망관리 정보베이스 접근 제어 시스템

김 종 덕† · 이 형 효†† · 노 봉 남†††

요 약

망관리 시스템의 여러 가지 구성 요소들 중 가장 핵심적인 요소 중의 하나는 망관리에 필요한 정보들인 관리 객체들의 개념적인 저장소인 관리 정보베이스이다. 관리 정보베이스에 저장된 관리 객체들은 망관리에 필수적이며 중요한 모든 정보들을 유지하고 있기 때문에 안전하게 유지되어야 한다. 본 논문에서는 접근 제어를 위한 포괄적인 클래스 정의 및 접근 제어 보안 모델을 정의한 ISO/IEC 10164-9 권고안을 바탕으로 기존의 표준 관리 객체 클래스 구조를 명시적 규칙과 묵시적 규칙으로 세분화함으로써 크게 확장 및 보완하였다. 또한 세분화된 접근 제어 규칙에 따라 해당 규칙이 적용되는 절차를 각 접근 제어 정책에 적용하여 봄으로써 접근 제어 규칙 수행의 타당성을 검증하였으며, 접근 제어 시스템의 각 기능과 권고안 및 확장된 모델에 정의된 GDMO의 비정형적인 구조를 명세언어 Z를 이용해 정형화된 구조로 표현하였다.

The Access Control System of Network Management Information Base

Jong-Duk Kim† · Hyung-Hyo Lee†† · Bong-Nam Noh†††

ABSTRACT

MIB(Management Information Base), one of the key components of network management system, is a conceptual repository for the information of the various managed objects. MIB stores and manages all the structural and operational data of each managed resources. Therefore, MIB should be protected properly from inadvertant user access or malicious attacks. International standard ISO/IEC 10164-9 describes several managed object classes for the enforcement of MIB security. Those managed object classes described access control rules for security policy. But the exact authorization procedures using those newly added managed object classes are not presented. In this paper, we divide managed object classes into two groups, explicit and implicit ones, and describe the access authorization procedure in Z specification language. Using Z as a description method for both authorization procedure and GDMO's action part, the behaviour of each managed object class and access authorization procedure is more precisely and formally defined than those of natural language form.

1. 서 론

망관리 시스템의 여러 가지 구성 요소들 중 가장 핵심적인 요소 중의 하나는 망관리에 필요한 정보 즉, 관리 객체들의 개념적 저장소인 관리 정보베이스이다. 관리 정보베이스에 저장된 관리 객체들은 망관리에 필수적이며 중요한 모든 정보들을 유지하고 있기 때문에 안전하게 유지가 되어야 한다.이러한 망관리 정보 베이스가 안전하게 운용되기 위해서는 망 사용자에 대한 정확한 인증 뿐만 아니라, 관리 객체에 대한 접근을 효율

※ 본 연구는 한국과학재단 핵심전문 연구(과제번호:951-0100-001-2) 지원에 의한 것임.
† 정 회 원 : 전남도립 담양대학 정보통신과
†† 정 회 원 : 전남대학교 대학원 전산학과
††† 종신회원 : 전남대학교 전산학과
논문접수 : 1997년 10월 23일, 심사완료 : 1998년 1월 16일

적으로 통제할 수 있어야 하며, 또한 관리 객체에서 발생하는 사건의 통지에 대해서도 효율적으로 제어하여야 한다.

망관리 정보의 접근 제어를 위한 포괄적인 클래스 정의 및 접근 제어 보안 모델을 정의한 ISO/IEC 10164-9(Objects and Attributes for Access Control) 권고안의 관리 객체 클래스 구조는 접근 제어 정책중 자율적 접근 제어(DAC: Discretionary Access Control) 정책인 접근 제어 리스트(Access Control List) 와 능력 리스트(Capability List), 그리고 강제적 접근 제어(MAC: Mandatory Access Control) 정책인 레이블 기반(Label based) 에 대해서만 정의를 하였을 뿐, DAC과 MAC의 단점을 보완한 새로운 접근 제어 정책으로서 최근 들어 활발히 연구가 진행되고 있는 역할기반 접근 제어(RAC: Role-based Access Control) 정책에 대한 정의가 포함되어 있지 않다[1][2]. 따라서 이에 대한 보완책으로서 기존의 구조에 역할기반 접근 제어 정책을 위한 객체 클래스와 속성을 정의해서 포함시킬 필요가 있다.

본 논문에서는 접근 제어를 위한 포괄적인 클래스 정의 및 접근 제어 보안 모델을 정의한 ISO/IEC 10164-9 권고안을 바탕으로 기존의 표준 관리 객체 클래스 구조를 크게 확장 및 보완하였다. 즉, 확장된 관리 객체 클래스 구조에서는 규칙의 구조를 보다 명확히 하기 위해 명시적 규칙과 묵시적 규칙으로 세분화하여 표현하였으며, 역할기반 접근 제어를 위한 역할 관리 객체 클래스를 포함시켰고, 또한 강제적 접근 제어의 보안등급 비교를 위해 제약사항 관리 객체 클래스를 추가함으로써 접근 제어 규칙의 명확성과 융통성을 함께 보장하였으며 확장된 접근 제어 규칙에 따라 프린터 관리 정보베이스를 이용해 실제 접근 제어 규칙이 적용되는 절차를 각 접근 제어 정책에 따라 적용하여봄으로써 접근 제어 규칙에 대한 타당성을 검증하였다.

