Journal of Korea Multimedia Society Vol. 8, No. 2, February 2005(pp. 211-224)

기업환경의 접근제어를 위한 확장된 GTRBAC 모델

황유동[†], 박동규[††]

요 약

인터넷과 웹이 활성화됨으로써 사용자는 문서, 디렉토리, 데이터베이스, 웹 페이지 등과 같은 자원들을 액세스하는 것이 훨씬 더 쉬워졌다. 그러나 이로 인하여 네트워크의 인증, 자원들을 액세스하기 위한 권한 허가, 데이터의 정책과 보안 그리고 보안 시스템의 무결성과 같은 몇 가지의 중대한 보안 문제들이 생기게 되었다. 본 논문에서는 기업 환경의 접근제어를 위하여 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있는 GTRBAC(Generalized Temporal Role Based Access Control) 모델에 부역할 개념을 적용한 확장된 GTRBAC (Extended GTRBAC)모델을 제안한다. 제안 모델은 부역할 계층을 사용하여 하위 역할에 할당된 권한을 상위 역할에 할당된 사용자가 모두 상속하여 실행할 수 없도록 하여 권한의 남용을 방지하고 최소권한의 원칙을 지킬 수 있도록 하여 기업 환경의 특성에 따라 다양하고 정교한 접근제어 정책을 적용할 수 있도록 한다.

Extended GTRBAC Model for Access Control Enforcement in Enterprise Environments

Yu-Dong Hwang[†], Dong-Gue Park[††]

ABSTRACT

With the wide acceptance of the Internet and the Web, volumes of information and related users have increased and companies have become to need security mechanisms to effectively protect important information for business activities and security problems have become increasingly difficult. This paper proposes a improved access control model for access control enforcement in enterprise environments through the integration of the temporal constraint character of the GT-RBAC model and sub-role hierarchies concept. The proposed model, called Extended GT-RBAC(Extended Generalized Temporal Role Based Access Control) Model, supports characteristics of GTRBAC model such as of temporal constraint, various time-constrained cardinality, control flow dependency and separation of duty constraints(SoDs). Also it supports unconditional inheritance based on the degree of inheritance and business characteristics by using sub-roles hierarchies in order to allow expressing access control policies at a finer granularity in corporate enterprise environments.

Key words: Access Control(접근제어), RBAC(역할기반접근제어), Temporal Constraint(임시제약), Sub-Role(부역할), GTRBAC(일반화된 임시역할기반접근제어)

1. 서 론

인터넷과 웹이 활성화됨으로써 사용자는 문서, 디렉토리, 데이터베이스, 웹 페이지 등과 같은 자원들을 액세스하는 것이 훨씬 더 쉬워졌다. 그러나 이로 인하여 네트워크의 인증, 자원들을 액세스하기 위한

※ 교신저자(Corresponding Author) : 황유동, 주소 : 충남 아산시 신창면(336-745), 전화 : 041)530-1347, FAX : 041) 530-1548, E-mail : coppermilk@sch.ac.kr
접수일 : 2004년 8월 17일, 완료일 : 2004년 11월 5일
[†] 순천향대학교 전기전자공학과

[††] 정회원, 순천향대학교 정보기술공학부 교수
(E-mail : dgpark@sch.ac.kr)
※ 본 논문은 정보통신부와 정보통신연구진흥원에서 지원하는 기초기술연구지원사업을 통해서 연구된 과제임.

권한 허가, 데이터의 정책과 보안 그리고 보안 시스템의 무결성과 같은 몇 가지의 중대한 보안 문제들이 생기게 되었다.

정보 보안은 시스템들이 인증(authentication), 접근제어(access control), 무결성(integrity), 신뢰성(confidentiality), 그리고 부인(non-repudiation)과 같은 5가지의 중요한 서비스를 제공하도록 요구한다. 이 중 접근제어는 컴퓨터내의 자원, 통신 자원 및 정보 자원 등에 대하여 사용, 변경, 조회 등의 작업을 할 수 있는 능력을 가능하게 하거나 제한할 수 있는 수단으로 식별 및 인증된 사용자만이 허가된 범위 내에서 시스템 내부의 정보에 대한 접근을 허용하는 기술적 방법이다.

접근제어를 위해 개발된 보안 정책으로는 임의 접근 통제(DAC : Discretionary Access Control)[1], 강제적 접근 통제(MAC : Mandatory Access Control), 역할 기반 접근 통제(RBAC : Role Based Access Control)[2,3] 및 행위 기반 접근 통제(ABAC : Activity Based Access Control)[4,5] 모델과 기업 환경에 적합한 과업-역할 기반 접근 통제 모델(T-RBAC : Task-Role Based Access Control)[6] 모델 등이 있다.

그러나 이들 모델들은 모두 기업 환경에 대한 애플리케이션에서 시간 제약에 따른 자원의 사용제한을 하지 못한다는 제약이 있고, 역할 계층상에서 상위 역할에 배정된 사용자가 하위 역할의 모든 접근 권한을 상속받게 되어 불필요한 권한의 실행을 허가하게 되어 최소 권한 원칙을 위배하게 되는 제약이 있다.

본 논문에서는 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있는 GTRBAC (Generalized Temporal Role Based Access Control) [10-12] 모델에 부역할(sub role)[7,8] 개념을 적용하여 하위 역할에 배정된 권한을 상위 역할에 배정된 사용자가 모두 상속하여 실행할 수 없도록 하여 권한의 남용을 방지하고, 최소권한의 원칙을 지킬 수 있는 확장된 GTRBAC (Extended GTRBAC) 모델을 제안한다.

본 논문에서는 2장에서 기존에 연구되어왔던 접근제어 모델들을 분석한다. 3장에서는 새로운 접근 통제 모델인 Extended GTRBAC모델에 대해 살펴보고 4장에서는 제안된 모델의 정형적 명세 및 검증을 한 후 5장에서 실제 기업 환경에서 Extended GTRBAC 모델을 이용한 접근제어의 예를 보여주며, 6장에서 결론을 유도한다.

2. 기존 접근제어 모델의 분석

이 장에서는 접근제어와 관련이 있는 기존 연구들을 재검토하고 그들이 기업 환경에 적용될 때 제한사항들을 분석한다. 접근제어의 기본 목적은 권한이 부여된 사용자만이 정보 자원에 접근할 수 있는 기법을 제공하는 것이다.

접근제어를 위한 보안 정책으로는 역할기반 접근제어(Role Based Access Control : RBAC) 및 행위기반 접근제어(Activity Based Access Control : ABAC)모델과 기업 환경에 적합한 과업-역할기반 접근제어 모델(Task-Role Based Access Control : T-RBAC), 시간(기간과 주기)에 따른 제약과 역할의 활성화/비활성화, 이벤트, 트리거를 이용하여 자원의 사용을 제한하여 최소 권한 원칙을 이행 할 수 있는 GTRBAC(Generalized Temporal Role Based Access Control)모델, 하위 역할에 배정된 권한을 상위 역할에 배정된 사용자가 모두 상속하여 실행 할 수 없도록 권한의 상속 정도에 따라 하나의 역할을 여러 개로 나누는 권한상속 제한 역할계층 모델이 있다.

역할기반 접근제어(RBAC)[2,3]는 사용자와 자원 관리를 경감시키기 위해 사용된다. 역할기반 접근제어에서 접근 권한은 역할과 관련이 있으며 그리고 사용자는 적절한 역할에 할당된다. 역할기반 접근제어는 접근제어 요구 사항을 지정하는 첫 번째 수단으로서 역할 추상화를 사용한다. 역할을 관리하는 동안에, 허가들은 역할들에 할당되고, 사용자들은 역할에 할당된다. 허가는 정보에 특정한 오퍼레이션을 수행할 능력을 승인하는 것이다. 현실 세계에서, 하나의 역할은 조직 내에서 하나의 직무 기능으로 정의할 수 있으며, 그 역할에 할당된 사용자에 부여된 권한과 책임을 의미한다. 하나의 역할 계층(role hierarchy)은 일반적으로 조직의 관리 구조에 따라서 역할사이의 권한 상속관계를 나타낸다. 역할 계층은 허가 권한 시스템과 유사하기 때문에 기업 조직 구조의 모델링에 적합하다. 그러나 역할기반 접근제어는 접근 권한의 동적 활성화와 응용 레벨 제약의 명세를 필요로 하는 워크플로우(workflow)를 고려하지 않고 있다.

