

---

# 통합 분산환경에서 타입 정보를 이용한 지속성 객체의 보안 모델 및 응용

김영수\* · 최홍식\*\*

## Security Model and Application of Persistent Object using Type Information in Integrated Environment of Distributed System

Young-Soo Kim\*, Heung-Sik Choi\*\*

### 요 약

분산시스템은 호환성과 이식성 그리고 보안 문제를 확대하고 있다. 이의 해결을 위하여 네트워크와 데이터에 대한 접근 투명성을 제공하는 코바와 객체데이터베이스가 폭넓게 사용되고 있으나 보안성의 보장을 위하여 사용하는 메서드와 속성지향적인 접근제어기법은 자료가 방대하고 사용자가 다수인 경우에는 가용성을 제한하고 효과적이지 못하다. 이는 접근처리 지연과 네트워크폭주를 야기하는 객체의 인스턴스를 통한 접근제어를 수행하기 때문이다. 따라서 본 논문에서는 이의 해결책으로 보안성과 가용성을 동시에 고려하는 접근제어시스템으로 타입정보를 사용한 지속성객체의 접근제어 모델을 제안하고 이의 검증을 위하여 코바와 객체DB시스템의 접근제어모델과 분리 통합되는 형태로 지속성보안시스템을 구현하였다.

### ABSTRACT

As a large number of distributed systems becoming more popular, interoperability, portability and security are becoming major concerns of modern computing. CORBA and object-oriented database which provide transparency of network and database are increasingly being used as the basis for distributed system to solve these problems. The two methods can help accomplish assurance of security by using a method-based access control technique or an attribute-based access control technique. These methods also enhance the unavailability or inefficiency caused by the delay of access process and bottleneck of the network due to the complex instance-based access control. We propose a security model on the type information based access control system that can enhance both security and availability by separating the functions delivered from CORBA and object-oriented databases. We apply the access control model specifically to enhancement of security system and also perform a test to verify the security and availability of our model.

### 키워드

분산시스템, 접근제어, 타입정보, 지속성객체, 보안모델

1. 서 론

네트워크의 확산과 아울러 기존의 클라이언트/서버 환경을 보다 발전시킨 분산시스템이 보편화되고 있다. 이러한 분산시스템은 다수의 호스트가 네트워크에 상호 연결되어 있고 다양한 사용자가 공통적으로 이용하기 때문에 심각한 보안문제를 갖고 있다. 또한 인터넷이 분산시스템을 위한 인프라로 폭넓게 사용되고 해커의 위협이 증가하면서 분산정보에 대하여 허가되지 않은 접근이 확대되고 있다.

따라서 이러한 불법적인 접근을 방어하는 접근 제어에 대한 필요성은 더욱 중요시되고 있다. 그러나 접근제어를 강화하면 사용자의 가용성이 제한되므로 분산정보에 대한 투명한 접근방법이 제시되어야 한다. 따라서 분산시스템은 대부분 객체지향 파라다임을 수용하고 있는 분산객체미들웨어인 코바시스템과 객체DB시스템을 사용하여 구축되고 있다. 클라이언트와 서버시스템 사이의 요청 전달은 코바기술을 이용하고 데이터에 대한 접근은 객체라는 형태로 단일화되고 일관된 데이터구조를 제공하는 객체DB시스템을 사용한다[1]. 서버나 클라이언트의 수명에 관계없이 유지되는 지속성객체는 기업이나 조직을 유지하고 관리하는데 필수적 데이터로서 클라이언트의 요청에 의하여 DB에서 서버시스템의 프로세스 메모리로 실체화되어 접근된다.

이들 지속성객체의 접근제어를 위하여 코바는 메서드 지향적인 접근제어기법을 사용하는 반면 객체DB시스템은 속성 지향적인 접근제어를 통하여 보안을 실현하고 있다[2]. 본 논문에서는 그림 1과 같이 코바와 객체DB시스템의 접근제어모델에 대한 통합적 접근을 통하여 클래스라는 타입정보를 이용한 접근제어 모델을 도출한다. 제안 모델은 지속성객체의 접근단계에서 지속성객체의 타입을 대상으로 접근제어를 수행함으로써 신속한 접근제어와 지속성객체의 인스턴스 과정을 제거함으로써 통신 오버헤드를 감소시킬 수 있다.

