CSCW 환경에서의 혼합형 접근제어 모델

° 김상진*, 고희창*, 김남용*, 왕창종*
인하대학교 전자계산공학과*

Hybrid Access Control Model in CSCW Environment

° S.J. Kim*, H C. Koh*, N.Y Kim*, C.J. Wang*
Dept. of Computer Science & Engineering, Inha University*

요약

CSCW 지원 도구에서 접근제어 정보의 효과적 관리는 매우 중요하다 이는 접근제어가 관리 비용이 많이 들고, 에러가 발생하기 쉬운 경향이 있기 때문이다 접근 제어 모델 중 ACL 은 다양한 시스템에서 사용되고 있으나 많은 문제점을 내포하고 있기에 이를 개선한 RBAC 모델이 대두되고 있다 하지만 다양한 공동작업 환경의 경우, 역할에 속한 사용자들의 집합이 아니라 특별한 개별 사용자 집합에 의한 작업이 요구될 수도 있다 따라서 CSCW 환경에서는 사용자 그룹기반의 접근제어 모델과 역할기반의 접근제어 모델을 혼합한 형태의 새로운 접근제어 모델이 필요하다

본 연구에서는 이러한 요구사항을 만족시키기 위해서 역할기반 접근제어 모델과 사용자 기반 접근제어 모델을 혼합한 형태의 접근제어 모델을 제안한다

1. 서론

다수의 참여자가 참가하는 작업의 경우, 참여자 간의 의사소통과 정보의 교환 및 공유가 원활히 이루어지는 것이 중요하며, 이러한 공동작업을 지원하는 컴퓨터 기술(CSCW Computer Supported Cooperative Work)의 보다 효율적인 지원이 요구된다[1]. 이러한 컴퓨팅 지원 도구의 성장에 있어 가장 큰 장애물 중의 하나가 접근제어 정보를 효과적으로 관리할 수 있는 능력이 없다는 것이다 일반적으로 접근제어는 관리 비용이 많이 들고, 에러가 발생하기 쉬운 경향이 있다[2]. 보통 보안 관리자는 접근제어 목록(ACL, Access Control Lists)의 생성과 유지를 통하여 공유 문서에 대한 사용자의 접근을 제어한다. ACL 은 각각의 제한 자원에 대하여 지정된 사용자들 또는 개별 사용자들로 구성된 그룹들의 목록을 서술한다. 이러한 ACL 의 사용은 다양한 이유로 인해서 많은 문제점이 있다[3] 근래에는 ACL 의 문제점을 개선하는 역할기반 접근제어(RBAC, Role Based Access Control) 모델이 많은 운영체제와 데이터베이스 관리 시스템에 사용되고 있다 RBAC 모델에서 인가권한(permission)은 개별 사용자보다는 역할에 의해서 분할되고 서술된다 따라서, RBAC 는 조직 또는 기관에 적합한 개념이고, 조직적인 관점에서 보안을 설계할 수 있게 하는 장점을 지닌다 또한 RBAC 는 역할에 정의된 보안 속성이 역할에 속해 있는 사용자 전체에 해당되기 때문에 사용자 기반 보안 기술보다는 규모 유동성이 크다. 이것은 접근제어 관리 비용과 관리상의 과부하를 감소시킨다. [3,4]

본 연구는 역할기반 접근제어 모델을 기반으로 다양한 공동작업 환경에 맞는 모델을 제안한다. 공동 작업 환경에서는 역할에 속한 사용자들의 집합이 아니라 특별한 개별 사용자 집합에 의한 작업이 요구될 수 있기 때문에 사용자 그룹기반의 접근제어 모델과 역할기반의 접근제어 모델을 혼합한 형태의 새로운 접근제어 모델이 필요하다 본 연구에서는 이러한 요구사항을 만족시키기 위해서 역할기반 접근제어 모델과 사용자 기반 접근제어 모델을 혼합한 형태의 접근제어 모델을 제안한다

2. 기존 접근 제어 모델

본 장에서는 ACL 와 RBAC 모델을 비교 분석하고, 공동작업 환경에서 필요로 하는 접근제어 모델의 일반적인 요구사항에 대하여 분석한다.