마지막으로 권고안과 확장된 모델에 정의된 [illegible]의 비정형적인 구조를 명세언어 Z를 이용해 [illegible] 구조로 표현함으로써 관리 객체간의 연관성을 [illegible] 근 제어 규칙에 대한 세부적인 명세가 가능하여 [illegible] 작업 및 구현이 한층 편리하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관리 정보베이스의 보안 유지와 관련된 대표적인 3가지 접근 제어 정책에 대해 살펴본다. 3장에서는 관리 정보베이스에 대한 접근 제어의 확장으로 기존의 ISO/IEC 10164-9 권고안의 표준구조 및 확장된 관리객체 클래스에 대해 설명한다. 4장에서는 확장된 관리 객체 클래스를 이용해 접근 제어 규칙이 실제로 적용될 수 있는 접근 제어 시스템과 접근 제어 규칙이 적용되는 절차를 살펴봄으로써 타당성을 검증하고, 마지막으로 5장에서 결론을 맺는다.

2. 관리 정보베이스의 접근 제어 정책

망 자원에 접근을 원하는 사용자가 자신의 신원을 제시하고 인증 시스템으로부터 신원 인증을 받은 후, 확인된 사용자에 대한 망 자원을 접근하는 권한을 확인하는 과정을 접근 제어라고 한다. 이러한 접근 제어를 효과적으로 수행하기 위해서는 접근 권한의 불법 취득을 방지하고, 접근 권한에 관한 불법 변조가 일어나지 않도록 하여야 한다.

관리 정보베이스의 보안 유지와 관련된 대표적인 접근 제어 정책은 크게 자율적 접근 제어(DAC: Discretionary Access Control) 정책, 강제적 접근 제어(MAC: Mandatory Access Control) 정책, 그리고 역할기반 접근 제어(RAC: Role-based Access Control) 정책 등이 있다. 이 중에서 어느 접근 제어 정책을 선택할 것인가는 관리되어야 할 환경의 특성과 그 응용에 따라 달라질 수 있다[3].

자율적 접근 제어는 접근을 요청한 관리자의 신원(Identification)에 근거를 두고 있다. '자율적'이라고 하는 말은 관리자가 관리 객체에 대한 접근 권한을 자율적으로 부여하거나 철회할 수 있다는 것을 의미한다. 자율적 접근 제어에서는 관리 객체에 대한 관리자의 접근 권한을 접근 제어 행렬의 형태로 표현할 수 있다.

강제적 접근 제어는 관리자와 관리 객체의 보안 등급에 따라 접근을 제어하는 방법이다. 각각의 관리자와 관리 객체에게는 보안 등급이 부여되며, 특히 사용자의 보안 등급을 인허 등급(Clearance Level)이라고도 한다. 관리 객체와 관련된 보안 등급은 관리 객체에 포함된 정보가 불법적으로 누출되었을 때 입게 되는 손해의 정도, 즉 그 정보의 중요도를 나타낸다.

자율적 접근 제어와는 달리 강제적 접근 제어는 새로운 객체가 생성될 때 특정한 보안등급 부여 메카니즘에 의하여 객체에 보안등급이 부여되어야 한다. 강제적 접근 제어 정책은 모든 주체 및 객체에 대하여 일정하며 어느 하나의 주체/객체 단위로는 접근 제한을 설정

할 수 없다. 강제적 접근 제어 정책을 이용한 대표적인 예로 BLP(Bell and LaPadula) 모델이 있다[3]. BLP 모델은 정보의 비밀성을 중요시하며, 정보의 흐름이 낮은 곳에서 높은 곳으로 흐르게 되므로 비밀성보다는 무결성이 더 중요시되는 상업적인 응용에는 적합하지 못하다.

강제적 접근 제어 정책이 군대와 같은 엄격한 보안 통제를 필요로 하는 환경에서 개발되었고, 자율적 접근 제어는 학술 연구 단체와 같은 자율적이고 협동적인 환경에서 개발되었기 때문에, 두 보안 정책이 모두 상업적인 분야에 적용하기에는 다소 부적합한 면이 있다. 따라서 전통적인 자율적 접근 제어에서처럼 사용자나 사용자의 그룹에게 객체에 대한 접근 권한을 부여하고, 강제적 접근 제어에서처럼 접근 권한을 부여하는데 제한을 가할 수 있는 접근 제어 방법에 대한 연구가 진행되었다. 그 결과로서 역할기반 접근 제어 정책이 만들어지게 되었다.

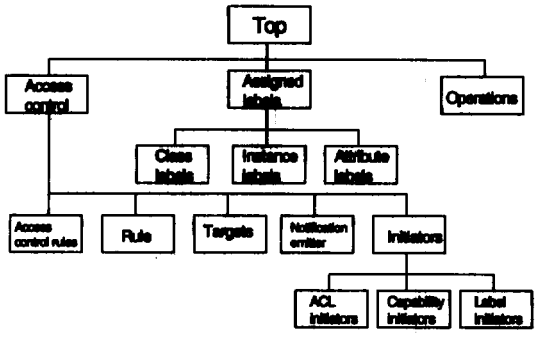
역할기반 접근 제어에서의 역할은 관리를 단순화하고 그에 따라 접근 권한을 단순화해주는 적당한 형태로 조직될 수 있다. 역할들은 함축적인 역할 상호간의 연관성에 따라 링크로 연결되며, 역할 격자구조를 형성한다. 역할기반 계층구조는 각 역할의 상호 연관성을 상위 역할로부터 하위 역할로 링크로 연결하여 계층구조를 이루도록 한다. 즉 이러한 계층구조에서 적은 권한을 가진 상위 역할은 보다 더 많은 권한을 갖는 하위 역할의 상위에 존재한다. 이 때 상위 역할을 묵시적으로 포괄하는 하위 역할이 아래쪽에 존재하여 하위에 있는 역할은 상위에 있는 역할의 권한을 묵시적으로 가질 수 있다. 이처럼 역할 계층 구조에는 객체지향의 중요한 개념인 상속개념이 적용되어 역할 권한의 흐름은 상위 역할에서 하위 역할로 적용될 수 있다. 역할의 계층구조는 권한 관리를 훨씬 단순화시킨다.