이를 위한 연구방법은 그림 2와 같이 코바와 객체데이터베이스의 접근제어 모델을 분석하여 타입을 이용한 접근제어 보안모델을 도출한 후 코바기반 서버시스템에서 이를 구현하여 모델의 실용성을 검증하였다.



그림 1. 통합적 접근제어 도출 모델  
Fig. 1 The suggested model of integrated access control

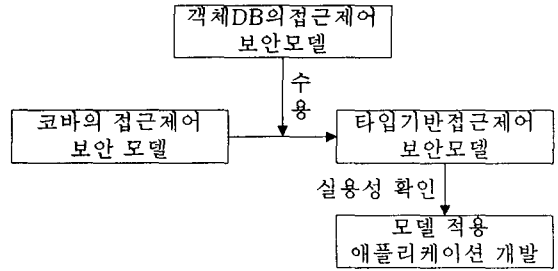


그림 2. 연구 접근 모델  
Fig. 2 The model of research approach

II. 객체보안시스템의 연구 모델

2.1 코바시스템의 보안 모델

코바는 다중 사용자 환경의 클라이언트/서버구조를 지원하는 분산객체기술로서 분산시스템의 구축을 위하여 폭넓게 사용되고 있다. 이는 이종 분산 시스템간의 상호호환성과 객체 모델의 일관된 접근을 지원해주기 때문이다. 코바시스템은 모든 컴포넌트를 객체지향 파라다임의 객체로써 취급하고 메서드를 통하여 접근제어를 수행한다. 클라이언트는 객체에 대한 접근을 위하여 객체 리퍼런스를 이용하며 객체의 타입과 수행될 동작에 대한 메서드 정보를 가지고 서비스를 요청하게 된다. 따라서 코바서버시스템내의 객체 데이터에 대한 접근제어는 각 객체의 메서드에 대한 클라이언트의 사용권한을 부여하는 과정을 필요로 한다[3].

코바시스템이 접근제어를 수행하는 절차는 그림 3과 같고 클라이언트가 인증수행 후 획득하는 보안토큰을 첨부하여 서버시스템의 구현객체에게 메서드의 수행을 요청하면 소프트웨어 버스 역할을 수행하는 ORB(Object Request Broker)가 요청 메시지를 인터셉트하고 이를 접근허용유무를 결정하는 접근제어객체에게 보낸다. 접근제어객체는 보안토큰과 메서드에 대한 접근권한리스트를 참조하여 접근을 제어한다. 코바의 보안시스템은 사용자

를 특정 그룹에 소속시키고 객체를 도메인으로 통합하고 메서드를 접근모드로 그룹핑함으로써 일괄적으로 보안등급을 설정할 수 있도록 하고 있다[4].

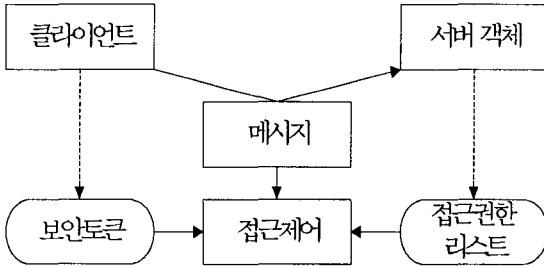


그림 3. 코바의 접근제어 모델  
Fig. 3 The access control model of CORBA

### 2.2 객체DB시스템의 보안 모델

데이터베이스는 응용시스템이 사용하는 기업의 필수적인 업무 데이터들을 캡슐화하고 있는 지속성 객체들의 집합이다. 분산 네트워크와 객체지향 기술이 확산되면서 분산DB시스템으로 객체라는 단일화되고 일관된 모델을 지원하는 객체DB시스템이 널리 사용되고 있다. 객체DB시스템은 객체지향 애플리케이션이 보는 관점 그대로 객체를 저장하고 추출하는 것을 가능하게 한다. 이는 DB의 자료구조와 애플리케이션의 자료구조가 객체라는 단일화되고 일관된 형태로 유지하기 때문에 데이터 변환과 같은 부가적인 작업을 필요로 하지 않고 질의를 던지고 결과를 전달받는 질의 전달형(query shipping)구조와는 달리 클라이언트와 DB서버간에 실제 자료가 오고가는 자료전달형(data shipping)구조를 채택하고 있어서 이들 데이터를 클라이언트에 캐싱해서 성능향상을 도모하는 것도 가능하게 한다[5].

