通信情報保護學會論文誌
第 8 卷 第 3 號, 1998. 9

객체 모델링 기법을 이용한 다단계 보안 데이터 모델의 설계와 구현 방안

심 갑식*

The Design and Implementation Methodology of Multilevel Secure Data Model Using Object Modelling Technique

Gabsig Sim*

요 약

본 논문은 객체 모델링 기법을 이용하여 다단계 보안 데이터베이스 응용에 대한 구조적 특징을 표현하기 위한 모델을 제시한다. 즉, 응용 영역에 대한 데이터와 보안 의미를 통합한다. 이는 응용 영역의 데이터에 대한 불법적 유출이나 수정을 방지하는 도구가 된다. 개발한 도구를 기초로 한 구현 모델에서는 다단계 데이터베이스를 단일 보안등급 데이터베이스들로 분해한다. 인스턴스뿐만 아니라 스키마도 보호하며 속성값 다중 인스턴스화 기법을 이용하여 커버 스토리를 표현한다. 그리고 그 모델에서의 생성, 검색, 삭제, 그리고 갱신과 같은 연산 의미를 설명한다.

Abstract

This paper describes structure aspects for representing multilevel secure applications. The structure aspects represent all of the entities of the applications, their associated security levels, and the relationships between them. The implementation model is based on the decomposition of a multilevel objected oriented database into a collection of single level databases. That model can protect instances and the database schema, represent cover story using the attribute polyinstantiaion, and describe the operational semantics.

Keywords : class, association, generalization, aggregation, and security level.

이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음
* 진주산업대학교 교양과정부

1. 서 론

다단계 데이터베이스 응용에서는 모든 개체가 보안등급을 갖을 수 있다. 보안등급들의 집합은 부분순서 격자(U⟨C⟨S⟨TS)를 형성한다. 여기서 기호(⟨, ⟩, ≤ 그리고 ≥)는 격자 모델에서 지배(dominate) 관계를 나타내는데 사용된다. 다단계 데이터베이스 응용을 설계하는 것은 복잡한 과정이다. 우선 개체, 그와 관련된 보안등급은 적절한 모델을 사용하여 모호성이 없이 표현되어야 한다. 정확하고 완전하게 의미를 파악하는 것이 중요하다. 그런 후 보안등급화된 정보를 참조할 때 나올 수 있는 잘못과 비권한 정보의 추론을 찾아내기 위해서 요구사항을 분석해야 한다. 그러므로 설계 도구로써 보안등급 안내서와 응용의 사양을 활용할 수 있다. 설계자는 어떤 잠재적 보안 문제점도 알고 있어야 한다. 그런 도구를 성공적으로 설계하기 위해서 강력한 모델링과 방법론이 필요하다.

일반 데이터베이스 응용을 설계하기 위한 객체 모델링 기법[RUMB91]이 있다. 객체 모델링 기법은 정보 시스템의 복잡한 응용에 대한 모델과 논법을 위해 특별히 개발되었으며, 데이터베이스 시스템, 행정/기업 정보 시스템, 의료 정보 시스템, 하이퍼미디어 시스템 등의 응용 설계를 위한 방법론의 하나이다.

객체 모델링 기법은 소프트웨어 공학 방법론이며 객체 모델, 동적 모델, 함수 모델로 구성된다. 객체 모델은 응용에 있는 객체와 그들 관련성의 정적 구조를 서술한다. 동적 모델은 시간 변화에 따른 응용의 특징을 서술한다. 이것은 제어 특징을 명세하고 구현하는데 사용된다. 함수 모델은 응용 내에서 데이터 값 변환을 서술한다. 객체 모델 기법은 응용 요구사항이 분석되는 분석 단계, 데이터베이스와 시스템 프로세스가 만들어지는 시스템 설계 단계, 알고리즘과 인터페이스가 만들어지는 객체 설계 단계로 구성된다.

본 논문에서는 다단계 데이터베이스 응용을 위한 다단계 객체 모델링 기법[SELL93]의 분석 단계 중에서 다단계 객체 모델과 유사하다. 그러나 단일 상속뿐만 아니라 다중 상속은 객체 지향 모델 및 기타 다른 모델들에서 실세계를 모형화하는데 중요한 기법으로 다루어 지고 있기 때문에, 보안 관점에서 위에서 언급한 상속성 및 연관성 상속성 등을 비롯하여 다단계 데이터베이스 응용의 표현을 위한 여러 가지 정적인 특징들을 본 논문에서 다루고 있다. 즉, 보안 위반이 발생하지 않는 방법으로 다단계 데이터베이스를 설계하는 것이다.

또한, 이들의 구현 방안을 제시하고 있다. 기초적인 보안 모델은 Bell-LaPadula 패러다임(paradigm)[BELP76]으로 표현된다. 이 패러다임은 주체(subject)와 객체(object) 개념에 기초한다. 객체는 정보를 저장하고 있는 수동 개체(passive entity)이며 분류등급(classification)이 할당된다. 주체는 객체를 접근하는 능동 개체(active entity)이며 인가등급(clearance)이 할당된다. 분류등급과 인가등급을 함께 언급할 때는 보안등급(security level)이라고 한다. 전에 언급한 바와 같이 분류등급은 부분 순서이다. 객체의 보안등급이 주체의 보안등급 이하일 때만 주체는 객체를 판독(read)할 수 있다(상향 판독 금지; no read up). 객체의 보안등급이 주체의 보안등급 이상일 때만 주체는 객체를 기록(write)할 수 있다(하향 기록 금지; no write down).

구현 방법으로는 다단계 데이터베이스를 단일 보안등급의 데이터베이스들로 분해하는 것이다. 목적은 나중에 물리적으로 데이터베이스에 저장될 단일 보안등급의 베이스 릴레이션을 얻는 것이다. 객체(인스턴스)뿐만 아니라 데이터베이스 스키마도 보호되며 다중 인스턴스화 기법을 사용하여 커버 스토리(cover story)를 지원한다.

본 논문의 나머지 구성은 다음과 같다. 2장에서는 다단계 객체 모델을 제시하며, 3장에서는 구현 모델을 제안한다. 여기서는 객체의 생성, 검색, 삭제, 그리고 갱신 연산을 설명할 것이다. 그리고 4장에서는 관련 연구들과의 비교를 하며, 마지막 5장에서는 결론과 추후 과제를 서술한다.