3. 관리 정보베이스 접근 제어의 확장

ISO/IEC 10164-9 권고안(국제표준)에는 접근 제어 관리기능에 대한 정의와 함께 관리 정보 및 연산에 대한 접근을 제어하기 위한 모델에 대하여 기술하고 있다[1]. 그리고 접근 제어 정책에 따라서 접근을 허용하거나 접근을 제한하는데 사용되는 관리 객체 및 속성들을 정의하고 있다[4][5].

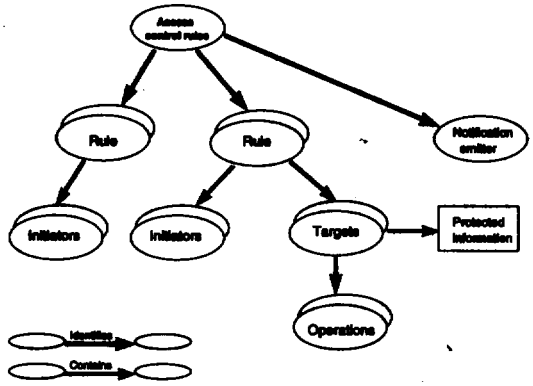
3.1 접근 제어 관리 객체 클래스 표준 구조

접근 제어 정보와 접근 제어 절차는 관리 객체로서 모형화 되며 그림 1은 ISO/IEC 10164-9 권고안(국제표준)에 정의된 관리 객체 클래스들의 상속 계층구조로서 여기에는 접근 제어를 위해 필요한 관리 객체 클래스들을 모형화하여 계층구조로 표현하였다.



(그림 3.1) 관리 객체 클래스 상속 계층구조
(Fig. 3.1) Managed Object Class Inheritane Hierarchy

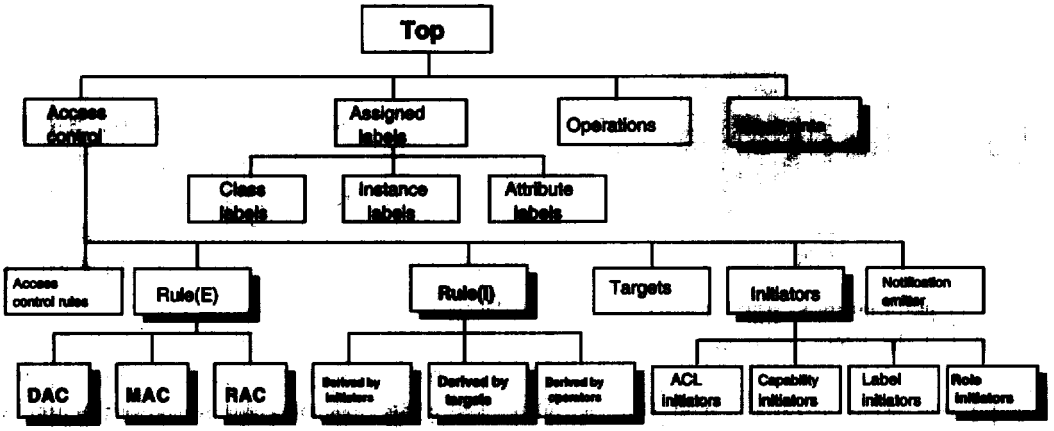
다음 그림 3.2는 그림 3.1에서 정의된 관리 객체 클래스 상속 계층구조를 이용해 관리 객체들 간의 상호 관계를 나타낸 것이다.



(그림 3.2) 관리 객체 상호관계
(Fig. 3.2) Managed Object Interrelation

3.2 확장된 접근 제어 관리 객체 클래스 구조

권고안(국제표준)에 정의된 접근 제어를 위한 관리 객체 클래스 구조는 접근 제어를 위한 각종 관련 정보 및 규칙을 보안관리자가 사전에 정의해 놓은 명시적인



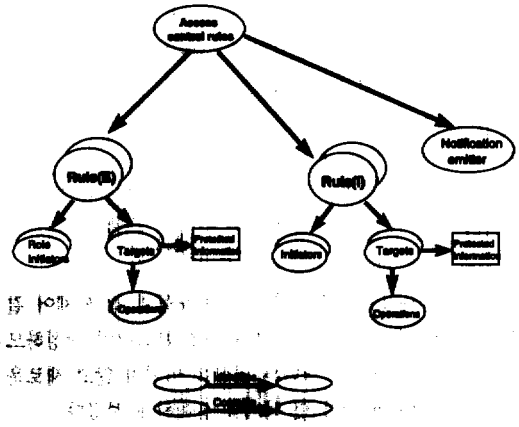
(그림 3.3) 확장된 관리 객체 클래스 상속 계층구조
(Fig. 3.3) Extended Managed Object Class Ingertance Hierarchy

접근 규칙(Explicit Access Rule)만을 정의함으로써 망의 규모가 점점 확대되고 이로 인한 관리 객체의 수가 급격히 증가되는 경우에 모든 관리 객체에 대해서 명시적 규칙을 낱낱이 정의하기가 사실상 불가능하다. 따라서 이에 대한 보완책으로서 기존의 권고안 구조에 묵시적인 규칙(Implicit Access Rule)을 포함하여 명시되지 않은 규칙에 대해서도 관리 객체간의 상호 관계를 이용해 접근 제어 규칙을 융통성 있게 적용함으로써 보안관리자의 권한부여 관리를 크게 단순화시킬 수 있고 각 규칙을 따로따로 정의하는데 따른 부가적인 간접경비를 대폭 줄일 수 있다.

다음 그림 3.3은 그림 3.1의 표준 관리 객체 클래스 계층구조를 확장한 것으로서 여기에는 규칙의 구조를 보다 명확하게 구체화하기 위해 명시적인 규칙(Explicit Rule)과 묵시적인 규칙(Implicit Rule)으로 구분하였다.