그러나 분산 데이터베이스는 액세스 포인트의 확대로 인하여 집중형 데이터베이스보다 많은 보안 위협에 노출되어 있다. 따라서 데이터베이스에 저장된 객체는 인가되지 않는 검색과 변경 그리고 부정확한 변경으로부터 보호되어야 한다. 이를 위하여 객체DB시스템은 쿼리에 의한 객체내의 데이터에 대한 직접 접근을 허용하는 매커니즘으로 인하여 속성 데이터에 대한 비밀등급을 지정하고 이를 통한 접근제어를 수행한다.

객체내의 데이터에 대한 접근제어를 위하여 코바는 메서드 지향적인 접근제어기법을 사용하는 반면 데이터베이스시스템은 속성 지향적인 접근제어를 사용한다. 이와같이 객체DB시스템은 객체를 보호하기 위하여 객체내의 속성 데이터에 대한 비

밀등급을 유지하고 접근을 제어하는 것에 초점을 맞추고 있다[6]. 데이터베이스 보안시스템은 그림 4와 같고 애플리케이션은 사용자 정보로부터 생성한 보안토큰을 쿼리와 함께 전송하면 DB서버시스템은 접근권한리스트내의 객체에 대한 비밀등급과 사용자의 권한을 표시하는 보안토큰내의 인가등급의 비교를 통하여 접근제어를 수행된다.

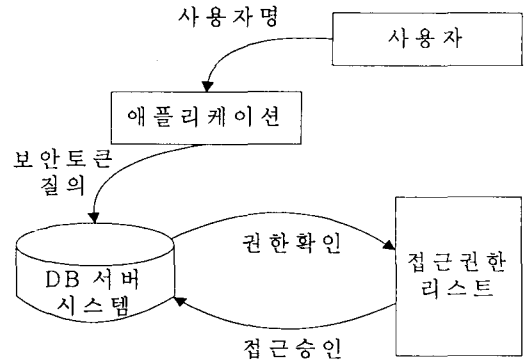


그림 4. 데이터베이스 보안  
Fig. 4 Database security

### 2.3 보안토큰과 접근권한리스트의 모델링

코바와 객체DB의 응용 클라이언트는 사용자의 인증을 통하여 접근권한정보로 구성된 보안토큰을 획득한다. 인증은 시스템의 부당한 사용이나 정보의 부당한 전송을 방어할 수 있는 기술로써 객체에 대한 접근제어 및 사용자의 책임 추구성을 수립하기 위한 기본 요소로서 클라이언트가 사용자에 대한 진정성을 확인하고 사용자의 인가등급을 탑재한 보안토큰을 획득하게 한다. 보안토큰의 구조는 그림 5와 같이 기본 구성요소로 사용자 식별자와 소속 그룹 그리고 인가등급을 포함한다.

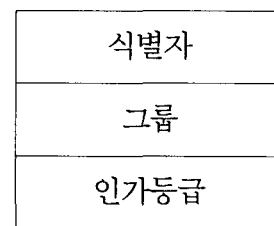


그림 5. 보안토큰의 구조  
Fig. 5 The architecture of security token

접근권한리스트는 테이블의 형태로 모델링되고

각 행렬의 엔트리는 접근허가(권한 또는 특권)을 나타낸다[7]. 접근권한메커니즘은 그림 6과 같이 ACL(Access Control List)과 CL( Capability List) 그리고 SL(Security Label)에 의하여 표현된다[8]. ACL은 객체를 소유하는 사용자들의 인가등급을 표현하는 반면 CL은 사용자가 소유할 수 있는 객체들의 비밀등급을 표현한다. 따라서 ACL은 사용자나 사용자 그룹이 소수일 때 유리하고 하나의 객체에 모든 사용자가 대응되므로 목록이 길어지면 유지 및 탐색 오버헤드가 크다. 반면 CL은 객체의 수가 적을 때 편리하나 주어진 객체를 접근할 수 있는 사용자들을 파악하는데 시간이 많이 걸리고 비효율적이다. SL은 사용자와 객체에 대하여 비밀 수준을 나타내는 레이블을 정의하고 사용자가 접근요청시 사용자의 보안레이블을 전송하면 객체는 자신에게 부여된 보안 레이블과의 비교를 통하여 접근을 승인할 것인지 또는 거부할 것인지 결정한다. 따라서 다수의 사용자와 분산정보로 구성되는 분산시스템의 보안을 위하여 널리 사용된다.