2.1 ACL 와 RBAC 의 비교

ACL 은 접근 제어 모델 중 가장 간단하면서도 많이 사용하고 있는 모델이다. ACL 은 객체, 인가권한의 집합, 주체로 구성된다

ACL = {S, P}, S Subject, P: Permission sets

ACL 은 접근 제어 행렬을 열을 기준으로 하여 구현하는 방법으로서 각각의 관리 객체는 자신과 관련된 접근제어 리스트를 하나씩 갖는다. ACL 은 객체들이 세분화되어 있는 경우는 유리하며, 주체의 수가 적으며 고정적인 환경에 적합하다. 그러나 특정 관리자가 접근할 수 있는 모든 관리 객체들을 검색해야 하는 문제점이 있기 때문에 보다 복잡한 응용에서는 역할 기반의 접근 제어 모델인 RBAC 가 제시되었다.[]

RBAC 는 해당 객체에 대한 인가권한의 집합을 역할에 할당한다 역할은 접근제어에서 사용자 그룹의 개념과 매우 유사하나 사용자 그룹은 전형적으로 사용자들의 집합으로 정의되지만, 역할은 사용자들의 집합과 권한들의 집합을 함께 다룬다. RBAC 는 다음과 같은 요소들로 구성된다.

1. 사용자, 역할, 연산, 세션

2. 역할/연산 연관성(N : M 의 관계)

3. 사용자/역할 연관성(N M 의 관계)

4 사용자/세션 연관성(1 : N 의 관계)

RBAC 모델은 전형적으로 구현되는 환경에 독립적이다. 예를 들면, RBAC 는 운영체제나 데이터베이스 시스템에 삽입될 수 있고, 또는 응용 레벨에서도 구현될 수 있다. 그림 1 은 RBAC 의 구성요소들간의 관계를 보여준다. 그림에서 user1 부터 user 1 까지는 role 1 에 포함된다 따라서 role 1 에 인가권한의 집합을 할당하면, role 1 에 포함된 모든 사용자들은 이 인가권한을 갖게 되다 또한 Role 1 이 접근할 수 있는 객체는 Object 1 과 Object 2 이다 따라서 role 1 에 포함된 사용자들은 즉 user 1 부터 n 까지는 Object1 과 Object 2 를 접근할 수 있는 권한을 가진다.

RBAC 은 ACL 모델보다 관리의 복잡성을 줄일 수 있고, 에러의 발생 가능성을 최소화할 수 있다

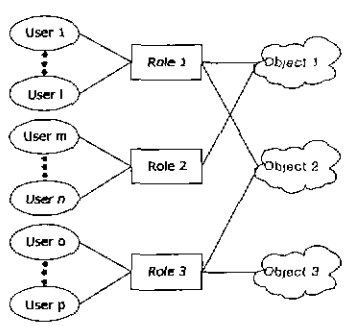

그림 1. RBAC 메커니즘

2.2 CSCW 환경에서 접근제어 모델의 요구사항

기존의 시스템들에서는 공유 자원에 대해 공동작업에 참여하는 모든 사용자들에게 동일한 권한을 부여하거나 특정 관리자에게 추가적인 권한을 허용하는 접근 제어 방식을 사용하고 있다. 따라서, 다수의 사용자가 참여하는 대규모의 공동작업과 여러 계층의 그룹을 관리하는 시스템에 있어서는 적합하지 않다 CSCW 시스템에서의 접근제어 모델의 요구사항은 일반적으로 다음과 같다[5]

- 역할 기반으로 그룹의 접근 권한을 제어함으로써 접근 제어의 복잡성을 감소시켜야 한다.
- 참여자의 역할은 다양하고, 동적으로 변경 가능해야 한다.
- 공유 자원에 대한 읽기, 쓰기 권한 외에 공동작업을 위한 권한이 필요하다.
- 예외적인 경우를 명세하는 부정적(negative) 권한 제이 기능이 필요하다
- 신뢰성을 높이기 위하여 조건적인 제약을 두는 방식이 필요하다.
- 해당 그룹의 접근 권한 정보를 저장하고 임의의 주체가 임의의 객체에 대한 접근이 가능한지 결정을 쉽게 내릴 수 있어야 한다.

또한 다양한 공동작업 환경의 경우, 역할에 속한 사용자들의 집합이 아니라 특별한 개별 사용자 집합에 의한 작업이 요구될 수도 있다 따라서 CSCW 환경에서는 사용자 그룹기반의 접근제어 모델과 역할기반의 접근제어 모델을 혼합한 형태의 새로운 접근제어 모델이 필요하다

3. 혼합형 접근제어 모델

3.1 모델 설계시 고려 사항

공동작업 환경을 고려할 때, 접근제어 모델에는 대한 두 가지 중요한 요구사항이 있다.

- N1. 역할기반의 권한 할당

개별 사용자가 아닌 역할 단위로 접근을 제어함으로써 사용자의 역할 변화에 따른 접근 권한의 관리를 쉽게 할 수 있는 장점을 가질 수 있다.