또한 RAC 정책을 지원하기 위해 'Initiators' 관리 객체에 'Role initiators' 관리 객체를 추가시킴으로써 DAC과 MAC의 단점을 보완할 수 있도록 하였으며, MAC에서 필요한 'Initiators'와 'Targets'의 레벨을 상호 비교하기 위해 'Constraints' 관리 객체를 추가하여 기존의 표준 구조를 크게 확장 및 보완하였다.

위에서 정의한 확장된 관리 객체 클래스 상속 계층 구조에 의해 다음 그림 3.4는 역할기반 접근 제어에 관련한 관리 객체 상호관계를 나타낸 것으로 명시적 규칙과 묵시적 규칙에 각각에 대해 'Role initiators'가 관리 객체들에게 역할을 부여하고, 주어진 역할에 따른 접근 제어를 할 수 있도록 함으로써 역할기반 접근 제어의 장점을 극대화하였다.

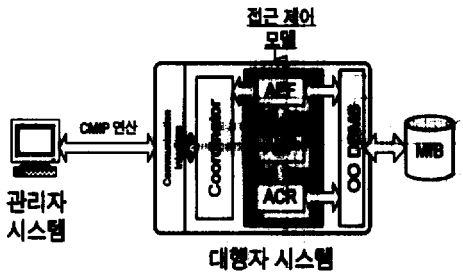


(그림 3.4) 관리 객체 상호 관계(역할 기반 접근 제어)
(Fig. 3.4) Managed Object Interrelation(RAC)

4. [illegible] 베이스 접근 제어 시스템

[illegible] 본 논문에서는 망관리에 필요한 모든 정보들을 저장하고 있는 개념적인 정보 저장소인 망관리 정보베이스에 대한 효율적인 접근 제어를 위해 접근 제어 모듈을 그림 5와 같이 크게 세 부분으로 나누었다[6].

먼저 접근 제어 수행(AEF:Access control Enforcement Function) 모듈은 관리자로부터 호출된 관리



(그림 4.1) 관리정보베이스 접근제어 시스템
(Fig. 4.1) MIB Access Control System

연산을 받아서 관리 객체에 접근하여 관리 연산을 수행하는 모듈로서, 관리 객체에 접근하기 위하여 먼저 접근 제어 결정 (ADF:Access control Decision Function) 모듈에 접근 허용 여부를 의뢰한다. 또한 관리 객체에서 발생한 사건 보고에 대하여도 마찬가지의 기능을 수행한다.

접근 제어 결정 (ADF) 모듈은 접근 제어 수행 모듈로부터 넘겨받은 접근 제어 정보를 접근 제어 규칙 모듈의 정보와 비교하므로써 접근 허용 여부를 결정하여 접근 제어 수행 모듈에게 통보하여 주는 역할을 수행한다.

접근 제어 규칙 모듈(ACR:Access Control Rule)은 접근 제어 결정 모듈에서 접근 허용 여부의 결정을 위하여 필요한 모든 정보를 제공하고 변경된 접근 제어 정보들을 첨가, 삭제, 그리고 수정하는 역할을 수행한다.

따라서 본 논문에서 제안한 접근 제어 규칙 모듈은 여러 가지 접근 제어 정책들에 대한 접근 규칙들을 하나의 모듈로 통합하였고, 접근 제어 수행 모듈과 접근 제어 결정 모듈 및 접근 제어 규칙 모듈들을 분리하여 작성하므로써 기존의 모듈들에 별다른 영향을 미치지 않고 새로운 접근 제어 정책의 삽입 및 삭제를 용이하게 하였다.

4.1 접근 제어 규칙

4.1.1 자율적 접근 제어 규칙

자율적 접근 제어 정책에 해당하는 접근 제어 리스트 스킴과 능력 리스트 스킴에 필요한 관리 객체 및 속성은 다음과 같다.

- ☐ 'ACL initiators' 관리 객체 클래스는 접근 제어 리스트의 이름을 포함하고 있으며 관련속성에는 Attributes of ACL initiators, Access control list가 있다.
- ☐ 'Capability initiators' 관리 객체 클래스는 능력 리스트의 이름을 포함하고 있으며 관련 속성에는 Capability identities list가 있다.

4.1.2 강제적 접근 제어 규칙

강제적 접근 제어 규칙에 필요한 관리 객체 및 속성을 중심으로, 권고안의 기본구조와 함께 확장된 구조에서는 'Initiators'에는 인가등급을 부여하고, 'Targets'에는 비밀등급을 부여한 후 'Constraints'를 추가하여 'Initiators'와 'Targets'간의 다양한 상호비교가 가능할 수 있도록 하였다.

- ☐ 'Assigned labels' 관리 객체는 레이블 타입 관리 객체를 포함하는 서브 트리의 최상위에 위치하며 우선 순위 상호관계를 통해 'targets'에 단일 보안 레벨을 할당한다. 관련속성에는 Label name, Security label이 있다.
- ☐ 'Class label' 관리 객체는 관리 객체 클래스의 'targets'에 대한 단일 보안 레벨을 연관시키는데 사용된다. 관련속성에는 Managed object classes가 있다.
- ☐ 'Instance label' 관리 객체는 개별적인 관리 객체의 'targets'에 대한 단일 보안 레벨을 연관시키는데 사용된다. 관련속성에는 Managed object instances가 있다.
- ☐ 'Attribute label' 관리 객체는 하나의 관리 객체안에 있는 특별한 속성에 대한 'targets'에 대해 단일 보안 레벨을 연관시키는데 사용된다. 관련속성에는 Managed object instance, Attribute identifier list가 있다.

4.1.3 역할기반 접근 제어 규칙

여기서는 ISO/IEC 10164-9 권고안을 바탕으로 역할기반 접근 제어 규칙을 Z 명세언어를 이용하여 모형화하고자한다[2][7].

역할기반 접근 제어 정책에 필요한 관리 객체 및 속성을 중심으로, 권고안의 기본구조와 함께 확장된 구조에서는 'Role Initiators'를 추가하여 역할기반에 의한 접근 제어가 가능하도록 하였다.