| <ACL 기법> |      | <CL 기법> |      | <SL 기법>       |  |
|----------|------|---------|------|---------------|--|
| 객체 X     |      | 사용자 A   |      | 사용자 A 또는 객체 X |  |
| 사용자      | 접근허가 | 객체      | 접근허가 | 인가등급, 비밀등급    |  |
| A        | 1    | X       | 1    | Secret        |  |
| .        | .    | .       | .    | .             |  |

그림 6. 접근권한리스트의 표현 방법  
Fig. 6 The representation method of access right list

### III. 지속성객체시스템의 보안 모델 및 응용

#### 3.1 지속성 객체시스템의 구조

지속성객체는 서버나 클라이언트의 수명에 관계 없이 존재하고 계속되는 오브젝트로서 기업의 업무데이터를 캡슐화하고 있다. 지속성 객체는 단일의 실체를 표현하는 객체로서 DB에 정적인 상태로 유지되고 접근시 프로세스 메모리에 인스턴스로서 활성화된다. 이는 지속성 객체에 대한 클라이언트의 작업 결과는 프로세스 메모리와 DB의 지속성객체에 투명하게 반영되어 일관된 상태로 유지되는 것을 필요로 한다[9]. 지속성객체시스템은 그림 7과 같이 임시객체시스템을 확장하는 구조로 구성한다. 이와 같은 지속성객체시스템을 사용하는 클라이언트는 지속성객체시스템을 통하여 임시객체

시스템의 모든 서비스를 사용할 수 있다.

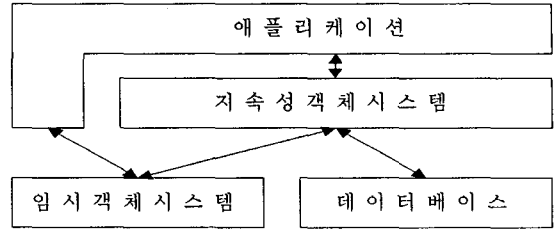


그림 7. 지속성객체시스템의 아키텍처  
Fig. 7 The architecture of persistent object system

객체DB시스템은 지속성객체의 컬렉션을 효율적으로 저장하는데 중점을 두는 반면 코바는 분산되어 있는 지속성객체의 연결을 위한 통신메커니즘이 관심의 대상이다. 따라서 코바와 객체DBMS의 통합에 의하여 객체의 지속성서비스는 효율적으로 달성된다[10]. 통합분산시스템의 지속성 객체 모델은 그림 8과 같고 클라이언트가 지속성 객체의 리퍼런스를 요청하면 지속성객체 서비스는 객체DBMS와 상호작용을 통하여 DB내의 지속성 객체를 프로세스 메모리상에 적재한 후 매핑을 위한 코바 객체를 생성한다. 또한 지속성객체 서비스는 DB와 코바의 지속성객체를 단일하고 일관된 형태로 유지 관리한다[2]. 따라서 애플리케이션은 두 시스템 사이의 상이한 자료구조간의 매핑작업이 불필요하게 되어 처리를 신속하게 할 수 있다. 기업의 응용업무를 위하여 사용되는 필수적인 데이터를 캡슐화하고 있는 지속성객체는 기밀성과 접근제어를 필요로 한다. 접근제어는 비인가된 변조 또는 조작을 방지함으로써 정보의 무결성을 보장하고 접근권한을 가진 사용자에게 한하여 접근을 허용함으로써 가용성을 증가시키고 비인가자에게 비밀 정보가 공개되는 것을 방지한다[11].