행위기반 접근제어(ABAC)[5,7]는 워크플로우에 의해서 표현된 공동 작업 환경을 위하여 연구된 것으로 이는 공통 목표를 달성하기 위하여 결합된 활동의 집합으로 정의된다. DAC, MAC 및 RBAC 모델의 경우는 접근 권한의 부여시점에서 권한이 활성화(activate)되어 임의의 시점에서 사용 가능한 반면에 행위기반 접근제어 모델에서는 사용자에 대한 접근 권한 할당(access right assignment)과 접근 권한 활성화(access right activation)로 분리된다. 어떤 사용자가 워크플로우 내의 과업에 대한 실행권한을 부여받았다하더라도 그 권한의 사용은 워크플로우의 진행 상태에 따라 제약을 받는다. 행위기반 접근제어 모델은 애플리케이션 레벨 제약을 위한 명세를 제공하고 현실 세계의 무결성 규칙의 구현을 지원한다. 그러나 행위기반 접근제어는 기업 환경에서 워크플로우에 속하지 않는 많은 작업들을 다루지 않고 있어 사용이 제한적이다.

과업 역할기반 접근제어 모델(Task-Role Base Access Control Model ; T-RBAC)[6]은 역할기반 접근제어 모델을 기초로 하여 행위기반 접근제어 모델을 통합한 모델이다. 과업 역할기반 접근제어 모델과 역할기반 접근제어 모델의 가장 큰 차이점은 접근 권한(access rights)을 부여하는 방법이다. 역할기반 접근제어에서는 접근 권한이 직접 역할에 부여되나, 과업-역할기반 접근 통제 모델에서는 접근 권한이 그 역할이 수행하는 과업(task)을 통해 부여된다는 점이다. 과업 역할기반 접근제어 모델에서 과업은 3개의 클래스로 분류된다. 클래스 S에 속하는 과업은 계승시킬 수 있으며, 그들의 접근 권한은 역할 계층에서 더 높은 역할로 상속된다. 클래스 P에 속하는 과업은 단일 역할에 할당 가능한 과업으로 역할계층에서 상위의 역할로 상속되지 않는다. 클래스 W에 속하는 과업은 활동적인 보안 정책으로 워크플로우 메커니즘에 의해서 관리된다. 그래서 과업 역할기반 접근제어 모델은 기업 환경에 대하여 행위기반 접근제어 모델과 역할기반 접근제어 모델의 제한 사항들을 해결한다. 그러나 과업 역할기반 접근제어 모델은 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한하여 최소 권한 원칙을 이행할 수 있는 방법을 제공하지 않아 최근의 변화하는 기업 환경에 적용하기에는 무리가 따른다.

GTRBAC 모델(Generalized Temporal Role Based Access Control Model)[10-12]은 TRBAC 모델(Temporal Role Based Access Control)[9]을 확장한 모델로, 역할의 사용과, 역할 - 권한 할당, 역할 활성화를 포함한 주기적이고 지속적인 시간의 제약 집합을 위한 명세를 포함한다.

GTRBAC 모델의 특징을 정리하면 다음과 같다.

-. Temporal constraints on role enabling/disabling : 이 제약은 지연시간 또는 일정 기간 동안 사용자 - 역할 또는 역할 - 권한에 할당된 역할이 가능하도록 한다.
-. Temporal constraints on user-role and role-permission assignments : 지정된 지연시간 또는 기간 동안 사용자와 권한을 역할에 할당한다.
-. Activation constraints : 이 제약은 사용자들이 역할을 활성화 할때 제한을 한다. 명세된 가간 동안 역할의 활성화를 제한하거나 세션 상에서 역할의 활성화 수를 제한한다.
-. Run-time events : 런타임 이벤트 들은 관리자가 GTRBAC 이벤트들을 동적으로 시작하거나, 역할 활성화 제약들 또는 기간을 가능하도록 한다.
-. Constraint enabling expressions : GTRBAC 모델은 가능 또는 불가능하게 하는 기간 제약들과 역할 활성화 제약들을 포함한다. 기간 제약들은 사용자 - 역할 할당관계와 역할 - 권한 할당관계들에 의해 역할이 가능하게 한다.
-. Triggers : 트리거 들은 다양한 임시 이벤트들 사이의 종속성을 표현하기 위하여 트리거 프레임 웍을 제공하여 시스템에 의해 동적으로 변화하는 접근제어 요구사항에 적절히 대응할 수 있다.

TRBAC 모델뿐만 아니라 GTRBAC 모델에서도 임시 제약들과 역할 계층 사이의 상호작용이 중요한 문제이다.

GTRBAC 모델은 역할들에 대한 임시 제약들의 존재로 I(permission - inheritance - only hierarchy), A(role - activation - only hierarchy), I-A (permission - inheritance - activation hierarchy) 역할 계층과 같은 부분 역할 계층이 존재하고 이들 역할 계층은 역할의 활성화/비활성화 제약과 시간 제약을 이용하여 제한된 상속 기능을 제공한다.

I 역할 계층은 상위 역할에 사용자가 할당되고 역할이 활성화 되면 세션 상에서 하위 역할의 활성화와 상관없이 모든 권한이 상위 역할로 상속되는 역할 계층이고 A 역할 계층은 상위 역할에 할당된 사용자가 하위 역할을 활성화 할 수 있는 역할계층이다.

I-A 역할 계층은 I 역할 계층과 A 역할 계층의 혼합형 부분 역할 계층이다.

GTRBAC 모델에서는 역할 계층에서 최소 권한 원칙에 위배되지 않도록 하기 위하여 활성화 가능한 역할 집합을 계산 하여야 하는데 특히 I 역할계층, A 역할계층, I-A 역할계층이 혼합되어있을 경우 활성화 가능한 역할의 집합을 계산하는 것은 매우 복잡하게 된다.

이러한 여러 역할 계층이 혼합된 역할계층에서 사용자에게 할당된 역할에 의해 활성화 될 수 있는 역할의 집합을 UAS(uniquely activable set)[12]이라 한다.

UAS는 단일 세션에서 사용자에 의해 활성화 될 수 있는 역할들의 집합을 말하고 역할 계층을 통하여 사용자에 의해 활성화 될 수 있는 역할집합을 결정할 수 있도록 해줌으로써 최소 권한 원칙을 유지할 수 있도록 도와준다.

다음 그림 1은 GTRBAC 모델의 일반적인 역할 계층을 위한 UAS 계산의 예이다.

그림 1의 a)는 GTRBAC 모델의 일반적인 역할 계층이고 b), c), d), e)는 역할계층의 UAS를 계산하기 위하여 역할계층(그림 1의 a))을 분할한 예이다. 그림 1의 일반 역할 계층을 위해 다음의 단계로 UAS를 계산하고 최종적으로 계산된 UAS가 일반 역할 계층(그림 1의 a))의 UAS가 된다.

step 1 : 분할된 역할 계층 b), c), d), e)의 UAS를 각각 계산한다.

역할계층 b) : L1 = {{r3},{r2},{r1}}

역할계층 c) : Lh2 = {{t1},{r2},{r3},{t1, r2},{t1, r3}}

역할계층 d) : Lh3 = {{r3},{s1},{t1},{s1, r3},{r3, t1}}

역할계층 e) : Lh4 = {{r3},{s1},{s2},{s3},{r3, s1}, {r3, s2},{r3, s3},{s1, s2}, {s1, s3},{r3, s1, s2}}

step 2 : L1과 Lh2로 UAS를 계산한다.

B = UAS(L1, t) ⊓ UAS(Lh2, t) = {{r3}, {r2}, {t1, r2}, {t1, r3}}.

(UAS(L1, t)/B) ⊗ (UAS(Lh2, t)/B) = {{r1}} ⊗ {{t1}} = {{r1, t1}}.

C = empty. UAS(H12, t) = I/C = I = {{r3}, {r2}, {r1}, {t1}, {r1, t1}, {r3, t1}, {t1, r2}}.

step 3 : H12와 Lh3으로 UAS를 계산한다.

B = UAS(H12, t) ⊓ UAS(Lh3, t) = {{r3}, {t1}, {r1, t1}, {r3, t1}, {t1, r2}}.

(UAS(H12, t) - B) ⊗ (UAS(Lh3, t) - B) = {{r2}, {r1}} ⊗ {{s1}} = {{r2, s1}, {r1, s1}}.

I = {{r3}, {r2}, {r1}, {t1}, {r1, t1}, {r3, t1}, {t1, r2}, {s1}, {r2, s1}, {r1, s1}}.

C = {r3, t1}. UAS(H13, t) = I/C = {{r3}, {r2}, {r1}, {t1}, {r1, t1}, {t1, r2}, {s1}, {r2, s1}, {r1, s1}}.

step 4: H13과 Lh4로 UAS를 계산한다.