다음은 역할기반 접근 제어 관련 관리 객체 클래스들과 속성을 이용하여 접근 제어가 이루어지는 모형화에 대한 Z 표현으로서 접근 제어 정책에 공통으로 적용되는 관리 객체를 제외한 역할기반 접근 제어 관리 객체만을 나타내었다.

- ☐ 'Role Initiators' 관리 객체 클래스는 역할 관리

오퍼레이션에 대해 허용 가능한 'initiators'에 대해서 정의한다. 관련속성에는 Role Access Control List가 있다.

RoleACLinitiators

Initiators
ATTRIBUTE
roleAccessControlList : RoleAccessControlList
METHOD
GetRoleAccessControlList
Ξ roleAccessControlList
output! : RoleAccessControlList

output! = roleAccessControlList

4.2 접근 제어 수행

여기에서는 앞에서 모형화한 각 접근 제어 모델을 통해 망관리 정보베이스에 대한 안전한 접근 제어가 수행되는 절차를 확인한다. 즉, 각 접근 제어 정책에 따라 명시적인 규칙과 묵시적인 규칙에 대해 접근이 허용되는 경우와, 허용되지 않는 경우를 Z 스키마를 이용하여 접근 규칙으로서 정의하고, 실제 접근 제어가 이루어지는 과정을 프린터 관리 정보베이스를 모델로 하여 적용하여 봄으로써 접근 제어에 대한 수행 및 보안 검증을 하고자한다.

4.2.1 자율적 접근 제어 수행

자율적 접근 제어 보안 정책을 채택한 망관리 정보베이스에 대한 접근 제어를 위해 관리 정보베이스에 저장된 관리 객체 데이터에 대한 접근 요청을 보안 관리자에 의해 기술된 명시적 규칙과 접근 요청 'initiator', 'operator', 'object' 관리 객체별 상속특성에 의해 추론된 묵시적 규칙을 이용한 접근 규칙을 통해 규정할 수 있다[8].

따라서 묵시적인 규칙을 적용하기 위해 필요한 관리 객체, 접근 'initiator'간의 계층 구조 및 'operator', 'object' 특성에 따른 접근 권한 전파(propagation) 특성은 그림 4.2와 같다.

○ 접근 요청 'initiator'

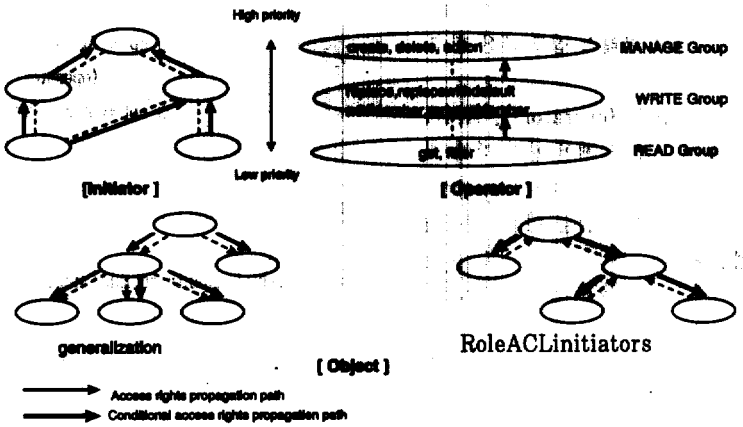
망관리 정보베이스에 접근할 수 있는 사용자(initiator)간 계층 구조에 의해 접근 권한은 상위 계층으로부터 하위 계층으로 전파된다.

○ 접근 요청 'operator'

ISO/IEC 10164-9 권고안에 제시된 9개의 관리 operation은 그 동작 특성에 따라 크게 3개의 그룹(READ, WRITE, MANAGE)으로 분류되며 접근 권한은 하위 그룹에서 상위 그룹으로 전파되는 특성을 갖는다.

○ 접근 요청 'object'

망관리 정보베이스에 저장되는 관리 객체 클래스간의 계층 구조와 관계에 의해 관리 객체 클래스간의 접근 권한 전파 특성이 달라진다. 일반화(generalization)



(그림 4.2) 접근 권한 전파 특성
(Fig. 4.2) Access Rights Propagation Properties

관계와 집단화(aggregation) 관계에서의 접근 권한 전파 특성은 일반적으로 상위 객체 클래스에서 하위 객체 클래스로 전달되지만, 집단화의 경우는 상위 객체 클래스에서 접근이 허용된 부분만이 하위 객체 클래스에서 접근이 허용된다.

망관리 정보베이스 접근 요청에 대한 타당성 검증용 묵시적 규칙은 주어진 접근 제어 요청에 대해 위의 접근 권한 전파 특성을 이용하여 구성한 추론된 접근 제어 규칙 집합에 하나 이상의 명시적 규칙이 포함되어 있는지를 확인하는 과정이다.

추론된 접근 제어 규칙
= {(inf-initiator, inf-operator, inf-object)} where
inf-initiator ∈ AccessPropagatedByIniiator (initiator?)
inf-operator ∈ AccessPropagatedByOperator (operator?)
inf-object ∈ AccessPropagatedByTarget (object?)

- AccessPropagatedByInitiator(initiator?), APBI(initiator?) : initiator들의 계층 구조에서 initiator?의 하위 계층에 속하는 initiator 집합 계산 함수
- AccessPropagatedByOperator(operator?), APBO(operator?) : operator들의 계층 구조에서 operator?의 하위 계층에 속하는 operator 집합 계산 함수
- AccessPropagatedByTarget(object?), APBT(object?) : 관리 대상 객체 클래스 계층 구조에서 object?의 상위 클래스 집합 계산 함수

위의 APBI(initiator?), APBO(operator?), APBT(object?) 계산 함수는 확장된 접근 제어 관리 객체 클래스 DerivedByInitiator, DerivedByOperator, DerivedByTarget 속성에 저장된 값을 이용, 추론된 접근 제어 규칙을 생성한다.