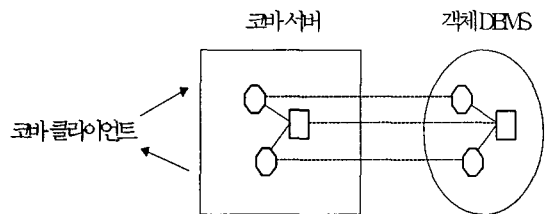


그림 8. 통합분산시스템의 지속성객체 모델  
Fig. 8 The integrated persistent object model of distributed system



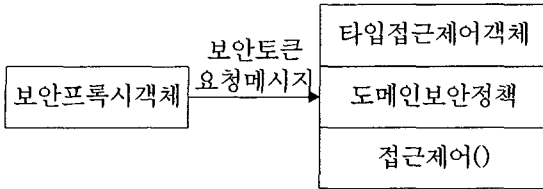


그림 12. 타입접근제어 객체  
Fig. 12 A type based access control object

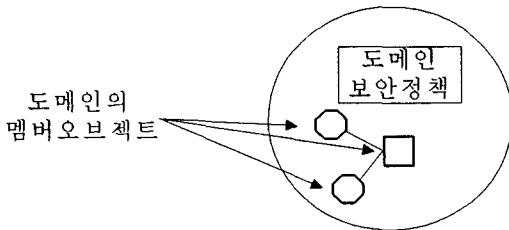


그림 13. 오브젝트의 도메인 구조  
Fig. 13 The domain architecture of objects

객체지향시스템에서 클래스는 유사한 객체들의 공통적인 속성과 메서드에 대한 정보로 표현되고 인스턴스를 생성하는 템플릿로서의 역할을 수행한다. 접근권한객체는 그림 14와 같고 지속성객체의 타입정보인 속성과 메서드에 대하여 비밀등급을 부여하고 있다. 지속성객체의 논리적인 구조와 정의를 기술하고 있는 클래스라는 타입을 이용한 지속성객체의 보호는 접근단위가 타입으로 정의되고 보안 등급도 속성과 메서드과 같은 타입단위로 이루어진다.

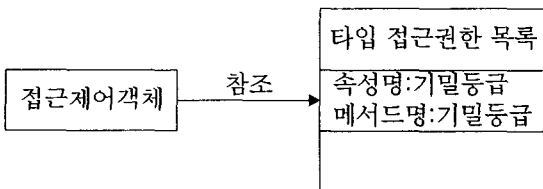


그림 14. 타입접근권한 객체  
Fig. 14 A type based access right object

IV. 지속성객체 응용시스템의 설계 및 모델의 검증

4.1 응용시스템의 통합구조

본 연구에서는 코바시스템 및 객체DB시스템과 함께 배포되고 자주 사용되는 은행의 계좌관리시스템을 연구시스템으로 사용하여 지속성객체의 보안모델을 검증한다. 사용자는 은행업무를 보기 위하여 사용자 인증을 받은 후에 예금계좌에 대한 입출금을 행한다. 따라서 계좌관리시스템은 그림 15와 같이 은행객체와 고객객체 그리고 계좌객체로 모델링된다. 고객객체는 사용자 인증을 위하여 사용되는 지속성 객체이고 은행객체는 입출금의 대상이 되는 계좌객체를 관리하는 팩토리객체(factory object)이다. 계좌객체는 보통예금 또는 당좌예금으로 인스턴스화되는 지속성객체로 객체DB시스템에 저장되어 관리된다. 클라이언트는 계좌객체에 대한 리퍼런스를 얻기 위하여 임시객체인 은행객체를 사용하여 DB내의 계좌객체에 접근한다.