B = UAS(H13, t) ⊓ UAS(Lm4, t) = {{r3}, {s1},

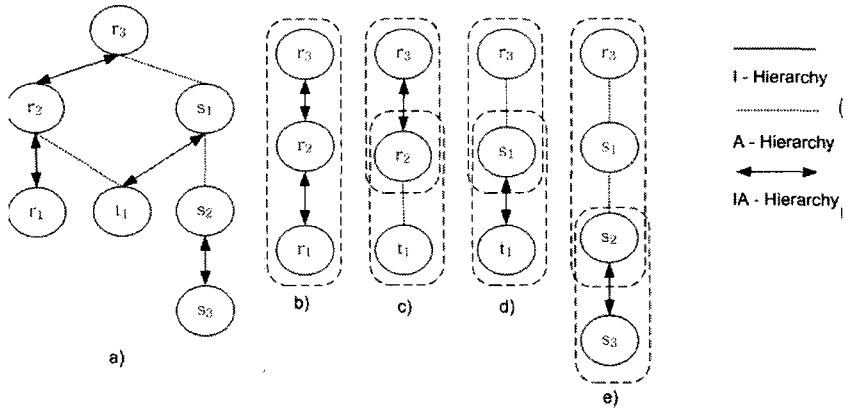


그림 1. GTRBAC 모델의 일반적인 역할 계층을 위한 UAS 계산

{r2, s1}, {r3, s2}, {s1, s2}}.
(UAS(H13 -B) ⊗ (UAS(Lm4, t) -B) = {{r2}, {r1}, {t1}, {r1, t1}, {t1, r2}} ⊗ {{ s2}} = {{r2, s2} {r1, s2}, {t1, s2}, {r1, t1, s2}, {t1, r2, s2}}.
C = empty. UAS(H13, t) = I/C = I = {{r3}, {r2}, {r1}, {t1}, {r1, t1}, {t1, r2}, {s1}, {r2, s1}, {r1, s1}, {s2}, {r3, s2}, {s1, s2}, {r2, s2} {r1, s2}, {t1, s2}, {r1, t1, s2}, {t1, r2, s2}}

위의 내용으로 알 수 있듯이 GTRBAC 모델은 역할 활성화/비활성화와 이벤트 제약, 트리거를 이용하여 기존 모델에서는 불가능했던 접근 권한의 동적 활성화와 응용 레벨 제약의 명세를 필요로 하는 워크플로우(workflow)를 고려할 수 있게 되었고, 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있게 되었다.

그러나 GTRBAC의 역할 계층 또한 기존 모델과 마찬가지로 I 역할 계층에서는 권한이 상위 역할로 무조건 상속되고, A 역할 계층에서는 상위 역할이 하위역할을 활성화 할 수 있으므로 하위 역할의 권한을 모두 획득할 수 있게 되어 권한의 남용을 방지할 수 없고, 최소권한 원칙을 위배하게 되는 단점이 있다. 또한 이러한 부분 역할 계층이 하나의 역할 계층에 다양하게 혼합되어 존재 한다면 세션 상에서 사용자가 역할을 활성화 하였을 때 활성화 된 역할에게 역할 계층에 의해서 어떤 권한들이 허가되는지를 결정하는 복잡한 계산을 해야 한다.

권한 상속 제한 역할 계층 모델[7,8]은 기존 모델들의 역할계층에서 권한의 무조건 상속에 따른 최소권한원칙 위배의 문제점을 해결하기 위해 제시되었다.

권한 상속 제한 역할 계층 모델은 하나의 역할 계층을 업무특성과 권한 상속 정도에 따라 조직공통 역할, 부서 공통 역할, 상속제한 역할, 고유 역할과 같은 네 개의 부역할로 나누어 보안 관리자가 권한 상속을 쉽게 통제 할 수 있도록 해준다.

이 모델은 역할 계층에서 다음과 같은 장점을 가진다. 첫째, 권한 상속의 관점에서 역할의 종류를 분류하여 부 역할로 분할하였기 때문에 하위역할이 상위역할로 무조건적으로 상속되는 것을 제한하는 상속 제한 기능을 제공한다. 둘째, 권한 상속을 유효권한과 명시적 권한의 상속으로 구분하여 나타냈으며, 이는 상속제한 역할 구조를 이용하여 무조건적인 상속을 방지함으로써 권한 남용문제를 해결하고 최소권한의 원칙을 유지하는 장점이 있다. 셋째, 역할 계층의 관리에 있어 기업 내에 존재하는 여러 역할 계층으로 분리하여 관리하는 기존의 방안 보다, 조직체계와 유사한 역할계층 안에 업무별, 직위별 역할 계층을 표현하여 조직체계와 유사한 모델링이 가능하게 한다.

또한 이 모델에서는 사용자 대 사용자 위임의 전체 위임과 부분 위임을 가능하게 하고 역할 대 역할 위임에서도 전체 위임과 부분 위임을 가능하게 한다.

이 모델은 권한의 상속을 제한하고 위임을 관리할 수 있는 장점은 있지만, 기존의 접근제어 모델과 마찬가지로 접근 권한의 동적 활성화와 응용 레벨 제약의 명세를 필요로 하는 워크플로우(workflow)를 고려하지 않고, 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있는 방법을 제공하지 않아 최근의 변화하는 기업 환경에 적용하기에는 무리가 따른다.

위 내용으로 각 접근제어 모델들이 장점을 가지고 있지만 기업 환경에 적용하기에는 여러 가지 제한사항들이 있음을 알 수 있다.

본 논문에서는 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있고, 역할 활성화와 이벤트, 트리거를 이용하여 사용자 수를 제한하고 워크플로우에 해당하는 작업을 다룰 수 있는 장점을 가지는 GTRBAC(Generalized Temporal Role Based Access Control)모델에 부역할(sub role) 개념을 적용하여 하위 역할에 배정된 권한을 상위 역할에 배정된 사용자가 모두 상속하여 실행할 수 없도록 하여 권한의 남용을 방지하고 최소권한 원칙을 이행할 수 있도록 하는 확장된 GTRBAC (Extended GTRBAC) 모델을 제안한다.

3. Extended GTRBAC(Extended Generalized Role Based Access Model)모델

위의 내용으로 알 수 있듯이 기업 환경에 접근제어를 적용하기 위해서는 다음과 같은 조건을 만족할 수 있는 접근제어 모델이어야 한다.

-. 역할 활성화/비활성화와 이벤트 제약, 트리거를 이용하여 접근 권한의 동적 활성화가 가능해야 한다.

-. 응용 레벨 제약의 명세를 필요로 하는 워크플로우(workflow)를 고려할 수 있어야 한다.
-. 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있어야 한다.
-. 역할 계층에서 하위 역할의 모든 권한이 상위 역할로 상속 되는 것을 제한 할 수 있어야 한다.

본 논문에서는 기업 환경에 적합한 접근제어를 적용하기 위하여 위의 조건을 만족하는 확장된 GTRBAC 모델을 제안한다.

제안된 (Extended GTRBAC) 모델은 다음 그림 2와 같이 표현 할 수 있다. GTRBAC 모델의 특징인 시간제약과 역할 활성화, 이벤트, 트리거 등의 제약은 그림 2의 "Temporal Constraint"로 표현하고, 하위 역할에 할당된 권한의 상속을 제한하고 하위 역할을 활성화하여 활성화되는 역할에 할당되는 권한의 제한을 위하여 하나의 역할을 여러 개의 부역할로 나누었음을 알 수 있다.

3.1 역할(Role)

그림 2의 Extended GTRBAC 모델에서 사용자-역할 할당관계인 URA는 부역할 중 고유 역할에 사용자를 할당하고 권한-역할 할당관계인 PRA는 권한에 각 부역할을 할당한다.

제안 모델에서 역할은 할당되는 권한에 따라 조직

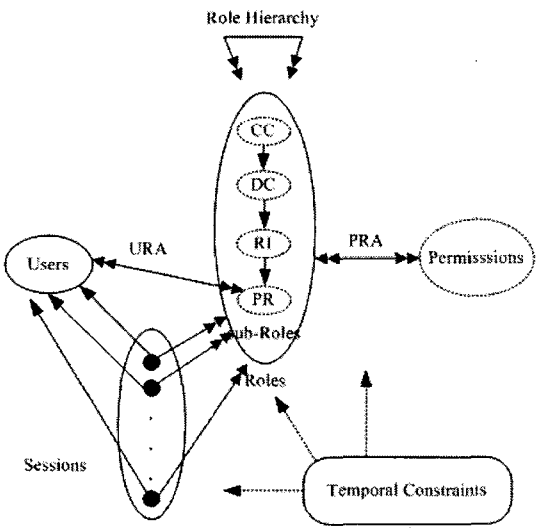


그림 2. Extended GTRBAC 모델

공통 역할(CC : Corporate Common), 부서 공통 역할(DC : Department Common), 상속 제한 역할(RI : Restricted Inheritance), 고유 역할(PR : Private Role)로 나누어진다.

조직 공통 역할과 부서 공통 역할은 역할 계층에서 하위의 역할에 할당된 권한이 제한 없이 상위의 역할로 상속되고, 상속 제한 역할은 지정된 상위 역할까지만 권한이 상속되며, 고유 역할은 상위 역할로 상속되지 않는 권한이 할당된 역할이다.

각 부역할은 다음 표 1과 같은 특징을 가진다.

또한 각 부역할 사이에서도 계층관계가 존재한다.