2. 다단계 객체 모델

2.1 객체, 클래스 그리고 속성

다단계 객체 모델의 목적은 객체(object), 성질(property), 객체간의 연관성(association), 관련성(relationship) 그리고 그들과 관계된 보안등급(security level)을 정의하는 것이다. 객체는 데이터 내용에 대한 논리적 레코드(record)나 튜플(tuple) 개념과 유사하다. 각 객체들은 객체의 존재 등급이라는 보안 등급과 연관된다. 즉, 만일 한 객체의 보안등급이 L이라면, 그의 객체 식별자가 L이다. 이것은 그 객체의 존재를 등급 L 이상에서 알 수 있다는 의미이다.

객체 식별자(object identifier)의 분류등급(classification)은 인가되지 않는 주체에게 객체의 존재를 숨기는데 사용된다. 만일, O1의 보안등급이 U 이면, 이것은 데이터베이스에 O1이 생성되었다는 사실을 모든 사람이 알 수 있다는 뜻이다. 그러나 O2의 보안등급이 C 이면, U 등급인 주체는 O2의 존재를 알 수 없다. 객체 식별자의 분류등급은 객체가 생성된 보안등급을 나타낸다.

대부분의 객체 모델에서처럼 유사한 성질(속성)을 갖는 객체들은 클래스(class)로 묶을 수 있다. 인스턴스(instance)라는 클래스의 객체는 이런 성질들을 상속받는다.

그렇다면, 클래스에도 보안 등급을 부여해야 하는가? 객체에 보안 등급이 있으므로, 객체들을 그루핑한 클래스도 역시 보안 등급이 있는 것이 타당하다. 이 등급은 클래스의 존재 등급이다. 객체의 등급과 그 클래스의 등급 사이의 관련성은 추론에 의한 보안 위반이 없어야 한다. 예를 들면, 한 객체(O)의 등급이 L1, 그 클래스(C)의 등급이 L2, 그리고 $L1 \leq L2$ 일 때, L1에서는 클래스에 대한 정보를 추론할 수 있다. 다시 설명한다면, 객체 O가 클래스 C의 인스턴스라는 사실을 L1에서 알 수 없다. 이것은 객체가 클래스의 성질을 상속하지 않았다는 것을 의미한다. 그러므로, $L1 \geq L2$로 하는 것이 더 안전하다.

모델은 클래스에 명시되어 있는 객체들의 성질을 나타내어야 한다. 이 성질들을 속성(attribute)이라고 한다. 각 클래스에는 ID라는 속성이 있다. 이것은 클래스에 속한 인스턴스의 식별자이다. 그러나 모델에서는 ID를 명시적으로 나타내지 않을 것이다. 클래스의 등급과 그 속성의 등급 사이의 관련성은 어떻게 해야 할 것인가? 더 높은 등급에서는 한 클래스에 대한 부가적인 속성에 대한 정보를 가질 수 있다. 또한, 만일 속성 등급 L1이 클래스 등급 L2 이하라면, 등급 L1에서 클래스의 존재를 추론할 수 있다. 그러므로, $L2 \leq L1$로 하는 것이 더 안전하다. 속성 등급은 속성의 존재 등급을 의미하며, 속성 ID의 보안등급이 클래스의 보안등급이다.

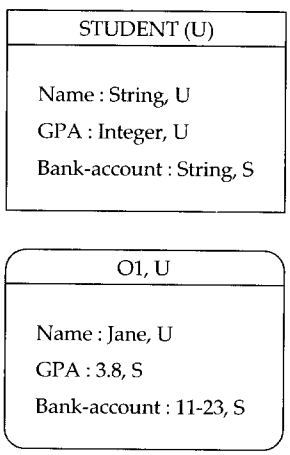
속성 등급과 그 값 등급 사이의 관련은 무엇인가? A가 클래스 C의 속성이라는 것을 안다고 해서 속성값(attribute value)을 판독할 수 있다는 뜻은 아니다. 그러므로, 값의 등급은 속성 등급 이상이다. 그렇지 않으면, 값을 판독하여 속성이 존재한다는 것을 추론할 수 있다.

속성값 보안등급은 객체 식별자와 속성간의 연관 정보를 분류하는 것이다. 예를 들어, O1의 속성 Name 값이 U 등급이다는 의미는 "O1의 이름이 Jane이다"라는 사실이 U이다는

정보를 뜻한다. 각 속성값은 독립적으로 보안 등급이 부여된다. 이런 원리는 모델에서 다단계 개체를 표현할 수 있게 한다.

객체의 속성값 보안등급의 최대하계(greatest lower bound) 이하인 분류등급을 객체 식별자에 할당한다. 이런 무결성 제약조건에 따라, 주체는 먼저 객체의 존재를 볼 수 있는 권한을 받은 후에, 그 객체의 속성 값들 중 하나를 접근할 수 있는 권한이 있다.

지금까지의 설명을 그림표로 나타내면 그림 1과 같다. STUDENT 클래스는 U 등급 이상의 인스턴스를 갖을 수 있다. U 등급 속성은 Name, GPA이며, Bank-account는 S 등급 속성이다. 그리고 O1이라는 인스턴스는 STUDENT 속성을 상속받으므로, U와 C 등급에서는 O1이 속성 Name, GPA가 있다는 것을 알 수 있고 S와 TS에서는 모든 속성을 알 수 있다. 인스턴스 속성값의 등급에는 객체 존재 분류등급이나 어떤 보안 제약조건(데이터에 보안 등급을 할당하는 규칙)에 따른다. U 등급 인스턴스는 객체 식별자와 Name에 대해서 U값을 갖으며, GPA와 Bank-account는 S값을 갖는다.