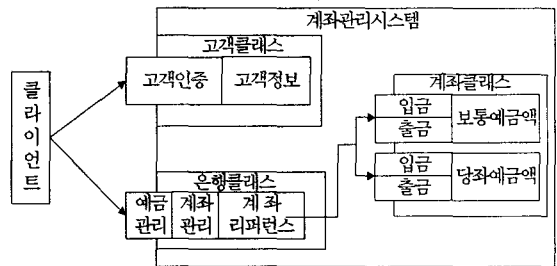


그림 15. 계좌관리시스템의 객체 모델  
Fig. 15 The object model of account management system

은행의 계좌관리시스템이 구현된 코바와 객체DB시스템의 통합분산시스템은 그림 16과 같다. 코바와 객체DB시스템을 연동하는 구조에서 코바기술은 클라이언트와 서버사이의 요구전달에 이용하고 데이터베이스 시스템의 접근은 코바의 서버 객체에 의해 간접적으로 이루어진다. 클라이언트는 사용자로부터 계좌객체에 대한 요청을 받아 코바 서버에 전달하고 그 결과를 적절하게 사용자에게 보인다. 코바 서버는 데이터베이스내의 계좌객체를 적재하기 위한 트랜잭션을 수행한다. 데이터베이스시스템의 관점에서 볼 때 코바서버는 데이터베이스시스템의 클라이언트이다. 객체기반의 통합분산시스템은 클라이언트가 계좌객체를 투명하게 사용할 수 있도록 코바와 DBMS를 연결하고 계좌객체를 전달하는 지속성서비스를 제공한다.

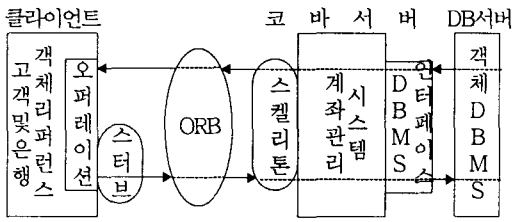


그림 16. 코바 및 객체DB 응용시스템의 통합 구조  
Fig. 16 The integrated architecture of applied CORBA and object database system

4.2 지속성객체 보안시스템의 응용구조

서버시스템의 계좌객체에 대한 클라이언트의 접근처리는 데이터베이스의 계좌객체로 투명하게 반영된다. 클라이언트가 DB의 계좌객체에 캡슐화되어 있는 예금액에 접근하기 위해서는 먼저 사용자 인증을 통하여 얻은 보안토큰과 계좌객체의 팩토리 객체인 은행객체의 리퍼런스를 획득하고 이를 사용하여 서버시스템에게 계좌객체에 대한 접근요청한다. 이때 계좌객체에 대한 접근제어를 수행하는 보안시스템은 그림 17과 같이 계좌객체가 DB로부터 적재되어 인스턴스화되는 이전단계에서 보안토큰과 계좌객체의 리퍼런스로부터 획득한 타입정보를 기초로 접근제어를 수행한다.

보안 프록시 객체는 계좌객체에 접근하려는 사용자가 허가된 사용자인가를 확인한 후에 타입접근제어객체에게 접근제어 실행권을 넘겨준다. 접근제어객체는 클라이언트가 요청한 지속성객체의 메서드 타입과 메서드가 참조하는 지속성객체의 속성 타입에 대한 비밀등급을 타입접근권한목록객체로부터 확인하고 적절한 접근제어를 수행한다.

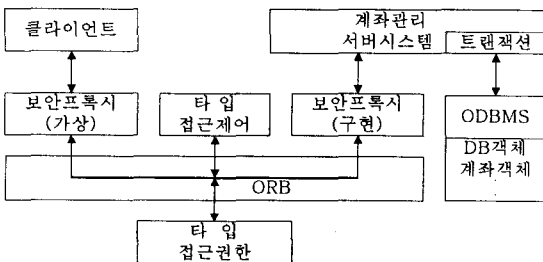


그림 17. 지속성 보안시스템의 응용구조  
Fig. 17 The application architecture of persistent security system

4.3 지속성객체의 응용보안 모델의 검증

타입기반 지속성보안시스템 응용 모델의 실용성

을 확인하기 위한 검증방식은 코바시스템의 메서드기반 접근제어 모델과 객체DB시스템의 속성기반 접근제어모델 그리고 타입기반 접근제어모델을 적용한 은행의 계좌관리시스템을 구현하여 비교 분석하였다. 그림 18은 타입기반의 지속성보안시스템에 대한 실용성 평가 모델을 보여주고 있다. 코바시스템과 객체DB시스템의 접근제어 방식은 서버시스템의 메모리상에 실체화된 지속성객체를 대상으로 접근제어를 수행하는 반면 타입기반 접근제어 기법은 지속성객체의 적재를 통한 인스턴스 수행 이전단계에서 타입을 대상으로 접근제어를 수행한다. 따라서 접근제어를 신속하게 수행할 수 있고 불필요한 통신량을 감소시키는 이점을 제공한다.