부역할 사이에 존재하는 부역할 계층은 각 부역할 사이의 권한 상속 관계를 나타내는 것으로 URA에 의하여 고유 역할이 할당되는 사용자에게 할당되어야 할 모든 권한이 상속되고 제한할 필요가 없다.

확장된 GTRBAC 모델에서 부역할의 상속정도와 부 역할에 배정된 권한 특징은 권한 상속 제한 역할 계층 모델과 동일하다.

상속제한(RI) 역할은 역할 계층 내에서 상위 역할로 권한의 제한 적 상속이 가능하다. 역할계층에서 역할이 상속 될 수 있는 최 상위 역할을 제약조건(Constraint Condition)으로 지정하여 지정된 상위 역할까지만 권한이 상속 된다.

3.2 역할계층 (Role Hierarchy)

제안모델에서 역할 계층은 GTRBAC 모델의 역할 계층과 동일하게 표현 할 수 있고, 각 역할에는 부역할 계층이 존재한다.

부역할 계층은 네 개의 부역할 사이의 권한 상속 관계를 나타내는 것으로 사용자에게 할당되어야 할 모든 권한이 상속되는 것으로 제한을 할 필요가 없다.

다음 그림 3은 기존 모델의 URA와 제안하는 모델의 URA를 비교하고, 부역할 사이의 역할 계층을 나타낸다.

그림 3의 좌측 부분에서 u1, u2는 사용자를 의미하며, r1, r2는 역할을 의미한다. 이때 역할 r2는 역할 r1보다 역할 계층에서 상위에 있는 역할이다. 역할 r1에서 역할 r2로의 실선 화살표는 역할 r1에 할당된 모든 권한이 역할 r2로 상속됨을 의미한다. 또한 그림에서 보이는 것처럼 기존 모델들에서는 역할을 사용자에게 할당한다.

표 1. 부역할 분류 기준

부 역할	상속의 정도	부 역할에 배정된 권한 특징
조직공통 (CC)	제한이 없다.	-. 조직 내 모든 사용자에게 허가된 권한 -. 상위 역할은 하위 역할의 모든 권한을 상속
부서 공통 (DC)	제한이 없다.	-. 부서에 속한 사용자들에게만 허가된 권한 -. 상위 역할은 하위 역할의 모든 권한을 상속
상속제한 (RI)	제한(지정된 단계 만큼)적이다.	-. 역할 분석과 설계 과정에서 상속이 제한되는 권한에 대한 조사 필요 -. 하위 역할의 권한이 지정된 상위 역할까지만 상위로 상속 -. 역할 간에 제한적 상속이 가능함
고유역할 (PR)	상속될 수 없다.	-. 상위 역할로 상속이 이루어지지 않는 권한을 할당

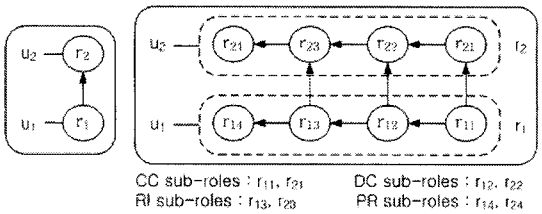


그림 3. URA와 부역할의 권한 상속 관계

그림 3의 우측 부분은 제안모델의 URA와 역할 계층에서 상속 관계를 간단히 표현 하였다.

먼저 기존의 역할을 조직공통(CC : r11, r21)역할, 부서공통(DC : r12, r22)역할, 상속제한(RI : r13, r23)역할, 고유(PR : r14, r24)역할과 같은 네 개의 부역할로 나누고 사용자에게는 상위역할로 상속되지 않는 권한을 할당하는 고유역할을 할당한다.

그림에서 네 개의 부역할 사이의 실선 화살표는 각 부역할 사이의 권한 상속 관계를 나타낸다. 즉, 부역할 사이의 역할 계층은 고유역할이 최상위에 존재하고, 상속제한 역할, 부서공통 역할, 조직공통 역할의 순서가 존재함을 알 수 있다. 이러한 부역할 사이의 상속관계로 인하여 사용자에게는 고유 역할만을 할당해도 모든 권한을 할당 받게 된다. 그림에서 상속제한 역할 r13에서 r23으로의 점선 화살표는 하위 역할에 할당된 권한이 상위 역할로 상속이 제한된다는 의미이다.

위 그림 4, 5, 6은 기존 모델의 세가지 역할 계층과 제안 모델의 역할 계층의 예를 보여준다. 그림에서 역할 그룹과 역할 그룹 사이의 선은 역할 계층의 종류(실선 : I 역할계층, 점선 : A역할계층, 양쪽 화살표 : IA 역할계층)를 나타내며, 역할 계층에서 하위 역할의 부 역할과 상위 역할의 부역할 사이에 존재하는

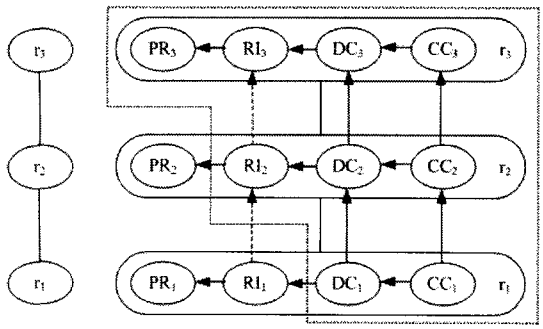


그림 4. 제안 모델의 I-역할계층에서 사용자에게 할당되는 권한 집합

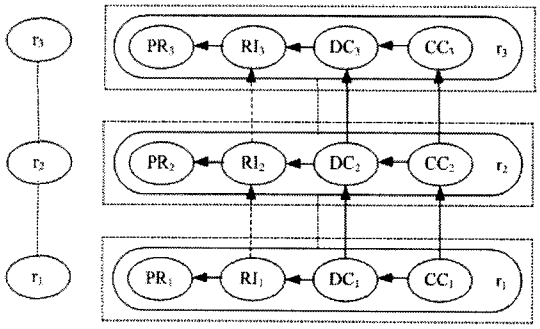


그림 5. 제안 모델의 A-역할계층에서 사용자에게 할당되는 권한 집합

실선화살표는 권한이 상위 역할로 무조건 상속됨을 의미하고, 점선 화살표는 권한이 상위 역할로 상속될 때 제한됨을 의미한다.

각 역할 계층에 대하여 설명하면 다음과 같다. 그림에서 상속 제한 역할(RI1, RI2, RI3)는 모두 바로 위 상위 역할까지만 상속되도록 제한되었다고 가정한다.

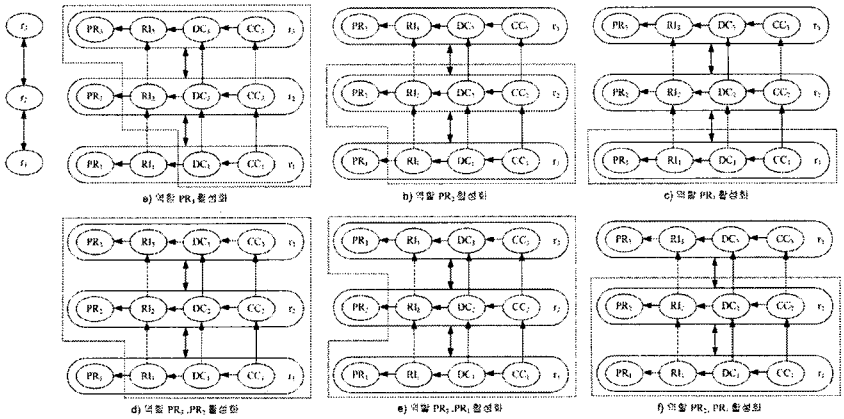


그림 6. 제안 모델의 IA-역할계층에서 사용자에게 할당되는 권한 집합

그림 4는 I-역할계층의 예를 보여준다. 기존 모델은 I-역할계층 관계일 경우 하위 역할에 할당된 모든 권한이 상위 역할로 할당된다. 그러나 제안 모델에서는 사용자에게 할당된 역할(r3 : PR3, RI3, DC3, CC3)의 권한은 모두 사용가능하고 하위에서 상속되는 역할은 제한된다. 그림 4의 점선으로 표시된 영역이 최상위 역할에 할당된 사용자에게 상속되는 역할 집합을 표시 한 것으로 사용자에게 할당되는 역할 집합은 {PR3, RI3, DC3, CC3, RI2, DC2, CC2, DC1, CC1}이다.