4.2.2 강제적 접근 제어 수행

망관리 정보베이스에 대한 강제적 접근 제어 규칙에서는 'Initiator'와 'Target'에 각각 인가등급과 비밀등급이 명시적으로 부여되기 때문에 묵시적 규칙은 적용되지 않는다.

따라서 접근 요청 타당성을 판단하는 기능은 'constraint' 객체의 접근 결정 메소드에 의해 수행된다. 접근 결정 메소드는 'initiator'의 인가등급과 'target'의 비밀등급을 비교하여 접근 요청 허용 여부를 판단하며 비교 방법은 적용되는 모델에 따라 달라질 수 있다.

4.2.3 역할기반 접근 제어 수행

망관리 정보베이스에 대한 역할기반 접근 제어 규칙은 자율적 접근 제어 규칙의 'initiator'에 역할을 부여한 'role initiator'를 정의함으로써 명시적 규칙과 묵시적 규칙, 그리고 접근이 허용되는 경우와 허용되지 않는 경우를 자율적 접근 제어와 유사하게 Z 언어로 표현할 수 있다[2][9].

MIBAccess_Validation

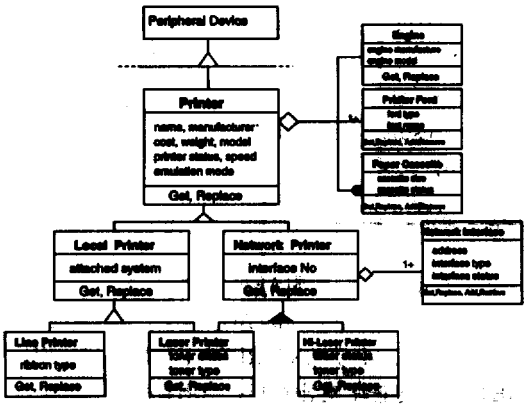
∃MIB
roleInitiator? : InitiatorName
operator? : OperationType
object? : ManagedObject
evaluation! : BOOLEAN

((∃ rule : Rule • roleInitiator? ∈ rule.roleInitiatorsList ∧
(object? ∈ rule.targetsList.managedObjectClasses ∨
object? ∈ rule.targetsList.managedObjectInstances) ∧
operator? ∈ rule.targetsList.operationsList)
//Explicit//
∨
(∃ rule : Rule; ∃ i ∈ AccessPropagatedByRoleInitiator (roleInitiator?);
∃ op ∈ AccessPropagatedByOperator (operator?);
∃ obj ∈ AccessPropagatedByObject (object?) •
i ∈ rule.roleInitiatorsList ∪ {roleInitiator?} ∧
(obj ∈ rule.targetsList.managedObjectClasses ∪ {object?} ∨
obj ∈ rule.targetsList.managedObjectInstances ∪ {object?}) ∧
op ∈ rule.targetsList.operationsList ∪ {operator?})
//Implicit//
⇔ evaluation! = True

위의 스키마 구조에서 묵시적인 규칙을 적용하기 위한 추론된 접근 제어 규칙은 자율적 접근 제어에서 적용된 규칙에 역할을 부여함으로써 역할에 의한 관리 객체 클래스 상호 관계를 추론할 수 있다.

4.3 접근 제어 시스템 예

여기에서는 망관리 정보베이스에 대한 확장된 접근 제어 규칙이 실제 적용될 수 있는 프린터 관리 정보베이스를 모델링하고자 한다. 모델의 구성요소는 주변장치인 프린터로서 계층구조로 구성된 프린터 자원에 대해서 다음 세 가지 유형의 사용자, 즉 일반 사용자, 중간 관리자, 그리고 시스템 관리자가 각각 프린터 자원에 대해 접근 요청을 했을 때 각 권한에 따른 적절한 접근 제어가 이루어지는가를 살펴보는 것이다. 위에서 정의한 프린터 관리 정보베이스에 대한 관리 객체를 OMT(Object Modeling Technique) 표기법에 의해 나타내면 다음 그림 4.3과 같다[10][11].



(그림 7) 프린터 관리 정보베이스 모델
(Fig. 7) Printer MIB Model

프린터 관리 정보베이스의 관리 객체에 접근하는 사용자의 계층구조는 크게 일반 사용자 층과 중간 관리자 층, 그리고 시스템 관리자층으로 구분되며 접근 권한은 시스템 사용자가 가장 높다.

4.3.1 접근 제어 검증 예

앞에서 정의한 접근 제어 규칙이 프린터 관리 정보베이스의 각 관리 객체에 대해 접근하려고 할 때, 각 사용자의 접근요청에 대해 명시적인 규칙은 물론 묵시적인 규칙에 대해서도 적절한 접근 제어가 이루어지는가를 확인해 봄으로써 접근 제어 규칙에 대한 수행 및 접근 제어 검증을 하고자한다.

명시적 규칙과 묵시적 규칙에 따른 접근 규칙의 구조는 다음과 같다.

(Initiator, Target, {Access_Type})

먼저, 자율적 접근 제어 규칙을 적용해보기위해 시스템 보안 관리자에 의해 사전에 각 관리 객체에 부여된 명시적 규칙은 다음과 같다.

□ **명시적 규칙(Explicit Rule)**

(U1, Printer, {all})
(U2, Local Printer, {get, replace})
(U3, Network Printer, {replace})
(U4, Local Printer, {print})
(U5, Laser Printer, {get})

여기서 get 오퍼레이션은 CMIP 오퍼레이션의 READ 그룹 오퍼레이션에 해당하고, replace 오퍼레이션은 CMIP 오퍼레이션의 WRITE 그룹 오퍼레이션에 해당한다. 그리고 print 오퍼레이션은 CMIP 오퍼레이션의 create, delete, action을 포함한 MANAGE 그룹 오퍼레이션에 해당한다. 위에서 정의된 명시적 규칙에 의한 접근 제어는 해당 규칙과 일치하는 접근 요청에 대해서만 망관리 정보베이스에 대한 접근을 허용한다.