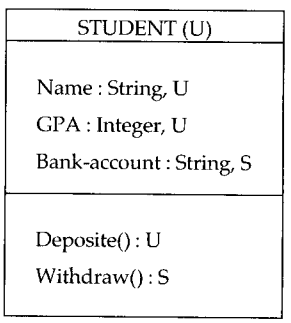


[그림 1] 클래스와 객체

2.2 연산과 메소드

연산은 클래스의 객체에 적용될 변환이다. 한 클래스의 연산 구현은 메소드(method)로 한다. 대부분의 객체 모델에서 적절한 메소드를 수행시킬 메시지(message)를 교환함으로써 객체들 간에 상호작용을 한다. 이 때 메소드에 어떤 보안등급을 할당할 것인가? 메소드의 보안등급과 클래스의 보안등급 사이에 어떤 관련성이 있는가? 메소드는 속성의 경우 처럼 클래스에서 정의되기 때문에 각 메소드에 보안등급을 할당하고 이 보안등급은 클래스의 보안등급 이상이어야 한다. 즉, "m이 클래스 c의 메소드이다"라는 사실의 기밀도(sensitivity)는 "c가 클래스이다"라는 사실의 기밀도 이상이다.

그림 2는 STUDENT 클래스의 메소드를 나타내고 있다. 여기서 Deposite 메소드는 U 등급이고 Withdraw 메소드는 S 등급이다.



[그림 2] 메소드

2.3 제약조건

무결성 제약조건(integrity constraint)은 객체, 그들의 속성, 그리고 클래스에 대해 시행될 제약조건이며, 또한 그런 구조체들 사이의 관련성(relationship)에 대해 시행될 제약조건들이다. 예를 들어, "4학년 학생이 수강할 과목 수는 적어도 2 과목이다" 혹은 "학생의 GPA가 2.0 미만이면, 그는 과목 333을 수강할 수 없다"와 같은 것이다.

분류등급 제약조건(classification constraint)에서는 제한 사항이 객체, 클래스, 속성, 혹은 객체 성질과 그 성질값 사이의 연관성에 있을 수 있다. 보안등급의 할당은 내용(content), 문맥(context), 혹은 시간(time)에 따라 다를 수 있다.

내용 제약조건은 객체의 속성에 부여될 보안등급이 속성값 자체에 종속될 때이다. 예를 들면, U등급인 속성 Name, 그리고 봉급이 2000 달러까지는 U등급이고 2000 달러 보다 더 크면 S등급인 속성 Salary를 가지는 클래스 EMPLOYEE를 고려할 수 있다.

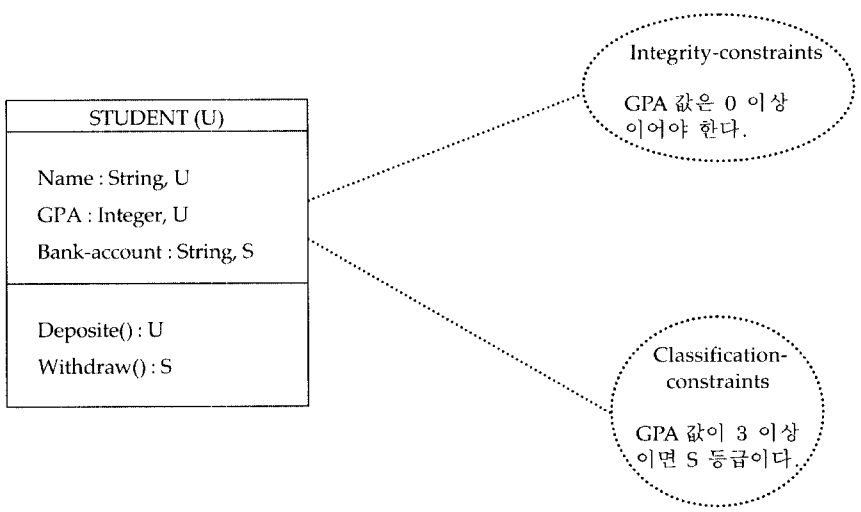
문맥 제약조건은 집단화 객체(aggregate object)들의 분류등급을 취급한다. 특히, 집단화된 정보가 이를 구성하는 각각의 정보보다 더 기밀할 때, 문맥 제약조건이 제기된다. 집단화 클래스의 보안등급을 집단화를 구성하는 각각의 클래스 보안등급 보다 더 높게 정의함으로써 이 제약조건을 표현한다. 단순 객체들로 이루어진 클래스와 그들의 연관성을 정의하는 클래스는 완전히 독립적이다. 예를 들면, EMPLOYEE에 속하는 속성 Name, Ssn, 그리고 Salary 모두 U 등급이라 하자. 그러나 한 고용인과 Salary 사이의 연관성은 S등급으로 분류한다. 문맥 제약조건은 2.6절에서 설명한 것처럼 그림표로 표현할 수 있다.

시간 제약조건은 외부 조건을 의미한다. 예를 들면, 비행 목적지는 출발 시각까지 S 등급이다. 분류등급 제약조건에 대한 더 포괄적인 설명은[김영균93][박선주93]을 참조하기 바란다.

제약조건 검사는 객체의 속성값이 변화할 때마다 자동적으로 개시된다. 각각의 분류등급 제약조건을 제한사항에 대해 자연어로 서술한다. 그림표에서 제약조건은 점선 원으로 나타내고 관련된 클래스와 점선으로 연결한다. 그림 3은 무결성 제약조건과 내용 제약조건을 표현하고 있다.

2.4 연관성

연관성(association)은 공통의 의미적 행위에 참여하는 여러 클래스들에 속하는 객체들을 서술한다. 연관성 표현은 클래스들을 연결하는 직선이며 연관성 명칭은 마름모 내에 기



[그림 3] 제약조건이 있는 클래스

입한다. 연관성은 이항, 삼항 혹은 그 이상일 수 있으며 임의의 속성을 갖을 수 있다. 또한 연관성에 참여하는 클래스들의 대응수를 다음과 같이 나타낼 수 있다.

- min . . max : 최소/최대 대응수 범위를 나타냄
- * : 영 또는 그 이상을 나타냄
- + : 일 또는 그 이상을 나타냄

보안등급을 각 연관성에 할당할 수 있는데 이 연관성의 보안등급은 연관성에 참여하는 각 클래스들의 보안등급 이상이어야 한다. 예를 들면, U 등급의 연관성에 S 등급의 클래스가 연결된다는 것은 의미가 없기 때문이다. 객체-지향 모델에서 연관성은 객체-값 속성(한 클래스에서 다른 클래스로의 포인팅)으로 보통 표현한다. 그러므로, 연관성 모델링은 클래스에 삽입될 속성으로써 할 것이다. 그리고 객체-값 속성으로 모델될 연관성을 객체 모델에서 제거할 때, 객체 모델에서 각각의 새로운 객체-값 속성에 대하여 긴 점선 화살표를 포함시킨다.