- N2 :개별 사용자와 객체의 수준에서 실행시간 권한 활성화

하지만 일반적으로 역할기반 접근제어 모델은 위의 요구사항 들을 동시에 수행할 수는 없다. 요구사항 N1 을 수행하면, 요구사항 N2 를 만족시키기 위한 유연성을 잃게 되고, 요구사항 N2 를 수행하게 되면, 역할기반의 권한제어가 필요없게 되며, 따라서 규모성과 접근제어 관리의 장점을 잃게 된다

위의 두 가지 요구사항을 모두 만족시키기 위해서는 다음과 같은 정의가 필요하다

- 사용자들의 집합을 모델링하기 위한 추상 개념와 사용자가 속해 있는 역할
- 사용자들의 집합을 위한 공동작업 문맥(collaboration context)

3.2 혼합형 접근제어 모델 제안

RBAC 에서 모델에 의해 인지되는 유일한 사용자들의 그룹은 같은 역할에 속하는 그룹이다. 이러한 제한으로부터 다양한 역할로부터의 사용자 집합을 모델링하기 위해서 작업 그룹 개념과 공동 작업 문맥등의 개념을 도입한다.

그림 2 는 제안된 접근 제어 모델에 대한 개념도이다.

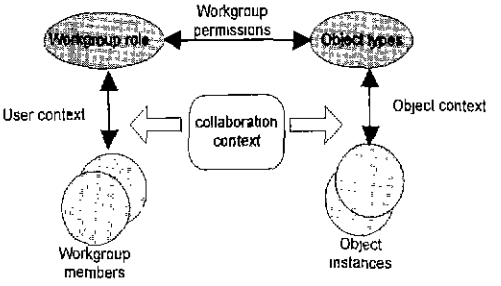

그림 2 제안된 모델의 개념도

(1) 작업그룹(Taskgroup)

작업그룹은 다음과 같은 것들로 구성된다

• name, t

작업그룹의 이름

• A set of members/users, TU

작업그룹 안의 멤버들/사용자들의 집합

• A set of roles, TR ⊆ R(all roles)

작업그룹 역할의 집합, 작업그룹의 역할은 전체 역할에 포함된다

• A special role called taskgroup_manager(m). m ⊂ TR

작업그룹의 관리자를 가르키는 특별한 역할, 관리자 역할은 작업그룹 역할의 집합에 포함된다.

• A set of object types. OT

객체 타입들의 집합

• A set of object instances, O

객체 인스턴스들의 집합

• A set of permissions, TP ⊆ TR × OT

객체의 인가권한들의 집합, 작업그룹의 인가권한은 직업그룹 역할의 집합과 객체 타입 집합의 곱집합 내에 포함된다.

(2) 공동작업 문맥

공동 작업 문맥 (collaboration context) 은 두 가지 정보를 포함하고 있다

• 사용자 문맥. 예를 들면 주어진 시간에 그룹을 구성하는 특별한 사용자들

user context(UC), UC : TR × TU

사용자 문맥은 작업그룹 역할과 작업그룹 멤버의 곱집합이다.

• 객체 문맥. 예를 들면 특정 작업을 수행하기 위해 그룹에 의해 요구되는 객체들의 집합

object context(OC), OC : OT × O

객체 문맥은 객체 타입 집합과 객체 인스턴스 집합의 곱집합이다.

(3) 작업그룹에 접근제어 함수

접근 제어를 위해 다음과 같은 기본 함수를 제공한다

• User_assign(user, taskgroup); //사용자에게 작업그룹을 할당

• User_deassign(user, taskgroup), //사용자에게서 작업그룹 제거

• TaskGroup_activate(taskgroup),

// 작업그룹 인가권한을 작업그룹의 멤버과 객체로 바인딩

• TaskGroup_deactivate(taskgroup);

// 전체 작업그룹으로부터 인가권한 비활성화

4. 결론

본 연구에서는 공동작업 환경에서의 사용자 그룹과 역할을 혼합한 형태의 접근제어 모델을 제안하였다.

제안된 모델은 RBAC 모델의 관리적 측면에서 모델링의 장점을 도입하였고, 사용자 그룹의 인가권한 할당을 문맥 기반 실행시간 인가권한을 줄 수 있으므로 개별적 권한 부여가 가능하였다.

참고 문헌

[1] S M. Easterbrook, *CSCW Cooperation or Conflict?*, Springer-Verlag,1993

[2] R K. Thomas, Team-based Access Control(TMAC), In Proceedings RBAC'97, 1997.

[3] K. Sikkel, A Group-based Authorization Model for Cooperative Systems, In Proceedings ECSCW'97, 1997.

[4] J. Barkley, A.Cincotta, Role Based Access Control for the World Wide Web, In 20th National Information System Security Conference NIST/NSA, 1997

[5] HongHai Shen and Prasun Dewan, "Access Control for Collaborative Environments", In Proceedings of CSCW'92, 1992,