객체-관련성 모델을 이용한 보안 관계 스키마 정의

○
김 영 균, 이 영 록, 노 봉 남

전남대학교 전산학과

Definition of Secure Relational Schema using an Object-Relationship Model

Y.K. KIM, Y.R. LEE, B.N. NOH

Dept. of Computer Science
Chonnam National University

요 약

개념적 데이타 모델의 스키마를 논리적 스키마의 하나인 관계 스키마로의 변환 과정은 개체 무결성과 참조 무결성을 보장하여야 한다. 또한 안전한 데이타베이스를 설계하고 구현하기 위해서는 보안성을 기존의 데이타 모델에 첨가시켜서 변환 과정에 보안성을 보장하기 위한 변환 규칙을 정의하여야 한다. 이 과정을 일관성있고 효율적으로 하기 위해 변환 과정을 자동적으로 수행하는 도구가 필요하다.

본 논문에서는 보안 객체-관련성 모델을 이용하여 보안 관계 스키마로 변환하기 위한 변환 규칙을 정의하고, 변환 처리 과정을 자동적으로 수행하는 자동화 도구를 설계 및 구현하였다.

1. 서론

개념적인 데이타베이스 설계의 목적은 특정한 DBMS에 독립적인 데이타베이스에 대해 개념적 스키마(conceptual schema)를 만들어 내는 것이다. 이 과정에서는 Chen의 ER 모델과 같은 고수준 데이타 모델을 사용한다[Theo86]. 개념적 설계가 끝난 후 논리적인 데이타베이스를 설계하는데, 이 과정에서는 데이타 모델의 변환이 수행된다. 즉, 각 사용자가 정의한 DBMS에 알맞게 개념적 데이타 스키마를 논리적인 스키마로 변환하는 것이다. 개념적 설계의 목적이 스키마의 완전성과 표현성의 달성인 반면에, 논리적 설계의 목표는 논리적인 모델에서 사용할 수 있는 데이타 구조나 제약 조건을 모델링하기 위한 기능을 가능한 효율적으로 사용하는 표현을 획득하는 것이다. 즉, 논리적 모델에서 사용할 수 있는 저수준 구조들의 조건에서 고수준 표현 메카니즘을 변환하는 것에 관심을 집중하는 것이다. 이 변환 과정에서는 데이타베이스의 개체 무결성(entity integrity)과 참조 무결성(referential integrity)이 보장될 수 있도록 변환 규칙들이 정의되어야 한다. 변환 과정에서는 개념적 설계 과정의 결과인 개념적 스키마를 선택한 DBMS의 데이타 모델들 즉, 계층형이나 망형 또는 관계형 스키마로의 변환을 수행한다.

이제까지의 개념적 데이타 모델을 설계하는 동안에는 주로 데이타의 무결성만을 고려 대상으로 삼고 정확한 개념적 스키마를 표현하는데 중점을 두어 왔다. 그러나 개념적 스키마는 데이타의 무결성 뿐만 아니라 보안성(security)을 함께 표현하여야 사용자 요구를 충족시키고, 더욱 안전한 데이타베이스를 구축할 수 있다. 데이타 무결성은 데이타에 대한 변화가 생겼을 때 고의적이든 부주의든 데이타가 틀리지 않게 하는 것을 보증하는 것이다. 보안성은 권한(clearance)이 주어지지 않은 사용자가 중요한 정보를 유출해 가는 것을 방지한다. 최근들어 컴퓨터 보안에 관한 보호(protection) 메카니즘의 시도가 관심거리로 대두되고 있다. 특

히, 고도의 안전성과 보안성을 유지하기 위해서는 다단계 보안(multilevel security)을 취급할 필요성이 증가되어 여러 형태의 다단계 보안 데이타베이스들이 소개되고 있다[Dwye87, Smit89,91, Gray90]. 대부분의 다단계 보안 데이타베이스의 연구는 관계형 모델에 대해 집중되어 있고, 개념적 모델에 대한 보안 연구가 조금씩 진행되고 있다.

본 논문에서는 데이타의 의미적 무결성 제약 조건에 의한 데이타베이스의 비일관성 상태를 방지하고 또한 보안성 제약 조건에 의해 데이타의 불법 유출을 차단하기 위해 보안 객체-관련성 데이타 모델을 이용한다. 생성된 개념적 보안 데이타베이스 스키마를 물리적 보안 관계형 데이타베이스를 설계하기 위한 중간 단계인 논리적인 변환 과정에 대해 연구를 수행한다. 개념적 스키마에 표현된 보안성 질을 논리적인 스키마에서도 일관되게 유지하기 위해서는 보안성을 변환하는 규칙들이 정의되어야 한다. 그래서 본 논문에서는 보안 객체-관련성 모델의 변환에 필요로하는 무결성을 유지하는 변환 규칙과 개념적 스키마에 표현된 보안 성질을 유지하는 보안 성질 변환 규칙을 각각 정의하고 개념적 보안 객체-관련성 스키마를 자동적으로 보안 관계 스키마로 변환시켜 주는 자동화 도구를 설계하고 구현하였다.