클래스 C_1(보안등급은 L_1)과 C_2(보안등급은 L_2)를 연결하는 연관성을 R(보안등급은 L_R)이라고 하고, C_1로부터 C_2로의 참조를 첨가하고자 한다면, 다음과 같이 할 수 있다. 물론 $L_1 \geq L_2$ 이어야 한다. 왜냐하면, 등급이 낮은 객체가 더 높은 객체를 참조한다는 것은 보안 유출이 일어나기 때문이다.

① 객체 모델에서 관련성 R을 제거한다.

② R이라는 새로운 속성을 클래스 C_1에 도입한다. R의 타입은 C_2(만일 클래스 C_1이 R에서 최대 대응수 일(one)일 때)이거나 *set-of*(C_2)(만일 최대 대응수가 다(many)일 때)이다.

③ 새로운 속성의 보안등급은 L_R로 한다.

④ C_1에서 C_2로 포인트하는 긴 점선을 객체 모델에 첨가한다.

2.5 일반화

일반화(generalization)는 한 클래스와 그 클래스의 정련된 버전(version) 사이의 관련성이다. 이 정련된 버전을 하위 클래스(subclass)라고 한다. 일반화에서는 IS-A 계층구조(hierarchy)라는 클래스 계층구조가 이루어진다. 클래스는 그와 연관된 여러 개의 하위 클래스들을 둘 수 있다. 예를 들면, 클래스 PERSON은 하위 클래스 STUDENT와 EMPLOYEE를 갖을 수 있다. PERSON은 STUDENT와 EMPLOYEE의 상위 클래스(superclass)라고 한다. PERSON의 성질들은 STUDENT와 EMPLOYEE로 상속된다.

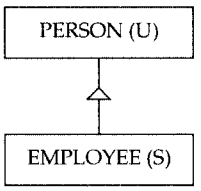
상위 클래스의 등급과 그 하위 클래스의 등급 사이의 관련성을 어떻게 할 것인가? 하위 클래스는 더 기밀한 속성들을 보호하는데 사용될 것이라고 생각된다. 그러므로, 시행해야 할 성질은 하위 클래스의 등급이 상위 클래스의 등급 이상이어야 한다. 이것은 그림 4에서 설명하고 있는데, 하위 클래스 EMPLOYEE는 S 등급인 반면에 상위 클래스 PERSON은 U 등급이다. 그렇지 않으면, 하위 클래스에 관한 정보로부터 상위 클래스에 관한 더 기밀한 정보를 추론할 수 있다.

상위 클래스에 있는 모든 것은 하위 클래스로 상속된다. 그러나 상위 클래스의 속성 보안등급이 하위 클래스 자신의 보안등급 보다 낮으면, 하위 클래스에 상속된 속성은 하위 클래스의 보안등급을 부여받는다. 즉, 그림 4에서 만일 EMPLOYEE가 PERSON의 Name 속성을 상속받는다면, EMPLOYEE의 Name 속성은 S 등급이다.

하위 클래스가 여러 개의 상위 클래스를 가지는 다중 상속(multiple inheritance)의 경우에는, 명칭 충돌(name conflict)이 발생할 수 있다. 이를 해결하기 위한 다중 상속 보안 성질은 두 가지로 나눌 수 있다. 첫째는 상속될 속

성 도메인이 동일할 경우인데, 두 개 이상의 상위 클래스가 동일 속성을 가질 때, 하위 클래스에는 가장 낮은 속성을 상속받는다. 이것은[LUNT91]의 방법과 유사하다. 예를 들어 WALKING-SHOE가 ATHLETIC-SHOE 클래스와 ORTHOPEDIC-SHOE 클래스라는 두 클래스의 하위 클래스라고 하자. 그리고 ATHLETIC-SHOE 클래스는 U등급 속성인 Size가 있고 ORTHOPEDIC-SHOE 클래스는 C등급 속성인 Size가 있다고 할 때, WALKING-SHOE 클래스는 ATHLETIC-SHOE 클래스에 있는 U등급 속성인 Size를 상속받는다. 그렇지 않으면, WALKING-SHOE 클래스가 ATHLETIC-SHOE 클래스와 ORTHOPEDIC-SHOE 클래스의 하위 클래스라는 사실을 아는 비권한 사용자는 WALKING-SHOE 클래스의 속성인 Size를 가지지 않는다는 것을 관찰함으로써 WALKING-SHOE 클래스가 C등급 속성인 Size를 갖는다는 것을 추론할 수 있기 때문이다.

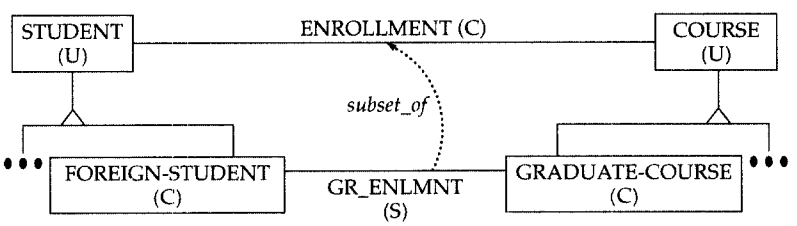
두 번째 방법은 상속할 동일 명칭의 속성들에 대한 도메인이 다를 경우인데, 상속한 상위 클래스의 명칭을 상속받은 속성 명칭에 덧붙혀서 명시적으로 모호성을 피하는 것이다.



[그림 4] 일반화

연관성을 클래스로 생각하고 연관성 클래스의 속성들은 대응하는 하위 클래스들의 연관성으로 상속될 수 있다. 또한 하위 연관성의 보안등급은 상위 연관성의 보안등급 이상이어야 한다. 이는 일반 상속성 원리와 같은 이유이다.

예를 들면, 그림 5는 STUDENT와 COURSE 사이의 ENROLLMENT 연관성, FOREIGN-STUDENT와 GRADUATE-COURSE 사이의 GR-ENLMNT 연관성을 보여주는 대학교 데이터베이스의 일부분을 나타내고 있다. 여기서 subset_of는 연관성 사이의 제약조건을 나타낸다.