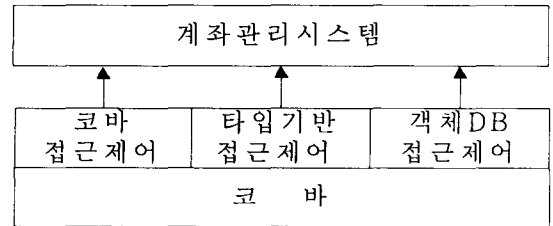


그림 18. 실용성 평가 모델  
Fig. 18 The evaluation model of practicality

타입기반의 지속성 보안 시스템의 응용 모델의 실용성을 확인하기 위한 모의실험은 쉘러론 850MHz와 윈도우즈 2000 운영체제 환경하에서 Iona사의 Orbix 코바시스템과 ObjectStore사의 객체DB시스템을 사용하여 표 1과 같은 사용자의 식별자와 소속그룹 그리고 인가등급을 갖는 보안토큰과 표 2와 같은 타입접근권한목록을 사용하여 응답 메시지를 얻는데 소요되는 시간을 측정하여 실험하였다.

표 1. 사용자의 보안토큰  
Table. 1 A security token of user

| 식별자  | 그룹        | 인가등급 |
|------|-----------|------|
| kys  | manager   | 2    |
| choi | executive | 1    |
| .    | .         | .    |

표 2. 지속성객체의 접근권한목록  
Table. 2 An access right list of persistent object

| 구분  | 계좌타입       | 보안등급 |
|-----|------------|------|
| 속성  | amount     | 3    |
| 메서드 | view()     | 5    |
|     | deposit()  | 3    |
|     | withdraw() | 3    |

모의실험을 통하여 얻은 결과값은 표 3과 같고 클래스라는 타입 정보를 이용하는 접근제어 방식이 접근처리에 소요되는 시간이 다소 우수하다는 것을 알 수 있다. 이는 메서드기반 접근제어 방식과 속성기반 접근제어 방식이 DB로부터 지속성객체를 적재하여 인스턴스화 한후에 수행되기 때문에 접근처리 수행 속도의 면에서 타입기반 접근제어의 접근처리보다 시간이 더 소요되었음을 확인할 수 있다. 또한 인스턴스의 개수의 증가에 따른 접근처리속도는 클래스 타입 정보를 이용하는 접근제어 방식에는 큰 영향이 없음을 확인할 수 있었다. 이는 동일 클래스로부터 생성된 인스턴스의 경우에는 갯수와는 독립적으로 단일 클래스 타입을 대상으로 접근제어를 수행하기 때문이다. 그리고 타입기반 접근제어방식은 코바시스템을 통하여 코바의 보안시스템 과 객체DB시스템 그리고 응용시스템과 분리되는 형태로 구현해서 통합하여 사용될 수 있다.

표 3. 접근제어 처리시간  
Table. 3 Throughout time of access control

| 인스턴스 갯수 | 메서드기반 접근제어 | 속성기반 접근제어 | 타입기반 접근제어 |
|---------|------------|-----------|-----------|
| 100     | 0.0353     | 0.0406    | 0.0030    |
| 500     | 0.0640     | 0.0765    | 0.0031    |
| 1000    | 0.0940     | 0.1020    | 0.0032    |