본 논문의 구성은 2장에서 관련 연구, 3장에서는 보안 객체-관련성 모델을 설명한다. 4장에서는 본 논문에서 제안한 변환에 필요한 규칙들을 정의하며 또한 변환 알고리즘을 제안한다. 그리고 마지막으로 5장에서 결론과 앞으로의 연구 방향을 제시하였다.

2. 관련 연구

이 장에서는 개념적 데이타 모델의 설계 과정동안 만들어진 개념적 스키마를 특정한 DBMS의 데이타 모델로 변환시키는 방법론에 대해서 지금까지 연구되어진 배경을 설명한다. 방법론의 종류에는 개체-관련성 모델, 확장된 개체-관련성 모델, 객체 지향 모델을 이용한 관계 스키마 변환에 대해 살펴본다.

2.1 개체-관련성 모델을 이용한 관계 스키마 변환

개체-관련성 접근 방식[Elma89]은 논리적 설계 단계에서 널리 사용되는 기법으로서 데이타베이스 생성 주기중 개념적 설계 과정동안 고수준 모델 즉, 개체-관련성 모델을 이용하여 설계된 개념적 스키마를 생성한다. 그리고 목표로 하는 DBMS의 데이타 모델인 관계 스키마로 변환시킨다. 이 방법은 개념적 데이타베이스 설계를 위해 1976년 Peter Chen에 의해 처음으로 제안되어 현재는 개념적 데이타베이스 설계 도구의 표준으로 자리를 잡아가고 있다. 개체-관련성 모델에서 사각형은 개체를 나타내고 마름모는 관련성을 나타내며 이들 사이를 실선이 연결한다. 그리고 개체의 속성은 타원형으로 표현한다.

이 방법에서는 관계 스키마로의 변환 처리 과정을 수행하기 전에 예비 변환 과정을 수행한다. 예비 변환 과정은 첫째로 외부 구별자의 제거이고 둘째로는 개념적 설계 과정에서 발생한 개념적 스키마의 복합(complex) 혹은 다중값(multivalued) 속성들을 제거하는 것이다. 일단 이 예비 변환 과정이 수행된 후에 다음과 같은 단계들을 수행한다.

(1) 스키마의 각 개체(entity)를 한개의 릴레이션(relation)으로 변환
(2) 각 관련성(relationship)을 릴레이션으로 변환

스키마의 각 개체를 한 릴레이션으로 변환하는 과정은 개체의 속성(attribute)과 주키(primary key)가 릴레이션의 속성과 주키로 변환된다. 두번째로 각 관련성의 변환은 일반적으로 이항(binary) 관련성을 의미하는데 일대일(one-to-one), 일대

다(one-to-many), 다대다(many-to-many) 그리고 다차원 이상의 관련성의 변환을 수행한다. 관련성의 변환에는 두가지의 일반적 규칙이 적용된다. 첫째, 관련성에 참여하는 개체들중에서 오직 한 개체의 대응수가 다(many)이고 존재성이 필수(mandatory)인 경우에 개체의 주키를 다른 개체의 후보키로 포함시켜서 변환을 수행한다. 둘째, 두 개체중 어느 한 개체도 대응수가 다이고 존재성이 필수이지 않으면 두 개체의 주키를 속성으로 포함하는 새로운 릴레이션으로 변환한다. 그리고 마지막 단계로서 생성된 릴레이션에 정규화(normalization) 규칙을 적용한다.

2.2 확장된 개체-관련성 모델을 이용한 관계 스키마 변환

Chen의 개체-관련성 모델에는 여러가지의 개선될 문제가 존재한다. 즉 각 개체들에 대한 하부 구조가 부족하고, 또한 일반화 계층(generalization hierarchy) 구조에 대응하는 부분이 존재하지 않는다. 일반화 계층 구조는 개체들의 구조를 더욱 세밀하게 하고, 필요할 때는 추가할 수 있는 기능을 제공한다. 또한 데이타 모델 설계자와 데이타베이스 설계 언어간의 의미적 차이를 좁혀 준다. 그래서 확장된 개체-관련성 모델은 개체 뿐만 아니라 일반화와 집단화(aggregation), 연관성(association)을 지원한다.

이 방법에서는 다음과 같은 변환 규칙들을 정의한다[Theo86].

[규칙 1] 본래의 개체와 같은 정보 내용을 갖는 개체 릴레이션으로 변환
[규칙 2] 상위 개체의 외래키를 가진 개체 릴레이션으로 변환
[규칙 3] 서로 관련있는 모든 개체들의 외래키를 가진 관련성 릴레이션으로 변환

변환 규칙의 정의가 수행된 후, 기본적인 변환 과정이 다음과 같이 수행된다.