[그림 5] 연관성 상속성

2.6 집단화

집단화(aggregation)는 클래스를 구성하고 있는 부품(part)들에 대한 서술이다. 집단화 개념은 하위 클래스와 그 상위 클래스 사이의 IS-PART-OF 관련성을 정의한다. 그리고 어떤 데이터 모델에서는 합성 객체(composition object)라고도 한다[BERT91, HURS93].

클래스 C가 클래스 C1, C2, ..., Cn 클래스들의 집단화이라면, C의 각 인스턴스는 C1, C2, ..., Cn 인스턴스들의 집단화이다. 예를 들면,

AUTOMOBILE 클래스가 ENGINE, CHASSIS 그리고 WHEELS 클래스들의 집단화일 수 있다. 집단화에서의 보안 성질은 보면, 집단화 클래스 C의 보안등급은 부품 클래스 C1, C2, ..., Cn 각각의 보안등급들의 최소 상계 이상이다.

2.7 다중 인스턴스화

속성값 다중 인스턴스화

객체의 속성 존재를 볼 수 있는 주체는 이 속성에 값을 줄 수 있다. 다중 인스턴스화(polyinstantiaion)는 데이터베이스에서 이런 사항을 시행하는 기법이다. 예를 들어, 객체 O1에서 속성 Salary의 값이 15000이라는 사실이 보안등급 L1일 때, 이 속성값에 또다른 10000를 삽입하고 이 사실에 보안등급 L2로 할 수 있다. 의미상으로 값 10000은 커버 스토리(cover story)로 간주 할 수 있다.

개체 다중 인스턴스화

[LUNT91]에서는 개체 다중 인스턴스화(entity polyinstantiation)를 설명하고 있다. 여기서는 관계형 데이터베이스에 초점을 두고 있다. 즉, 동일한 기본키(primary key) 값이지만 기본키의 보안등급이 서로다른 여러 튜플(tuple)들이 릴레이션에 포함된다는 것이다. 객체 지향 데이터베이스 맥락에서는 객체 O가 서로 다른 보안등급과 연관된다면 유사한 문제가 발생한다. 이에 대한 해석을 해 보면, 서로 다른 보안등급과 연관된 객체는 외부 세계에서 구별되는 개체라고 생각할 수 있다. 그러나 이것은 객체 식별자가 실세계 개체를 유일하게 식별하는데 사용된다는 객체 지향 모델의 개념에 위배된다.

즉, 객체 식별자는 객체 상태와 무관하며 전체 데이터베이스 내에서 유일하다. 그러므로 개체 다중 인스턴스화는 발생하지 않는다.

3. 구현 모델

다단계 객체를 직접 구현하는 것은 복잡하고 상당히 어렵기 때문에 구현 방안으로 다단계 객체 지향 데이터베이스를 단일 보안등급인 데이터베이스로 분해하는 것이다. 단일 등급인 데이터베이스는 기존의 비보호 객체 지향 데이터베이스와 매우 유사하다. 목적은 나중에 물리적으로 데이터베이스에 저장될 단일 보안등급의 베이스 릴레이션(base relation)들을 얻는 것이다. 즉, 다단계 데이터베이스를 단일 보안등급 데이터베이스로 표현한다는 것은 각 데이터베이스를 분류등급(classification) 별로 나눈다는 것이다. 이런 분해로써 사용자들은 자신의 신원인가(clearance)나 트랜잭션(transaction)을 수행할 분류등급에 따라 데이터베이스를 다르게 볼 수 있다.

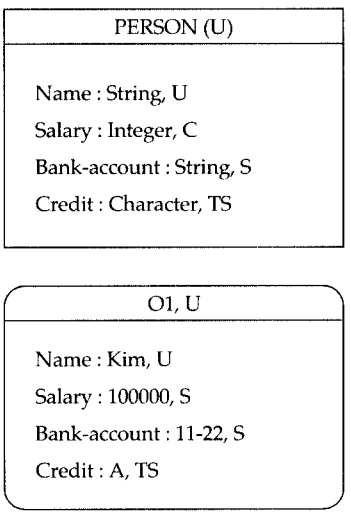
구현 모델에서 다단계 객체들과 다단계 클래스는 여러 객체 뷰와 클래스 뷰로 각각 표현된다. 객체 패러다임(paradigm)의 주요 원칙은 객체 식별자의 유일성이다. 각각의 객체 뷰와 클래스 뷰는 유일하게 식별되어야 한다. 이런 원칙을 다음과 같이 구현하기로 한다. 다단계 객체 o 혹은 다단계 클래스 c에서 등급 l인 각각의 뷰는 쌍(o, l) 혹은 (c, l)에 의해 유일하게 식별된다.

동일한 보안등급인 모든 단일 보안등급 클래스와 객체 뷰들은 단일 보안등급 객체 지향 데이터베이스에 저장된다. 각각의 단일 보안등급 데이터베이스는 비보호 객체 지향 데이터베이스처럼 관리된다. 차이점은 서로 다른 단일 보안등급 데이터베이스들 사이에 어떤 동적인 링크(link)가 형성된다는 것이다.

낮은 보안등급 뷰에 속하는 모든 클래스 속성과 메소드는 더 높은 모든 등급의 뷰들에 중복된다. 이런 중복 메카니즘을 동적으로 구현하기 위하여, 임의의 주어진 클래스의 높은

등급 뷰로부터 이와 동일한 클래스의 바로 아래 등급인 뷰로 *Isa* 링크를 생성한다. 그런 *Isa* 링크는 높은 등급의 뷰들에 있는 낮은 등급 속성과 낮은 등급 메소드를 중복되지 않게 하며, 단지 높은 등급 뷰들이 상속받게 한다.

이제 그림 6의 예제를 가지고 구현하는 방법을 알아보기로 하자.