그림 5는 A-역할 계층의 예를 보여준다. A-역할 계층에서는 권한은 상속되지 않고 역할의 활성화만 상속되므로 기존 모델과 동일하다. 즉, 활성화 하는 역할의 부역할에 할당된 권한만 사용자가 획득할 수 있다. 그림 5의 점선 부분은 역할 PR3, PR2, PR1을 각각 활성화 하였을 때 획득 가능한 권한 집합을 나타낸다. 즉, A-역할 계층에서는 PR3역할을 활성화 하면 하위 계층의 권한이 상속되지 않고 PR3에 할당된 권한과 부역할의 권한만 획득할 수 있고, PR2 역할을 활성화 하면 PR2에 할당된 권한과 부역할의 권한만 획득 가능하고, PR1을 활성화 하면 PR1에 할당된 권한과 부역할의 권한만 획득가능하다.

그림 6은 IA-역할계층의 예를 보여준다. 기존 모델에서는 사용자가 역할 r1을 활성화 했을 때는 역할 r1에 할당된 권한만 할당되고, 역할 r2를 활성화 하면, 역할 r2의 권한을 할당받고 I-역할 계층에 의해 r2의 하위 역할인 r1의 권한을 상속받는다. 사용자가 역할 r3를 활성화 하면, 역할 r3의 권한을 할당받고 I-역할계층에 의해 r1과 r2의 역할을 모두 할당 받는다. 따라서 A-역할계층에 의해 두개 이상의 역할을 활성화 할 필요가 없었다. 그러나 제안 모델에서는 그림 6의 a), b), c), d), e), f)와 같이 사용자가 활성화 하는 역할에 따라 사용자에게 할당되는 권한의 집합이 달라짐을 알 수 있다. PR3역할을 활성화 하면 하위의 권한 PR2를 제외한 권한 RI2, DC2, CC2를 상속받고, 권한 PR1와 RI1를 제외한 DC1, CC1를 상속받게 된다. 또한 PR2 역할을 활성화 하면 하위의 권한 PR1를 제외한 RI1, DC1, CC1를 상속 받게 된다. 즉 기존 모델에서는 그림 6의 예와 같은 역할 계층에서는 사용자에게 할당되는 역할의 조합이 세가지 경우만 존재하지만 제안모델에서는 A-역할계층에서와 같은 개수의 권한 집합이 얻어진다.

위 표 2는 그림 4, 5, 6의 예에서 각 역할이 세션에서 활성화 될 때 사용자에게 할당되는 역할과 권한이 상속되는 역할의 집합을 보여준다.

위 표 2에서 제안 모델은 I-역할계층에서 권한의 상속이 제한되고, IA-역할계층에서도 권한의 상속이 제한됨을 알 수 있고, A-역할 계층에서는 세션에서 역할을 활성화하고 하위 역할의 권한이 상속되지 않으므로 기존 모델과 제안 모델이 동일함을 알 수 있다.

또한 IA-역할 계층에서 기존 모델에서는 권한의 상속이 제한되지 않으므로 역할 R1과 R2, R2와 R3, R1과 R3, R1과 R2와 R3를 활성화 하지 않아도 R2 또는 R3만을 활성화 하여 동일한 권한이 상속되었으나, 제안 모델에서는 상속 제한 역할과 고유역할로 인하여 A-역할 계층에서와 같은 개수의 역할 집합이 얻어진다.

표 2. 기존 모델과 제안 모델의 역할 계층별 권한 상속 비교

		활성화 되는 역할						
		R1 (PR1)	R2 (PR2)	R3 (PR3)	R1, R2 (PR1, PR2)	R2, R3 (PR2, PR3)	R1, R3 (PR1, PR3)	R1, R2, R3 (PR1, PR2, PR3)
I 역할 계층	기존 모델	활성화 불가능	활성화 불가능	R1, R2, R3	활성화 불가능	활성화 불가능	활성화 불가능	활성화 불가능
	제안 모델	활성화 불가능	활성화 불가능	PR3, RI3, DC3, CC3, RI2, DC2, CC2, DC1, CC1	활성화 불가능	활성화 불가능	활성화 불가능	활성화 불가능
A 역할 계층	기존 모델	R1	R2	R3	R1, R2	R2, R3	R1, R3	R1, R2, R3
	제안 모델	PR1, RI1, DC1, CC1	PR2, RI2, DC2, CC2	PR3, RI3, DC3, CC3	PR1, RI1, DC1, CC1, PR2, RI2, DC2, CC2	PR2, RI2, DC2, CC2, PR3, RI3, DC3, CC3	PR1, RI1, DC1, CC1, PR3, RI3, DC3, CC3	PR1, RI1, DC1, CC1, PR2, RI2, DC2, CC2, PR3, RI3, DC3, CC3
IA 역할 계층	기존 모델	R1	R2, R1	R3, R2, R1	R2, R1	R3, R2, R1	R3, R2, R1	R3, R2, R1
	제안 모델	PR1, RI1, DC1, CC1	PR2, RI2, DC2, CC2, RI1, DC1, CC1	PR3, RI3, DC3, CC3, RI2, DC2, CC2, DC1, CC1	PR2, RI2, DC2, CC2, PR1, RI1, DC1, CC1	PR3, RI3, DC3, CC3, PR2, RI2, DC2, CC2, RI1, DC1, CC1	PR3, RI3, DC3, CC3, RI2, DC2, CC2, PR1, RI1, DC1, CC1	PR1, RI1, DC1, CC1, PR2, RI2, DC2, CC2, PR3, RI3, DC3, CC3

3.3 임시 역할 계층의 정형적 명세

제안 모델을 정형적으로 표현하면 다음의 정의[1~9]와 같고 정의에서 사용되는 기호들은 다음의 정리와 같다.

정리 : 모든 r, u, p, s는 시간 상수 $t \geq 0$ 일 때 다음과 같은 의미를 가진다. 이때 r(역할), u(사용자), p(권한), s(세션), r_{PR}(고유 역할), r_{RI}(상속 제한 역할), r_{DC}(부서 공통 역할), r_{CC}(조직 공통 역할), p_{PR}(고유 역할에 할당된 권한), p_{RI}(상속 제한 역할에 할당된 권한), p_{DC}(부서 공통 역할에 할당된 권한), p_{CC}(조직 공통 역할에 할당된 권한) 이다.

1. assigned(p, r_{PR}, t) → can_be_acquired({p_{PR}, p_{RI}, p_{DC}, p_{CC}}, {r_{PR}, r_{RI}, r_{DC}, r_{CC}}, t)
2. assigned(u, r_{PR}, t) → can_activate(u, r_{PR}, t)
3. can_activate(u, r_{PR}, t) ∧ can_be_acquired ({p_{PR}, p_{RI}, p_{DC}, p_{CC}}, {r_{PR}, r_{RI}, r_{DC}, r_{CC}}, t) → can_acquire(u, {p_{PR}, p_{RI}, p_{DC}, p_{CC}}, t)
4. active(u, r_{PR}, s, t) ∧ can_be_acquired({p_{PR}, p_{RI}, p_{DC}, p_{CC}}, {r_{PR}, r_{RI}, r_{DC}, r_{CC}}, t) → acquires(u, {p_{PR}, p_{RI}, p_{DC}, p_{CC}}, s, t)

각 함수의 의미는 다음과 같다.

-. assigned() : 사용자 또는 권한에 역할의 할당
-. can_be_acquired() : 권한의 획득 가능
-. can_activate() : 역할 활성화 가능
-. active() : 세션 상에서 역할의 활성화
-. acquires() : 세션 상에서 권한의 획득

[정의 1] Unrestricted inheritance only hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때($x \geq^t y$) I-역할 계층은 다음과 같은 의미를 가진다.

- y_{RI}의 권한 상속 범위가 역할 x 또는 x 보다 상위 역할로 지정되었을 경우 :

$\forall p, (x \geq^t y) \wedge$ can_be_acquired({p_{RI}, p_{DC}, p_{CC}}, {y_{RI}, y_{DC}, y_{CC}}, t) → can_be_acquired(p, x_{PR}, t)

- y_{RI}의 권한 상속 범위가 역할 x 보다 하위 역할로 지정되었을 경우 :

$\forall p, (x \geq^t y) \wedge$ can_be_acquired({p_{DC}, p_{CC}}, {y_{DC}, y_{CC}}, t) → can_be_acquired(p, x_{PR}, t)

[정의 2] Activation hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때($x \geqslant^t y$) A-역할 계층은 다

음과 같은 의미를 가진다.

$\forall u, (x \geqslant^{t} y) \wedge can\_activate(u, x_{PR}, t) \rightarrow can\_activate(u, y_{PR}, t)$

[정의 3] General inheritance hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geq^{t} y)$ IA-역할 계층은 다음과 같은 의미를 가진다.

$(x \geq^{t} y) \leftrightarrow (x \geq^{t} y) \wedge (x \geqslant^{t} y)$

[정의 4] Weakly restricted inheritance only hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geq^{w,t} y)$ I-역할 계층은 다음과 같은 의미를 가진다.