### V. 결 론

인터넷을 인프라로 사용하는 통합 분산환경에서

가용성을 위한 분산정보에 대한 허가되지 않는 불법적인 접근은 기업시스템의 보안성을 훼손되고 경쟁력의 상실을 가져온다. 따라서 보안성과 가용성을 동시에 고려하는 접근제어시스템으로 타입정보를 사용한 지속성객체의 보안 모델을 제안하였다. 제안 모델은 통합분산시스템에 독립적으로 구축되어 사용될 수 있고 코바시스템이 제공하는 메서드기반 접근제어 보안모델 그리고 객체DB시스템의 보안모델인 속성기반 접근제어모델과 분리 통합되는 형태로 구축되어 사용될 수 있다. 코바의 서버시스템과 객체DB시스템이 수행하는 접근제어 방식은 방대한 양의 지속성객체를 갖는 데이터베이스를 대상으로 클라이언트가 업무처리를 수행하는 경우에 서버의 부하를 야기하고 응답시간의 지연으로 인한 서비스 정지에 대한 문제를 야기할 수 있으므로 효과적이 못하다. 이는 접근제어를 수행하기 위하여 지속성객체가 프로그램 메모리상에 활성화되어 있는 인스턴스를 필요로 하기 때문에 파생되는 문제로서 전송부하와 접근처리의 지연에 따른 정보의 가용성을 제한한다. 본 논문에서 제안하고 있는 타입정보를 이용한 지속성객체의 접근제어 모델은 코바와 객체DB의 보안 모델로부터 분리 확장되는 형태로 접근제어 기능을 제공함으로써 개발구조를 개선하고 접근제어 모듈의 재사용을 향상시킬 수 있다. 또한 급속하게 늘어나고 있는 불법적인 침해사고에서 기업의 핵심적인 업무데이터를 캡슐화하고 있는 지속성객체를 보호함으로써 시스템의 신뢰성과 안정성을 높여 줄 수 있을 것으로 기대된다. 향후 타입접근권한목록으로 표현되는 클래스타입에 대한 기밀등급을 보안정책 DB와 통합하여 사용할 수 있는 자동화 도구에 대한 연구가 필요하다.

### 참고문헌

- [1] Cuppens, F. and A. Gabillon "A logical approach to model a multilevel object oriented database," in Database Security, Chapman and Hall, London, pp.145-166, 1997.
- [2] Reddy M., ORBs & ODBMS : "Two complementary ways to distribute objects," Object Magazine, pp. 24-30, June, 1995.
- [3] Karjoth, G., "Authorization in CORBA Security," In Proceedings of Fifth European Symposium on Research in Computer Security, pp. 143-158, 1998.



[4] Blakley, B., R. Blakley and R.M. Soley, CORBA Security: An Introduction to Safe Computing with Objects, Addison-Wesley, 2000.

[5] Byrne R, M. Roantree, "An Object Architecture for ODMG Database," Proceeding of the 34th International HICSS Conference, IEEE Computer Press 2001.

[6] Elisa, B., et al., "An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning," ACM Transactions on Database Systems, 23(3), 1998.

[7] Joshi, J.B.D. et al., "Security Models for Web-based Applications," Communications of the ACM, 2. pp. 44-52, 2001.

[8] Hale, J., J. Threet, and S. Sheno, Capability-based primitives for access control in object-oriented systems, in Database Security, Chapman and Hall, London, pp. 134-150, 1998.

[9] Pernul, G., A. M. Tjoa and W. Winiwarer, Modelling Data Secrecy And Integrity, Data & Knowledge Engineering, Vol. 26, pp. 291-308, 1998.

[10] Joon P., and S. Ravi, "RBAC on the web by smart certificates." In Proceedings of 4th ACM Workshop on Role-Based Access Control. ACM, Fairfax, VA, October pp. 28-29 1999.

[11] Hale, J., J. Threet, S. Sheno, "A framework for high assurance security of distributed objects," in Database Security, Chapman and Hall, London, pp.99-115, 1997.

[12] Evered, M., "A Two-Level Architecture for Semantic Protection of Persistent Distribu-

ted Objects," Proc, Intl. Conf. on Software Methods and Tools, Heidelberg 2000.

저자소개

김영수(Young Soo Kim)



1989년 2월 전북대학교 회계학과 졸업 (경영학사)  
 1992년 2월 경희대학교 경영학과 졸업(경영학석사)  
 2003년 8월 국민대학교 정보관리학과 졸업 (정보관리학박사)

※주관심분야: 인터넷 응용, 분산보안시스템

최흥식(Heung Sik Choi)



1983년 2월 한양대학교 산업공학과 졸업  
 1985년 2월 한국과학기술원 경영과학과 석사  
 1991년 2월 University of Rochester, 경영학석사

1995년 2월 University of Rochester, Computers and Information Systems, 경영학박사  
 1985년 3월 ~ 1988년 6월 데이콤정보통신연구소 연구원  
 1995년 3월 ~ 현재 국민대학교 비즈니스IT학부 교수

※주관심분야: 통신경영, 통신정책전략, 네트워크 설계