[그림 6] 다단계 데이터베이스 생성 예

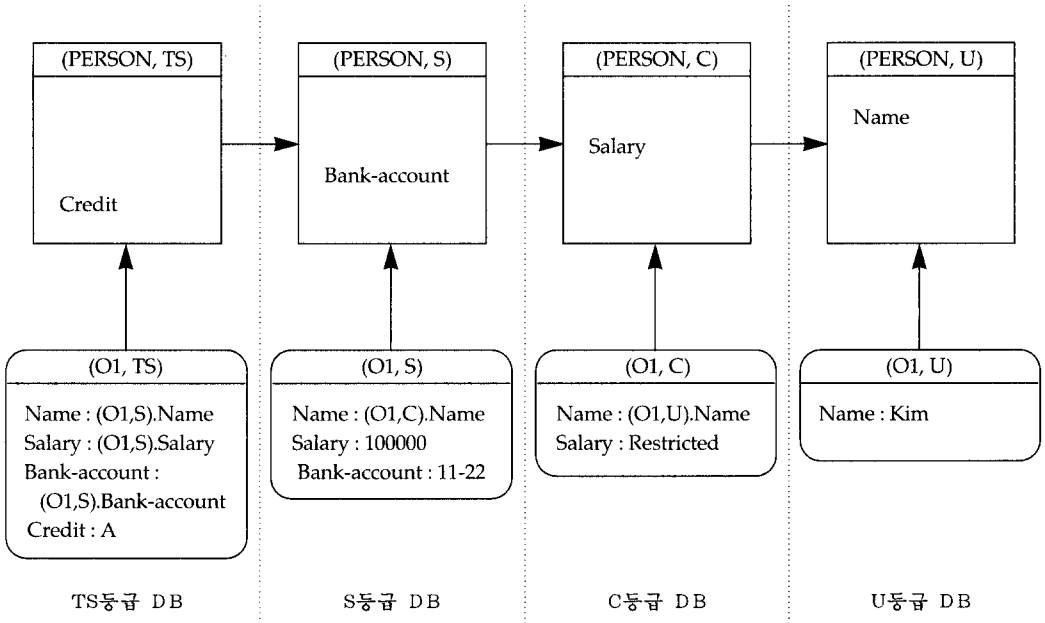
3.1 생성

사용자 A가 클래스 PERSON에서 생성하려는 객체(인스턴스)를 O1이라고 하자. A는 클래스 PERSON에 명시된 각 속성에다 O1의 값을 할당한다. 물론 이들 값들은 PERSON의 속성 타입에 속한다. 뿐만 아니라 사용자 A는 객체 식별자에 기밀도를 할당하며 각 속성 값에 분류등급을 할당한다. 객체 식별자의 기밀도는 인가되지 않은 주체에게 객체의 존재를 숨기는데 사용한다. 반면에, 속성값 분류등급은 속성과 객체 식별자간의 연관성 정보를 분류한다. 객체에 있는 속성값 분류등급들의 최대하계(great lower bound) 이하인 분류등급을 객체 식별자의 분류등급으로 한다.

예제의 생성 과정은 네 단계로 행하여진다. 이 네 단계는 U 등급, C 등급, S 등급 그리고 TS 등급 트랜잭션(transaction)에서 각각 수행된다. 그림 7은 네 단계 실행 후의 데이터베이스를 나타내고 있다.

U 등급 뷰는 U 등급 트랜잭션 내에서 생성된다. 사용자는 자신의 인가등급 이하인 임의의 보안등급에서 트랜잭션 수행을 선택할 수 있다고 가정한다. 그러면, 사용자는 예제에 대한 U 등급 뷰를 U 등급 데이터베이스에 생성할 수 있다. 클래스 뷰 (PERSON, U)는 U 등급 데이터베이스에 삽입된다. 객체 뷰 (O1, U)는 (PERSON, U)의 인스턴스로 생성된다. 클래스 PERSON의 속성 Salary, Bank-account, 그리고 Credit는 U 등급 데이터베이스에 삽입되지 않는다. 이유는 그의 존재가 각각 C 등급, S 등급 그리고 TS 등급이기 때문이다. U 등급 뷰는 자동적으로 더 높은 등급의 데이터베이스에 중복되어 더 높은 등급의 뷰를 생성한다. 즉, U 등급 트랜잭션이 완료되면, U 등급 트랜잭션은 C 등급, S 등급 그리고 TS 등급에서 각각 이 뷰를 중복하는 C 등급, S 등급, 그리고 TS 등급 프로세스를 호출한다. 이 과정은 여러 개의 상향-기록(write-up)에 의해 수행될 수 있다. 이들 상향-기록은 다단계 보안 정책에 의해 허용된다. 그러나, 실제적으로 이런 U 등급 뷰는 높은 등급의 데이터베이스에서 완전 중복되지 않는다. 클래스 속성들은 중복되지 않고 세 개의 *Isa* 링크(각 클래스들 사이의 관련성)로 인하여 단순히 상속된다. 같은 방법으로, 포인터 (O1, U).Name을 통해서 객체 뷰 (O1, C)는 객체 뷰 (O1, U)의 속성값을 직접 포인팅한다. 그리고 포인터 (O1, C).Name과 (O1, S).Name를 통해서 객체 뷰 (O1, S)와 (O1, TS)는 객체 뷰 (O1, U)의 속성값을 간접적으로 포인팅한다.

C 등급 뷰는 C 등급 트랜잭션 내에서 갱신된다. 이 C 등급 트랜잭션은 C 등급인 사용자



[그림 7] 네 개의 단일 보안등급 데이터베이스

나 C 등급에서 작업을 하기로 한 S 등급 또는 TS 등급인 사용자가 생성할 수 있다. 객체 뷰 (O1, C)는 (PERSON, C)의 인스턴스로서 생성된다. 객체 뷰 (O1, C)에서 Salary 속성값 100000이 S 등급이기 때문에 특수값 Restricted 이다. 이 의미는 C 등급 사용자들은 더 높게 분류된 봉급이 존재한다는 것을 안다는 것이다. 여기서 이 값에 다른 값을 입력한다면, 이 값은 커버 스토리가 된다. 그리고 객체 뷰 (O1, C)에서 Salary 속성값은 객체 뷰 (O1, S)에 나타낸다. 그 다음에는 C 등급 트랜잭션 완료 동안에 호출되는 S 등급 그리고 TS 등급 중복 프로세서는 S 등급 그리고 TS 등급 데이터베이스에 이런 C 등급 갱신들을 자동적으로 중복한다. 이것은 U 등급 트랜잭션 완료 동안에 호출되는 중복 프로세서와 유사하다.

