접근제어를 구현하는 연구를 진행하고 있다.

[표 4] 기존 모델과의 비교

문제점	Sandhu Model[5]	TMAC[7]	C-TMAC[6]	Our Model
업무의 비표준화	미리 정해져야함	고려 없음	고려 없음	개런티 개념도입
자치적 관리	부분적 허용	불허	불허	역할 구조의 이중화
권한충돌	언급 없음	언급 없음	언급 없음	권한충돌 해결 기법
업무권한의제한	없음	팀역할 개념 사용	시간,장소등에 따른 권한	워크개념 도입
사용자 자원	불가	불가	불가	위임역할을 이용한 위임기법

참 고 문 헌

[1] Gareth R. Jones, "Organizational Theory 3rd Edition", Prentice Hall, 2001.

[2] Robert H., Jr. Waterman, "Adhocracy", W.W. Norton & Company, 1993.

[3] U.S. Department of Defence, Department of Defence Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, National Computer Security Center, 1985.

[4] David D. Clark and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", Proc. of the 1987 IEEE Symposium on Security and Privacy, 1987

[5] Ravi Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", IEEE Computer Magazine Vol. 29, 1996.

[6] Georgiadis C., Mavridis I., Pangalos G. and R.Thomas, "Flexible Team-based Access Control Using Contexts", in Proc. of the 6th SACMAT, 2001.

[7] Rosan K. Thomas, "Team-based Access Control(TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments", ACM RBAC'97, 1997.

[8] Horst F. Wedde, "Modular Authorization", University of Dortmund, SACMAT, 2001.

[9] John Hale, Pablo Galiasso, Mauricio Papa and Sujeet Shenoi, "Security Policy Coordination for Heterogeneous Information Systems", Proc. of the 15th Annual Computer Security Applications Conference, 1998.

[10] Myong H. Kang, S. Park and Judith N. Froscher, "Access Control Mechanisms for In-ter-Organizational Workflow", Proc. of the 6th SACMAT, 2001.

[11] Sejong Oh, Seog Park, "Task-Role Based Access Control(T-RBAC):An Improved Access Control Model for Enterprise Environment", DEXA, 2000.

[12] S. Jajodia, P. Samarati, ML Sapino, and VS Subrahmanian, "Flexible Support for Multiple Access Control Policies", ACM Transactions on Database Systems, 2001.

[13] Elisa Bertino, Sushil Jajodia, Pierangela Samarati, "Supporting Multiple Access Control Policies in Database Systems", Proc. of the IEEE Symposium on Security and Privacy, 1996

[14] Rabitti, Fausto, Bertino, Elisa, Kim, Won, Woelk, Darrell, "A model of authorization for next-generation database systems", ACM Transactions on Database Systems Vol. 16, No. 1, 1991.

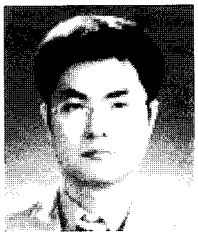
[15] HongHai Shen, Prasun Dewan, "Access Control for Collaborative Environments", Proc. of ACM CSCW, 1992.

[16] Won Bo Shim, Seog Park, "Implementing Web Access Control System for the Multiple Web Servers in the Same Domain Using RBAC Concept", proc. of the eighth IEEE ICPADS, 2001.

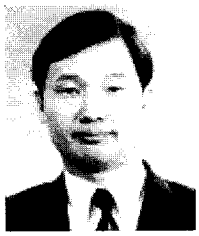
[17] Won Bo Shim, Seog Park, "The Work Concept RBAC Model for the Access Control of the Distributed Web Server Environment", Web Intelligence: Research and Development, LNAI 2198, 2001.

[18] Rosan K. Thomas, Ravi Sandhu "Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management", 11th IFIP Working Conference on Database Security, 1997.

〈著者紹介〉



심 완 보 (Won Bo Shim) 정회원
1985년 2월 : 한양대학교 전자공학과 학사
1986년 5월 : 스티븐스공과대학(SIT) 전산과학 석사
1997년 3월~현재 : 서강대학교 컴퓨터학과 박사과정
1989년 5월~1995년 8월 : 삼성 SDS 정보기술연구소 선임연구원
1995년 9월~현재 : 충청대학 컴퓨터학부 조교수
〈관심분야〉 접근제어, 웹보안, 정보보호



박 석 (Seog Park) 정회원
1978년 2월 : 서울대학교 계산통계학과 학사
1980년 2월 : 한국과학기술원 전산학과 석사
1983년 9월 : 한국과학기술원 전산학과 박사
1983년 9월~현재 : 서강대학교 컴퓨터학과 교수
1989년~1991년 : Univ. of Virginia 방문교수
2000년 4월~현재 : DASFAA Steering Committee 멤버
〈관심분야〉 실시간 데이터베이스, 데이터베이스보안, 데이터웨어하우스, 웹과 데이터베이스, 접근제어