다음에는 명시적 규칙에 정의되어 있지 않은 관리 객체에 대해 접근을 요청하는 경우, 즉 묵시적 규칙(Implicit Rule)에 대한 접근을 제어하는 과정을 살펴보자.

□ **접근 요청1**

(U3, Network Printer, getInterfaceNo)

중간 관리자 U3가 Network Printer에 대해 interfaceNo 속성값을 가져오기 위해 접근을 요청한다.

위의 접근 요청1에 대한 명시적 규칙은 정의가 되어 있지 않으므로 묵시적인 규칙을 적용하기 위해서는 관리 객체 클래스간의 계층구조를 분석하여 해당 관리 객체를 캡슐화 해주는 함수를 이용하여 해당 관리 객체를 추출한 후 규칙을 만족하면 접근을 허용하고 그렇지 않으면 접근을 허용하지 않는다.

각 관리 객체에 대해 추론된 접근 제어 규칙을 생성하여 적용해보면 다음과 같다.

APBI(U3) = {U4,U5} ∪ {U3} = {U3,U4,U5}
APBT(Network Printer)

= {Printer} ∪ {Network Printer}
= {Printer, **Network Printer**}
APBO(getInterfaceNo)
= {create, delete, action, replace, add Member, removeMember, replaceWithDefault} ∪ {get}
= {create, delete, action, **replace**, add Member, removeMember, replaceWithDefault, get}
Evaluation Result ⇒ **Grant**

위의 결과를 분석해 보면 접근 요청1에 대한 명시적 규칙은 정의되어 있지 않으나 묵시적 규칙에 의해 각 관리 객체를 추출해 보면 결국 U3에 대한 명시적 규칙(U3, Network Printer, {replace}) 이 적용됨으로써 접근 요청1에 대해 접근을 허용(Grant)한다는 것을 알 수 있다.

□ **접근 요청2**

(U2, Hi-Laser Printer, print)

중간 관리자 U2가 Hi-Laser Printer 대한 접근 요청이다.

접근 요청1에서와 같은 방법으로 추론된 접근 제어 규칙을 생성하여 적용해보면 다음과 같다.

APBI(U2) = {U2, U4}
APBT(Hi-Laser Printer)
= {Hi-Laser Printer} ∪ {Network Printer} ∪ {Printer} = {Hi-Laser Printer, Network Printer, Printer}
APBO(print) = ∅ ∪ {print} = {print}
Evaluation Result ⇒ Deny

위의 결과를 분석해 보면 접근 요청3에 대한 명시적 규칙이 정의되어 있지 않아 묵시적 규칙에 의해 각 관리 객체를 추출해 보았으나 U2 및 U4에 대한 명시적 규칙이 적용되지 않아 결국 접근 요청3에 대해 접근을 거절한다는 것을 알 수 있다.

4.4 망관리 정보베이스 접근 제어 모듈

망관리 정보베이스 접근 제어 모듈은 그림 5와 같이 세 부분, 즉 접근 제어 수행(AEF) 모듈, 접근 제어 결정(ADF) 모듈, 접근 제어 규칙(ACR) 모듈로 나뉜다. 접근 제어 규칙에 관한 세부적인 명세는 4.1에 기술되어있다. 따라서 여기서는 접근 제어 결정 모듈과 접근 제어 수행 모듈에 대한 알고리즘 명세를 Z언어를 이용해 나타낸다[12].

4.4.1 접근 제어 결정(ADF) 모듈

접근 제어 결정 모듈은 접근 제어 수행 모듈의 요청을 받아 세가지 접근 제어 정책 즉, 자율적 접근 제어, 강제적 접근 제어, 역할기반 접근 제어 중 해당 접근 제어를 결정하게 된다. 이들중에서 강제적 접근 제어를 결정하는 모듈(MAC_ADF)을 다음과 같이 기술할 수 있다.

MAC_ADF

∃MAC_ACL_RULE
initiator? : InitiatorName
operator? : OperationType
object? : ManagedObject
result! : BOOLEAN

IF (initiator ?, operator?, object?) ∈ MAC_ACL_RULE
THEN
result! = TRUE
ELSE
result! = FALSE

4.4.2 접근 제어 수행(AEF) 모듈

M-GET, M-SET, M-ACTION, M-CREATE, M-DELETE, M-EVENT-REPORT, M-CANCEL-GET 등의 망관리 연산에 대한 API들을 명세한다. 이들중에서 M-GET 연산에 대한 API는 다음과 같이 기술할 수 있다.

M-GET

∃MIB
base_object_class? : ManagedObjectClass
base_object_instance? : ManagedObject
scope? : ScopeType
filter? : FilterType
access_control? : AccessConrolType
synchronization? : SynchronizationType
attribute_identifier_list? : Seq AttributeType
attribute_value_list! : Seq Attribute Type

```
attribute_value_list! = 〈 〉
IF synchronization = "best-effort"
THEN
  ∀filtered_object : ManagedObject
        ∈ Filtering(Scoping(base_object_class,
            base_object_instance, scope))
    IF ADF(m-get, filtered_object, initiator)
    THEN
        attribute_value_list!'
          = attribute_value_list! ∪
            〈 Read_attribute (filtered_object,
              attribute_identifier_list) 〉
ELSE
  ∀filtered_object : ManagedObject
    ∈ Filtering(Scoping(base_object_class,
      base_object_instance, scope)) ∧
      ADF(m-get, filtered_object, initiator)
      attribute_value_list!' =
                        attribute_value_list! ∪
       〈 Read_attribute (filtered_object,
         attribute_identifier_list) 〉
```

본 논문에서 제안한 접근 제어 모듈은 여러 가지 접근 제어 정책들에 대한 접근 규칙들을 하나의 모듈로 통합하였고, 접근 제어 수행 모듈과 접근 제어 결정 모듈 및 접근 제어 규칙 모듈들을 분리하므로써 기존의 모듈들에 별다른 영향을 미치지 않고 새로운 접근 제어 정책의 삽입 및 삭제를 용이하게 하였다.