S 등급 뷰는 S 등급 트랜잭션 내에서 갱신된다. 이 S 등급 트랜잭션은 S 등급 사용자나 S 등급에서 작업을 하기로 한 TS 등급 사용자가 생성할 수 있다. (O1, S)는 (PERSON, S) 클래스 뷰의 인스턴스가 된다. 그래서 속성 Bank-account는 객체 O1의 S 등급 뷰에 나타난다. 여기서도 위와 마찬가지로 TS 중복 프로세서가 호출되어 (O1, TS).bank-account는 (O1, S).Bank-account를 포인트하게 한다.

TS 뷰는 TS 트랜잭션 내에서 갱신된다. 이 TS 트랜잭션은 TS 등급인 사용자에 의해서만 생성될 수 있다. TS 트랜잭션이 끝나면 예제에 대한 생성도 완료된다.

3.2 삭제 및 갱신

검색 : 보안등급 L인 데이터베이스를 검색하려면, 먼저 사용자는 자신의 작업 보안등급을 L로 설정하고 보안등급 L인 트랜잭션을 생성해야 한다. 그런 후에 사용자는 보안등급 L인 데이터베이스를 접근할 수 있어서 L 뷰를 볼 수 있다. L 등급 스키마나 더 낮은 등급의 스키마로 연결되는 Isa 링크들을 통해 데이터베이스 스키마를 사용자는 본다. 포인터 속성

값을 평가한 후 포인트된 값을 검색할 수 있다.

갱신 : 등급 L인 뷰를 갱신할 수 있으려면, 사용자는 먼저 자신의 작업 등급을 L로 설정하고 보안등급 L인 트랜잭션을 생성해야 한다. 속성 값의 갱신은 임의의 보안 제한 없이 수행할 수 있다. 갱신될 값이 포인터인 경우, 포인터를 제거하고 등급 L인 새로운 값을 삽입한다. 속성 값에 대해 수행되는 임의의 갱신은 자동으로 더 높은 등급으로 자동으로 전파된다. 이것은 높은 등급으로부터 L 등급으로의 포인터가 있기 때문이다. 예를 들어, S 등급 사용자가 Name 값을 Smith로 갱신한다면, (O1, S).Name 값이 Smith로 갱신된다. 그러므로 (O1, U).Name 값은 커버 스토리가 된다.

삭제 : 객체를 삭제하기 위해 사용자는 먼저 자신의 작업 등급을 L 보안등급으로 설정해야 한다. L 등급인 삭제 프로세서가 객체의 L 등급 뷰를 삭제하고 연이어서 더 높은 등급인 모든 뷰들을 삭제한다. 이것은 더 높은 등급인 뷰들을 상향-기록함으로써 구현할 수 있다. 그러나 삭제될 객체와 관련된 높은 등급의 정보가 이 삭제 과정 중에 제거된다. 이는 무결성 위협이 될 수 있다.

4. 관련 연구

다단계 데이터베이스 응용을 설계하기 하기 위한 많은 접근 방법들이 있었다[PERN92, PERN93, SMIT91, SELL92, SELL93]. 그러나 이들이 제공하는 정적인 구조들을 대부분 수용하지만, 다중 상속성에서의 보안성이나 연관성에서의 보안성 및 그림 5의 연관성 상속성 표현 등은 새로운 시도이다. 또한 본 논문에서는 구현 방안을 제시하고 있다. 다시 말해서, 객체 지향 시스템에서 강제적 제어의 응용을 위해서는 객체가 단일 보안등급이어야 한다. 이 의미는 객체의 모든 속성들은 동일 보안등급이어야 한다는 것이다. 실제로 다단계 객체를 직접적으로 지원한다는 것은 매우 어렵다. 그러나 실세계 개체들은 보통 다단계이다. 즉, 동일 개체의 속성들은 서로 다른 보안등급을 가질 수 있다. 데이터베이스에서 다단계 개체를 표현할 수 없다면, 모델링 유연성이 없게 된다. 단일 보안등급 객체를 통한 다단계 개체를 모델링하는 방법들이 있었다[BERT93, JAJO90, THUR89, CUPP93].

단일 보안등급 객체 입장에서 다단계 개체들을 모델링하기 위해[JAJO90][THUR89]에서는 상속계층구조를 사용한다. 특히[JAJO90]에서는 보안 상속성(security inheritance) 개념을 도입하고 있다. 이런 접근방법들의 주된 단점은 정보를 서로 다른 보안등급으로 중복시켜야 한다는 것이다. 그러므로 서로 다른 사본들 간에 일관성을 보장하는 문제가 제기된다. [BERT93]에서는 단일 보안등급 객체를 통한 다단계 객체들을 모델링하기 위한 대안적 접근방법을 제안했다. 이 접근방법은 복합 객체를 이용한다. 특히, 더 높은 객체에 낮은 데이터를 중복하는 대신에, 더 높은 객체에 삽입된 낮은 데이터를 포함하고 있는 객체를 참조한다. 그러나 단점으로는 스키마를 수정해야 하고 메소드를 재작성해야 한다. [CUPP93]에서는 객체 레벨에만 보안등급을 할당한다. 즉, 데이터베이스 스키마는 보호되지 않아서 사용자 신원인가와 무관하게 모든 사용자는 데이터베이스의 모든 클래스를 접근할 수 있다.

국내 연구에서 MORION[이성원94]은 버전과 메소드 개념을 확장하여 객체 지향 데이터 모델에 다단계 보안을 통합했다. [조완수95]에서는 다단계 보안을 지원하는 관계 데이터 관리 시스템을 설계하기 위한 표준 관계 모델을 확장하고, 새로운 다단계 무결성 제약조건을 제시했다. [강석중95]에서는 스키마 통합과 분할을 이용해서 데이터 객체를 보호한다. 그러나 이들 연구는 응용 영역을 모델링하는 방법에 대한 초점이 아니지만, 본 연구와 상호보완적이 될 것이다.