- y_{RI}의 권한 상속 범위가 역할 x 또는 x 보다 상위 역할로 지정되었을 경우 :

$\forall p, (x \geq^{w,t} y) \wedge enabled(x_{PR}, t) \wedge can\_be\_acquired(\{p_{RI}, p_{DC}, p_{CC}\}, \{y_{RI}, y_{DC}, y_{CC}\}, t) \rightarrow can\_be\_acquired(p, x_{PR}, t)$

- y_{RI}의 권한 상속 범위가 역할 x 보다 하위 역할로 지정되었을 경우 :

$\forall p, (x \geq^{w,t} y) \wedge enabled(x_{PR}, t) \wedge can\_be\_acquired(\{p_{DC}, p_{CC}\}, \{y_{DC}, y_{CC}\}, t) \rightarrow can\_be\_acquired(p, x_{PR}, t)$

[정의 5] Weakly restricted activation hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geqslant^{t} y)$ A-역할 계층은 다음과 같은 의미를 가진다.

$\forall u, (x \geqslant^{w,t} y) \wedge enabled(y_{PR}, t) \wedge can\_activate(u, x_{PR}, t) \rightarrow can\_activate(u, y_{PR}, t)$

[정의 6] Weakly restricted general inheritance hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geq^{w,t} y)$ IA-역할 계층은 다음과 같은 의미를 가진다.

$(x \geq^{w,t} y) \rightarrow (x \geq^{w,t} y) \wedge (x \geqslant^{w,t} y)$

[정의 7] Strongly restricted inheritance only hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geq^{s,t} y)$ I-역할 계층은 다음과 같은 의미를 가진다.

- y_{RI}의 권한 상속 범위가 역할 x 또는 x 보다 상위 역할로 지정되었을 경우 :

$\forall p, (x \geq^{s,t} y) \wedge enabled(y_{PR}, t) \wedge enabled(x_{PR}, t) \wedge can\_be\_acquired(\{p_{RI}, p_{DC}, p_{CC}\}, \{y_{RI}, y_{DC}, y_{CC}\}, t) \rightarrow can\_be\_acquired(p, x_{PR}, t)$

- y_{RI}의 권한 상속 범위가 역할 x 보다 하위 역할로 지정되었을 경우 :

$\forall p, (x \geq^{s,t} y) \wedge enabled(y_{PR}, t) \wedge enabled(x_{PR}, t) \wedge can\_be\_acquired(\{p_{DC}, p_{CC}\}, \{y_{DC}, y_{CC}\}, t) \rightarrow can\_be\_acquired(p, x_{PR}, t)$

[정의 8] Strongly restricted activation hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geqslant^{s,t} y)$ A-역할 계층은 다음과 같은 의미를 가진다.

$\forall u, (x \geqslant^{s,t} y) \wedge enabled(x_{PR}, t) \wedge enabled(y_{PR}, t) \wedge can\_activate(u, x_{PR}, t) \rightarrow can\_activate(u, y_{PR}, t)$

[정의 9] Strongly restricted general inheritance hierarchy : 시간 t에 역할 x가 역할 y의 상위 역할 일 때$(x \geq^{s,t} y)$ IA-역할 계층은 다음과 같은 의미를 가진다.

$(x \geq^{s,t} y) \rightarrow (x \geq^{s,t} y) \wedge (x \geqslant^{s,t} y)$

단, 위 [정의 1, 2, 3]은 시간 제약을 이용하여 권한의 상속과 역할의 활성화를 제한하지 않는 경우이다.

3.4 Extended GTRBAC 모델의 UAS

GTRBAC 모델은 역할 계층에 I 역할계층, A 역할계층, I-A 역할계층과 같은 세가지 역할 계층이 혼재한다. GTRBAC 모델에서는 역할 계층에서 최소 권한 원칙에 위배되지 않도록 하기 위하여 활성화 가능한 역할 집합을 계산 하여야 하는데 특히 I 역할계층, A 역할계층, I-A 역할계층이 혼합되어있을 경우 활성화 가능한 역할의 집합을 계산하는 것은 매우 복잡하다. 이러한 여러 역할 계층이 혼합된 역할계층에서 사용자에게 할당된 역할에 의해 활성화 될 수 있는 역할의 집합을 UAS(uniquely activable set)이라 한다.

UAS는 단일 세션에서 사용자에 의해 활성화 될 수 있는 역할들의 집합을 말하고 역할 계층을 통하여 사용자에 의해 활성화 될 수 있는 역할집합을 결정할 수 있도록 해줌으로써 최소 권한 원칙을 유지할 수 있도록 도와준다.

Extended GTRBAC 모델은 그림 4, 5, 6과 표 2에서 알 수 있듯이 IA-역할계층에서 기존 모델과는 다르게 상속 제한 역할과 고유역할로 인하여 A-역할 계층에서와 같은 개수의 역할 집합이 얻어 진다. 따

라서 활성화 가능한 역할의 집합인 UAS도 기존 모델과 다르고 UAS의 계산 방법 또한 재정의 되어야 한다.

[정리 1] 역할 집합 X = {x1, x2, ..., xn}과 역할 계층 관계 <f> ∈ {$\geq^t$, $\gg^t$, $\succeq^t$}에서 단일 역할 계층을 H = (X, <f>)라 하면,

- if (<f> = $\geq^t$) then UAS(H, t) = S_H
- if (<f> = $\gg^t$) then UAS(H, t) = $2^x/\varnothing$
- if (<f> = $\succeq^t$) then UAS(H, t) = $2^x/\varnothing$

[정리 2] L_1 = (X_1, <f_1>), LH_2는 역할 X_2 상위의 역할 계층 경로일 때 Lh = {L_1, LH_2}라 하면

- if <f_1> = $\geq^t$ then UAS(Lh, t) = UAS(L_1, t)
- if <f_1> = $\gg^t$ then UAS(Lh, t) = UAS(L_1, t) : if (<f_x> = $\geq^t$)

UAS(L_{1U}, t) ∪ UAS(LH_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(LH_2, t)) : if (<f_x> = $\succeq^t$)

- if <f_1> = $\succeq^t$ then UAS(Lh, t) = UAS(L_1, t) : if (<f_x> = $\geq^t$)

UAS(L_{1U}, t) ∪ UAS(LH_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(LH_2, t)) : if (<f_x> = $\gg^t$)

[정리 3] ∃x, y, z ∈ X, (x <f> y) ∧ (x <f> z) 이고, 역할 계층 H = (X, [f]) = {LH_1, H_1} 이면 UAS(H, t) = I 이다.

- I = (UAS(LH_1, t) ∪ UAS(H_1, t) ∪ (UAS(LH_1, t)/B ⊗ UAS(H_1, t) / B)),
- B = (UAS(LH_1, t) ∩ UAS(H_1, t)) = {X, Y | X ∈ UAS(LH_1, t), Y ∈ UAS(H_1, t) and X ∩ Y ≠ ∅}

4. Extended GTRBAC 모델의 적용 예

4. 1 health care 시스템의 Extended GTRBAC 모델 적용 예

이번 절에서는 health care 시스템에서 역할 계층이 다음 그림 7과 같다고 할 때 Extended GTRBAC 모델이 어떻게 적용되는지를 보여준다.

다음 그림 8은 위 그림 7의 health care 시스템 역할 계층인 혼합 선형 역할 계층의 UAS를 계산하기 위하여 역할 계층을 분할한 예이다.

그림 8의 각 선형 역할 계층을 이용하여 아래의 단계와 같이 UAS를 계산 할 수 있다. 아래의 단계에서 H_{12}는 L_1과 Lh_2, H_{13}은 L_1, Lh_2, Lh_3을, H_{14}는 L_1, Lh_2, Lh_3, Lh_4를 H_{15}는 L_1, Lh_2, Lh_3, Lh_4, Lh_5를 H_{16}은 L_1, Lh_2, Lh_3, Lh_4, Lh_5, Lh_6의 역할 계층을 고려한 UAS 이다.

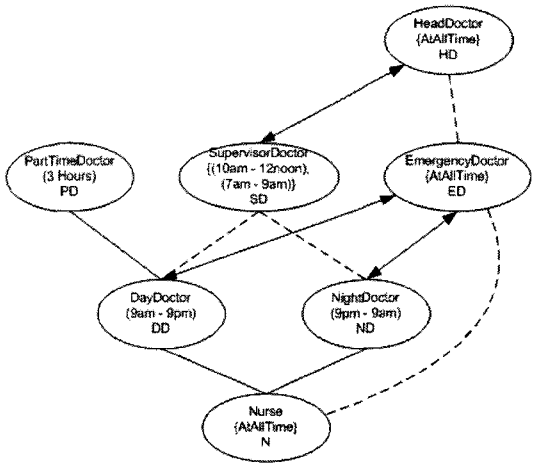


그림 7. health care 시스템의 역할 계층 예

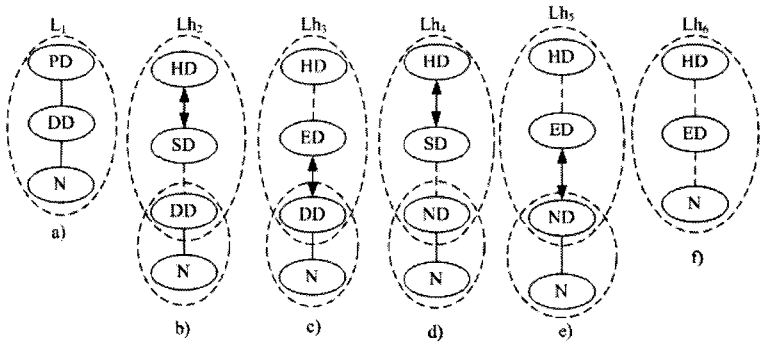


그림 8. health care 시스템의 혼합 선형 역할 계층을 위한 UAS 계산 예

그림 8을 이용한 단계별 UAS 계산.