5. 결 론

망관리 시스템의 여러 가지 구성 요소들 중 가장 핵심적인 요소 중의 하나는 망관리에 필요한 정보들인 관리 객체들의 개념적인 저장소인 관리 정보베이스이다. 관리 정보베이스에 저장된 관리 객체들은 망관리에 필수적이며 중요한 모든 정보들을 유지하고 있기 때문에 안전하게 유지되어야 한다.

본 논문에서는 관리 영역간 접근 제어 상호 운영을 통해 전체적인 망관리를 보다 효율적으로 수행할 수 있는 관리자 기본 모델을 제안하였다. 즉 접근 제어를 위한 포괄적인 클래스 정의 및 접근 제어 보안 모델을 정의한 ISO/IEC 10164-9 권고안을 바탕으로 기존의 표준 관리 객체 클래스 구조를 크게 확장 및 보완하였다. 확장된 관리 객체 클래스 구조에서는 규칙의 구조를 보다 명확히 하기 위해 명시적 규칙과 묵시적 규칙으로 세분화하여 표현하였다. 그리고 역할기반 접근 제어를 위한 역할 관리 객체 클래스를 포함시켰고 강제적 접근 제어의 보안등급 비교를 위해 제약사항 관리 객체 클래스를 추가함으로써 접근 제어 규칙의 명확성과 융통성을 함께 보장하였다. 또한 확장된 접근 제어 규칙에 따라 프린터 관리 정보베이스를 이용해 실제 접근 제어 규칙이 적용되는 절차를 각 접근 제어 정책에 따라 적용하여봄으로써 접근 제어 시스템에 대한 수행부분을 검증하였다.

마지막으로 접근 제어 시스템의 각 기능별 알고리즘과 권고안 및 확장된 모델에 정의된 GDMO의 비정형적인 구조를 명세언어 Z를 이용해 정형화된 구조로 표현함으로써 관리 객체간의 연관성은 물론 접근 제어 규칙에 대한 세부적인 명세가 가능하여 표준화 작업이 한층 편리하도록 하였다.

참 고 문 헌

[1] ISO/IEC 10164-9/ITU-T X.741, "Objects and Attributes for Access Control"

[2] Matunda Nyanchama, Sylvia Osborn, "Role-Based Security, Object Oriented Databases & Separation of Duty," SIGMOD RECORD, Vol. 22, No. 4, December pp. 45-51, 1993.

[3] David d. Clark, David R. Wilson, " A Comparison of commercial and Military computer security policies," IEEE, 1987.

[4] Oliver Festor, Georg Zormtlein, "Formal Description of Managed Object Behaviour - A Rule Based Approach," IFIP Integrated Network Management, pp45-58, 1993.

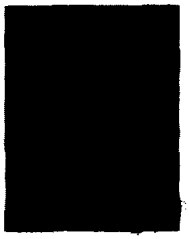
[5] E. B. Fernandez, R. B. France, and D. Wei, "A formal specification of an authorization model for object-oriented database," Worksh-ops In Computing security for object-oriented systems, Washington DC, 1996.

[6] Rumbaugh J, Michael Blaha, "Object-Oriented Modeling and Design," Prentice Hall, Inc 1991.

[7] David Rann John Turner and Jenny Whitworth, "Z: A Beginner's Guide," School of Computing Staffordshire University UK, 1994.

[8] Ravi S. Sandhu, Pierangela Samarati, "Access

Control: Principles and Practice," IEEE Communications Magazine, September 1994.
[9] ISO/IEC 10165-2/ITU-T X.721, "Definition of Management Information," 1992.
[10] ISO/IEC 10165-4/ITU-T X.722, "Guidelines for the Definition of Managed Objects," 1992.
[11] ISO/IEC 10164-3/ITU-T X.732, "Attributes for Representing Relationships," 1992.
[12] Jong Duk Kim, Yong Min Kim, Young Kyun Kim, Bong Nam Noh, "An Access Control Modeling for Network Management Information Base", IEEE Korea International Symposium on Network Operations and Management, April 1996.

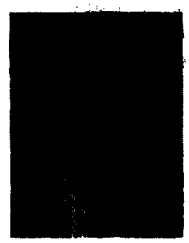


김 종 덕

1983년 전남대학교 전산학과 졸업(이학사)
1988년 국방대학원 전자계산학과 졸업(이학석사)
1997년 전남대학교 대학원 전산통계학과 졸업(이학박사)
1995년~1997년 전남대학교 전산학과 시간강사
1998년~현재 전남도립 담양대학 정보통신과 전임강사
관심분야 : 정보통신 보안, 컴퓨터 네트워크, 객체지향 시스템 등

이 형 효

1987년 전남대학교 전산학과 졸업(이학사)
1989년 한국과학기술원 전산학과 졸업(공학석사)
1990년~1992년 삼보컴퓨터 기술연구소
1993년~1997년 한국통신 연구개발원
1995년 정보처리기술사(전자계산조직응용)
1997년~현재 전남대학교 전산학과 박사과정
관심분야 : 통신망관리, 정보보안, 객체지향 시스템 등



노 봉 남

1978년 전남대학교 수학교육과 졸업(이학사)
1982년 한국과학기술원 전산학과 (공학석사)
1994년 전북대학교 대학원 전산통계학과(이학박사)
1983년~현재 전남대학교 전산학과 교수
관심분야 : 객체지향 시스템, 통신망 관리, 정보보안, 컴퓨터와 정보사회 등