5. 결 론

어떤 응용 영역을 데이터베이스화 할 때, 그 응용의 개체를 보안 관점에서 정확하게 분석하는 것이 중요하다. 또한, 분석된 사양은 비권한 사용자가 직·간접적으로 정보를 유출할 수 없게 해야 한다. 다시 말해서, 강제적 정책을 응용할 때는 데이터들 간의 의미적 상호관계를 통해서 정보의 어떤 불법적 흐름도 없어야 한다. 예를 들면, 하위 클래스가 상위 클래스 보다 보안등급이 낮다면, 하위 클래스에 관한 정보로부터 상위 클래스에 대한 더 기밀한 정보를 추론할 수 있다.

본 논문에서는 추론이 일어나지 않는 방법으로 다단계 데이터베이스 응용의 다양한 개체를 모델링하고 응용의 보안성 의미를 파악하기 위한 방법론을 제시했다. 특히, 다단계 데이터베이스 응용을 표현하기 위한 객체, 속성, 메소드, 클래스, 연관성, 일반화 그리고 집단화는 물론 연관성의 상속성 등과 같은 정적인 구조를 표현한다. 이는 응용 영역의 보안 데이터베이스 구축 시 설계 과정에서 저장 데이터의 불법적 유출이나 수정을 방지하기 위한 효과적인 도구가 될 것이다. 즉, 사용자 관점에서 보안 요구사항의 명세를 잘 표현한다.

또한, 개발한 도구를 바탕으로 구현 방안은 다단계 데이터베이스를 단일 보안등급의 데이터베이스들로 분해하여, 인스턴스뿐만 아니라 데이터베이스 스키마도 보호하며 속성값 다중 인스턴스화 기법을 이용하여 커버 스토리를 지원한다. 또한, 생성, 검색, 삭제, 그리고 갱신 등과 같은 연산 의미를 제시했으며, 본 논문의 연구를 바탕으로 실질적인 다단계 보안 데이터베이스 구현에 기초가 될 것이다.

앞으로의 연구 과제는 동적 성질(dynamic property)에 대한 연구와 정형화이다. 다항 연관성 및 양방향 참조에 대한 더 많은 연구가 필요하다.

참고 문헌

[BELP76] D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation, The MITRE Corp., March 1976.

[BERT91] E. Bertino and L. Martino, "Object-oriented database management systems: Concepts and issues", Computer, vol. 24, no. 4, April 1991, pp.33-47.

[BERT93] E. Bertino and S. Jajodia, "Modelling Multilevel Entities Using Single-level Objects", In Proceedings of the Third Conference on Deductive and Object-Oriented Databases volume 760 of Lecture Notes in Artificial Intelligence, Springer Verlag, 1993.

[CUPP93] N. B. Cuppens, F. Cuppens, A. Gabillon and K. Yazdanian, "MultiView Model for Object-Oriented Database", Proceedings 9th Annual Computer Security Applications Conference, 1993, pp.222-231.

[HURS93] A. R. Hurson and S. H. Pakzad, "Object-oriented database management systems: Evolution and performance issues", Computer, vol. 26, no. 2, February 1993, pp.48-60.

[JAJO90] S. Jajodia and B. Kogan, "Integrating an Object-Oriented Data Model with Multilevel Security", IEEE Symposium on Research in Security and Privacy, Oakland, 1990, pp.76-85.

[LUNT91] T. F. Lunt, "Polyinstantiation: an inevitable part of a multilevel world",

Proceedings of the computer security foundations workshop Ⅳ, 1991, pp.236-238.

[PERN93] G. Penul, W. Winiwarter and A M. Tjoa, "The Entity-Relationship Model for Multilevel Security", In Entity-Relationship Approach - ER '93: 12th Interational Conference on the Entity-Relationship Approach, Springer Verlag, 1994, pp.166-177.

[PERN92] G. Pernul, "Security Constraint Processing During Multilevel Secure Database Design", Proceedings of the 8th Computer Security Applications Conference, 1992, pp.75-84.

[RUMB91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, 1991.

[SMIT91] G. W. Smith, "Modelling Security-Relevant Data Semantics", IEEE Transactions on Software Engineering, vol.17, no.11, Nov. 1991, pp.1195-1203.

[SELL92] P. Sell, "The SPEAR Data Design Methodology", DATABASE SECURITY, Ⅵ Status and Prospects, North-Holland, 1993, pp.55-72.

[SELL93] P. Sell and B. M. Thuriasingham, "Applying OMT for Designing Multilevel Database Applications", DATABASE SECURITY, Ⅶ Status and Prospects, North-Holland, 1994, pp.41-64.

[THUR89] M. B. Thuraisingham, "Mandatory Security in Object-Oriented Database Systems", Proceedings Conference on Object-Oriented Programming: Systems, Languages, and Applications(OOPSLA), 1989, pp.203-210.

[김영균93] 김영균, 노봉남, 장옥배, "보안 객체 지식 모델에서 보안 제약조건", 통신정보보호학회논문지, 제3권 제1호, 1993. 6, pp.48-657.

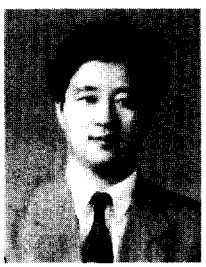
[박선주93] 박선주, 노봉남, "다단계 보안 멀티미디어 데이터 모델을 위한 보안 제약조건", 통신정보보호학회논문지, 제3권 제2호, 1993. 12, pp.16-30.

[이상원94] 이상원, 김형주, "객체지향 데이터모델에서 다중다례화를 위한 버전개념 확장", 정보과학회논문지, 제21권 제2호, 1994. 2, pp.238-247.

[강석준95] 강석준, 김용원, 황종선, "객체의 분할과 통합에 의한 스키마 기반 데이터베이스 보안 모델", 통신정보보호학회논문지, 제5권 제1호, 1995. 3, pp.51-64.

[조완수95] 조완수, "다단계 보안을 위한 관계 데이터 모델의 확장", 통신정보보호학회논문지, 제5권 제3호, 1995. 12, pp.61-68.

□ 著者紹介



심 갑 식

1985년 2월 전남대학교 계산통계학과(학사)
1987년 2월 전남대학교 대학원 계산통계학과(석사)
1993년 8월 전남대학교 대학원 전산통계학과(박사)
1993년 11월 - 현재 진주산업대학교 교양과정부 조교수

※ 주관심분야 : database security, data warehousing and mining