■ 단계 1 : L_1, UAS(L_1, t) = *{(PD)}*

■ 단계 2 : Lh_2, UAS(Lh, t) = UAS(L_{1U}, t) ∪ UAS(Lh_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(Lh_2, t)) = *{(HD }, {SD }, {DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, SD, DD)}*

■ 단계 3 : Lh_3, UAS(Lh, t) = UAS(L_{1U}, t) ∪ UAS(Lh_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(Lh_2, t)) = *{(HD}, {ED}, {DD}, {HD, ED}, {HD, DD}, {ED, DD}, {HD, ED, DD)}*

■ 단계 4 : Lh_4, UAS(Lh, t) = UAS(L_{1U}, t) ∪ UAS(Lh_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(Lh_2, t)) = *{(HD }, {SD}, {ND}, {HD, ND}, {SD, ND}, {HD, SD}, {HD, SD, ND)}*

■ 단계 5 : Lh_5, UAS(Lh, t) = UAS(L_{1U}, t) ∪ UAS(Lh_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(Lh_2, t)) = *{(HD}, {ED}, {ND}, {HD, ED}, {HD, ND}, {ED, ND}, {HD, ED, ND)}*

■ 단계 6 : Lh_6, UAS(Lh, t) = UAS(L_{1U}, t) ∪ UAS(Lh_2, t) ∪ (UAS(L_{1U}, t) ⊗ UAS(Lh_2, t)) = *{(HD}, {ED}, {N}, {HD, ED}, {HD, N}, {ED, N}, {HD, ED, N)}*

■ 단계 7 : L_1과 Lh_2, H_{12}를 계산하면,

B = (UAS(Lh_2, t) ⊓ UAS(L_1, t)) = empty.

(UAS(L_1,t) / B) ⊗ (UAS(Lh_2, t) / B) = *{(PD)}* ⊗ *{(HD }, {SD }, {DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, SD, DD)}* = *{(PD, HD }, {PD, SD }, {PD, DD }, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD)}*.

UAS(H_{12}, t) = I = (UAS(Lh_2, t) ∪ UAS(L_1, t) ∪ (UAS(Lh_2, t)/B ⊗ UAS(L_1, t) / B)) = *{(PD}, {HD}, {SD}, {DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, SD, DD}. {PD, HD}, {PD, SD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD)}*

■ 단계 8 : H_{12}와 Lh_3, H_{13}을 계산하면,

B = (UAS(Lh_3, t) ⊓ UAS(H_{12}, t)) = *.{(HD}, {DD}, {HD, ED}, {HD, DD}, {ED, DD}, {HD, ED, DD}, {SD, DD}, {HD, SD}, {HD, SD, DD}, {PD, HD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD)}*

(UAS(H_{12}, t) / B) ⊗ (UAS(Lh_3, t) / B) = *{(PD}, {SD}, {PD, SD)}* ⊗ *{(ED)}* = *{(PD, ED}, {SD, ED}, {PD, SD, ED)}*

UAS(H_{13}, t) = I = (UAS(Lh_3, t) ∪ UAS(H_{12}, t) ∪ (UAS(Lh_3, t)/B ⊗ UAS(H_{12}, t) / B)) = *{(PD}, {HD}, {SD}, {DD}, {ED}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}, {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, SD, DD}, {PD, HD}, {PD, SD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD)}*

■ 단계 9 : H_{13} 과 Lh_4, H_{14}를 계산하면,

B = (UAS(Lh_4, t) ⊓ UAS(H_{13}, t)) = *.{(HD}, {SD}, {HD, ED}, {SD, ED}, {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, ND}, {SD, ND}, {HD, SD, DD}, {HD, SD, ND}, {PD, HD}, {PD, SD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD)}*

(UAS(H_{13}, t) / B) ⊗ (UAS(Lh_4, t) / B) = *{(PD}, {DD}, {ED}, {ED, DD}, {PD, ED}, {PD, SD, ED}, {HD, ED, DD}, {PD, DD)}* ⊗ *{(ND)}* = *{(PD, ND}, {DD, ND}, {ED, ND}, {ED, DD, ND}, {PD, ED, ND}, {PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, DD, ND)}*

UAS(H_{14}, t) = I = (UAS(Lh_4, t) ∪ UAS(H_{13}, t) ∪ (UAS(Lh_4, t)/B ⊗ UAS(H_{13}, t) / B)) = *{(PD}, {HD}, {SD}, {DD}, {ED}, {ND}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}. {PD, ND}, {DD, ND}, {ED, ND}, {ED, DD, ND}, {PD, ED, ND}; {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, ND}, {SD, ND}, {HD, SD, ND}, {HD, SD, DD}, {PD, HD}, {PD, SD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD}{PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, DD, ND)}*

■ 단계 10 : H_{14}와 Lh_5, H_{15}를 계산하면,

B = (UAS(Lh_5, t) ⊓ UAS(H_{14}, t)) = *{(HD}, {ED}, {ND}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}, {PD, ND}, {DD, ND}, {ED, ND}, {ED, DD, ND}, {PD, ED, ND}, {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {HD, SD}, {HD, ND}, {SD, ND}, {HD, SD, ND}, {HD, SD, DD}, {PD, HD}, {PD, HD, DD}, {PD, HD, SD}, {PD, HD, SD, DD}, {PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, DD, ND)}*

(UAS(H_{14}, t) / B) ⊗ (UAS(Lh_5, t) / B) = *{(PD}, {SD}, {DD}, {SD, DD}, {PD, SD}, {PD, DD}, {PD, SD, DD)}* ⊗ *{(HD, ED, ND)}* = *{(PD, HD, ED, ND}, {SD, HD, ED, ND}, {DD, HD, ED, ND}, {SD, DD, HD, ED, ND}, {PD, SD, HD, ED, ND}, {PD, DD, HD, ED, ND}, {PD, SD, DD, HD, ED, ND)}*

UAS(H_{15}, t) = I = (UAS(Lh_5, t) ∪ UAS(H_{14}, t) ∪ (UAS(Lh_5, t)/B ⊗ UAS(H_{14}, t) / B)) = *{(PD}, {HD}, {SD}, {DD}, {ED}, {ND}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}, {PD, ND}, {DD, ND}, {ED, ND}, {ED, DD, ND}, {HD, ED, ND}, {PD, ED, ND}, {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, ND}, {SD, ND}, {HD, SD, ND}, {HD, SD, DD}, {PD, HD}, {PD, SD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD}, {PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, DD, ND}, {PD, HD, ED, ND}, {SD, HD, ED, ND}, {SD, DD, HD, ED, ND}, {PD, SD, HD, ED, ND}, {PD, DD, HD, ED, ND}, {PD, SD, DD, HD, ED, ND)}*

■ 단계 11 : H_{15}와 Lh_6, H_{16}를 계산하면,

B = (UAS(Lh_6, t) ⊓ UAS(H_{15}, t)) = *{(HD}, {ED}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}, {ED, ND}, {HD, N}, {ED, N}, {HD, ED, N}, {ED, DD, ND}, {HD, ED, ND}, {PD, ED, ND}, {PD, SD, ED}, {HD, ED, DD}, {HD, DD}, {HD, SD}, {HD, ND}, {HD, SD, ND}, {HD, SD, DD}, {PD, HD}, {PD, HD, DD}, {PD, HD, SD}, {PD, HD, SD, DD}, {PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, HD, ED, ND}, {SD, HD, ED, ND}, {SD, DD, HD, ED, ND}, {PD, SD, HD, ED, ND}, {PD, DD, HD, ED, ND}, {PD, SD, DD, HD, ED, ND)}*

(UAS(H_{15}, t) / B) ⊗ (UAS(Lh_6, t) / B) = *{(PD}, {SD}, {DD}, {ND}, {PD, ND}, {DD, ND}, {PD, SD}, {SD, DD}, {SD, ND}, {PD, SD}, {PD, DD}, {PD, SD, DD}, {PD, DD, ND)}* ⊗ *{(N)}* = *{(PD, N}, {SD, N}, {DD, N}, {ND, N}, {PD, ND, N}, {DD, ND, N}, {PD, SD, N}, {SD, DD, N}, {SD, ND, N}, {PD, SD, N}, {PD, DD, N}, {PD, SD, DD, N}, {PD, DD, ND, N)}*

UAS(H_{16}, t) = I = (UAS(Lh_6, t) ∪ UAS(H_{15}, t) ∪ (UAS(Lh_6, t)/B ⊗ UAS(H_{15}, t) / B)) = *{(N}, {PD}, {HD}, {SD}, {DD}, {ED}, {ND}, {HD, ED}, {ED, DD}, {PD, ED}, {SD, ED}, {PD, ND}, {DD, ND}, {ED, ND}, {HD, N}, {ED, N}, {PD, N}, {SD, N}, {DD, N}, {ND, N}, {HD, ED, N}, {PD, ND. N}, {DD, ND, N}, {PD, SD, N}, {SD, DD, N}, {SD, ND, N}, {PD, SD, N}, {PD, DD, N}, {PD, SD, DD, N}, {PD, DD, ND, N}, {ED, DD, ND}, {HD, ED, ND}, {PD, ED, ND}, {PD, SD, ED)}, {HD, ED, DD}, {HD, DD}, {SD, DD}, {HD, SD}, {HD, ND}, {SD, ND}, {HD, SD, ND}, {HD, SD, DD}, {PD, HD}, {PD, SD}, {PD, DD}, {PD, HD, DD}, {PD, SD, DD}, {PD, HD, SD}, {PD, HD, SD, DD}, {PD, SD, ED, ND}, {HD, ED, DD, ND}, {PD, DD, ND}, {PD, HD, ED, ND}, {SD, HD, ED, ND}, {SD, DD, HD, ED, ND}, {PD, SD, HD, ED, ND}, {PD, DD, HD, ED, ND}, {PD, SD, DD, HD, ED, ND)}*

위 그림 7의 health care 시스템의 역할계층에서 UAS를 계산 하면 위 단계 11의 UAS(H_{16}, t)와 같다.

4.2 활성화되는 역할에 따른 권한 획득 알고리즘

4.1절의 단계 11에서 Extended GTRBAC 모델의 UAS는 기존 모델 보다 더 복잡해짐을 알 수 있다. 따라서 세션에서 활성화 되는 역할에 따라 사용자가 획득할 수 있는 권한의 계산 또한 복잡해지게 되므로, UAS 계산 알고리즘과 계산된 UAS에 따른 권한 획득 알고리즘이 필요하다.

```
Algorithm Compute_Permission_Acquire(H, Active_Role)
Permission_set = {Active_RolePR, Active_RoleRI, Active_RoleDC, Active_RoleCC}
// JuniorRoles_Exist(); 활성화된 역할의 하위 역할이 존재하는지 검색
while (JuniorRoles_Exist(H, JuniorRole)) do begin
    // 역할 계층에서 활성화 역할의 하위 역할이 존재하는 경우 반복
      if ((f = 'I') or (f = 'IA')) //역할계층이 I 또는 IA 인경우
          if (JuniorRole->RI ≤ Active_Role->RI)
    // JuniorRole의 상속 제한 역할 RI가 Active_Role에 상속 가능한 경우.
    // JuniorRole->RI의 상속 범위에 Active_Role가 속하는 경우.
                    Permission_set +={JuniorRole->RI,.
                                      JuniorRole->DC, JuniorRole->CC}
              else Permission_set += {JuniorRole->DC, JuniorRole->CC}
      if (f = 'A') break          //A 역할계층인경우
      Active_Role = JuniorRole
end
return Permission_set
End Compute_Permission_Acquire
```

위 알고리즘은 역할계층에서 사용자가 활성화 하는 역할에 따라 획득할 수 있는 권한 집합을 계산하는 알고리즘이다. 사용자에게 그림 7의 PD(Part Time Doctor) 역할이 할당되었을때 이 알고리즘으로 사용자가 획득할 수 있는 권한을 계산하면 다음의 권한 집합과 같다.

step 1. {PR_{PD}, RI_{PD}, DC_{PD}, CC_{PD}}

step 2. {PR_{PD}, RI_{PD}, DC_{PD}, CC_{PD}, RI_{DD}, DC_{DD}, CC_{DD}}

step 3. {PR_{PD}, RI_{PD}, DC_{PD}, CC_{PD}, RI_{DD}, DC_{DD}, CC_{DD}, DC_{N}, CC_{N}}

5. 결 론

본 논문에서는 시간(기간과 주기)에 따른 제약으로 자원의 사용을 제한할 수 있고, 역할 활성화와 이벤트, 트리거를 이용하여 사용자 수를 제한하고 기업 환경의 워크플로우에 해당하는 작업을 다룰 수 있는 장점을 가지는 GTRBAC(Generalized Temporal Role Based Access Control)모델에 부역할(sub role) 개념을 적용하여 하위 역할에 배정된 권한을 상위 역할에 배정된 사용자가 모두 상속하여 실행할 수 없도록 하여 권한의 남용을 방지하고, 최소권한의 원칙을 지킬 수 있는 확장된 GTRBAC (Extended GTRBAC) 모델을 제안하였다. 그러나 본 논문에서는 권한의 상속 문제만을 다루고, 역할 또는 권한의 위임 문제를 고려하지 않는 단점이 있다.

향후에는 실제 기업환경에서 발생할 수 있는 다양한 권한의 위임 문제를 해결하기 위하여 제안 모델에 위임과 부분 위임을 고려하고, 또한 접근제어 시스템에 복잡하고 다양한 제약을 적용하였을 때 보안 관리자가 정책을 보다 간편하고 효율적으로 관리할 수 있는 형식 언어에 대한 연구가 필요할 것으로 사료된다.

참 고 문 헌

[1] C.P.Pfleeger, Security in Computing, second edition, Prentice-Hall International Inc, 1997.

[2] R.S.Sandhu and E.J.Coyne and H.L.Feinstein and C.E.Youman "Role-Based Access control Method", IEEE Computer, vol. 29, 1996.

[3] D.Ferraiom and J.Cugini and R.Kuhm "Role-based Access Control(RBAC) : Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.

[4] Dagstull and G.Coulouris and J.Dollimore "A Security Model for Cooperative work : a model and its system implications" Positions paper for ACM European SIGOPS Workshop, 1994.

[5] R.K.Thomas and R.S.Sandhu "Task-based Authorization Controls(TBAC) : A Family of Models for Active and Enterprise-oriented Authorization Management" Proc. of the IFIP WF11.3 Workshop on Database Security, 1997.

[6] S. Oh and S. Park "Task-Role Based Access Control (T-RBAC): An Improved Access Control Model for Enterprise Environment",

Proceedings of the 11th International Conference on Database and Expert Systems Applications, pp. 264-273, 2000.

[7] HyungHyo Lee and YoungRok Lee and BongNam Noh "A New Role-Based Delegation Model Using Sub-Role Hierarchies" Proceedings of the 18 th Computer and Information Sciences - ISCIS2003, 2003.

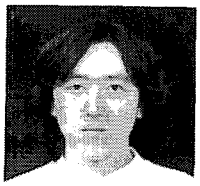
[8] YongHoon Yi and MyongJae Kim and YoungLok Leem and HyungHyo Lee and BongNam Noh "Applying RBAC Providing Restricted Permission Inheritance to a Corporate Web Environment", Proceedings of the 5 th Asia-Pacific Web Conference, 2003.

[9] E. Bertino and P. A. Bonatti and E. Ferrari "TRBAC: A Temporal Role-based Access Control Model", Proceedings of the fifth ACM workshop on Role-based access control, pp.21- 30, 2000.

[10] J. B. D. Joshi and E. Bertino and A. Ghafoor "Temporal Hierarchies and Inheritance Semantics for GTRBAC", Seventh ACM Symposium on Access Control Models and Technologies, pp. 74-83, 2002.

[11] J. B. D. Joshi and E. Bertino and A. Ghafoor "Hybrid Role Hierarchy for Generalized Temporal Role Based Access Control Model", Proceedings of the 26 th Annual International Computer Software and Applications Conference, 2002.

[12] J. B. D. Joshi and E. Bertino and A. Ghafoor "Temporal Role Hierarchies in GTRBAC", CERIAS, 2002.



황 유 동

1998년 순천향대학교 제어계측공학과 공학사
2000년 순천향대학교 전기전자공학과 석사
2003년 순천향대학교 전기전자공학과 정보보호전공 박사과정 수료
관심분야 : 네트워크 보안, 시스템 보안

박 동 규

1992년 한양대학교 대학원 전자공학과 공학박사
1992년~1995년 순천향대학교 정보통신공학과 전임강사
1995년~1998년 순천향대학교 전기전자공학부 조교수
1999년~2003년 순천향대학교 정보기술공학부 부교수
2004년~현재 순천향대학교 정보기술공학부 교수
관심분야 : 네트워크 보안, 시스템 보안