(1) 각 개체를 개체에 포함된 키속성과 키가 아닌 속성을 가진 하나의 릴레이션으로 변환
(2) 대응수가 다대다인 모든 이항 혹은 일항 연관성을 관련성 릴레이션으로 변환
    만약, 개체들사이의 대응수가 일대다일 경우는 대응수가 다인 개체의 주키를 대응수가 일인 개체에 삽입한다. 그리고 개체들 사이의 대응수가 일대다이면 한쪽 개체의 주키를 다른쪽 개체의 속성으로 포함시킨다. 일반화나 부분 집합 계층 구조에 관여하는 모든 개체는 각각 릴레이션으로 변환되지만 이 릴레이션들 각각은 상위 개체의 주키를 포함한다. 상위 개체도 또한 관계하는 모든 개체들에 공통적인 키가 아닌 속성들을 포함한다.
(3) 모든 다대다 이항 혹은 일항 연관성을 관련성 릴레이션으로 변환
    즉, 변환된 관련성 릴레이션이 관련성의 속성들과 모든 개체들의 키들을 속성으로 포함하도록 변환시킨다.
(4) 모든 삼항 혹은 그 이상의 관련성을 관련성 릴레이션으로 변환

각 개체들이 변화된 릴레이션은 변환 과정을 수행하는 동안 정규화를 만족하게 된다. 그래서 함수 종속성이나 다중값 종속성을 이용하여 관계 릴레이션에 대한 정규화만을 수행한다[Theo86].

2.3 객체 지향 모델링 기법을 이용한 관계 스키마 변환

객체 지향 모델링 기법은 개체-관련성 모델과 확장된 개체-관련성 모델의 방법들을 개선시킨다. 즉, 객체 지향 모델링 기법은 이해도, 확장성, 성능을 향상시킨 데이타베이스를 설계하기 위한 개체-관련성 모델의 새로운 접근법이라 할 수 있다[Mich88]. 객체 지향 모델링 기법은 고수준 표현층, 중간 표현층, 저수준 표현층의 세가지 층으로 구성된다. 고수준 표현층에서는 객체, 관련성, 일반화 ,집

단화, 연관성 그리고 관련성의 적정화(qulification) 등과 같은 개념을 사용하여 개념적 스키마를 설계하고 고수준 표현층에서 설계된 개념적 스키마를 관계형 테이블로 변환한다. 마지막으로 저수준 표현층에서는 선택한 목표 데이타베이스 관리 시스템의 데이타 정의 언어를 생성한다.

객체 지향 모델링 기법의 중간 표현층에서 개념적인 스키마를 관계형 스키마로 변환하는 과정은 다음과 같다.

(1) 널값(null value)의 사용을 제어하면서 객체 클라스를 직접 하나의 릴레이션으로 변환
(2) 일반화 관련성은 하나의 상위 클래스(super class) 릴레이션과 여러개의 하위 클래스(subclass) 릴레이션으로 변환
(3) 다대다 관련성을 서로 다른 독립적 릴레이션으로 변환
(4) 이항 연관성의 변환

3. 보안 객체-관련성 모델 (Secure Object-RElationship Model : SOREM)

이 장에서는 데이타 모델에 데이타 무결성과 보안성을 함께 표현할 수 있는 보안 객체-관련성 모델을 설명한다. 보안 객체-관련성 모델은 객체 지향 모델링 기법에서 클라스, 일반화, 집단화 ,메소드(method)의 개념과, 개체-관련성 모델에서 대응수, 연관성의 개념을 그리고 관계형 모델에서 키(key)의 개념들을 차용하여 만든 모델이다. 보안 객체-관련성 모델에서는 사용할 그래픽 표기법, 보안성 성질 등을 정의한다. 모델을 표현하는 방법은 간결하고, 일반화된 기호를 사용하여 데이타 의미와 보안성 정보를 사용자와 쉽게 의사 소통할 수 있게 해준다.

3.1 모델의 기본 개념

보안 객체-관련성 모델을 구성하는 기본적인 구성요소는 객체, 클라스, 속성, 연산, 트랜잭션(transaction), 관련성 등이 있다. 모든 객체와 개념들은 객체로 표현하고 객체는 식별자에 의해서 유일하게 구별된다. 식별자는 속성의 특별한 형태이고 클라스는 추상적 객체를 나타낸다. 클라스와 클라스, 클라스와 속성의 관계를 나타내는 관련성 표현이 있다. 보안 객체-관련성 모델에서 지원하는 추상화 관련성은 연관성, 일반화 그리고 집단화 등이 있다. 여기서 일반화는 'is-a' 관련성을 나타내며, 집단화는 'a-part-of' 관련성을 나타낸다. 그리고 일반화와 집단화를 제외한 객체들 사이의 일반 관련성은 연관성으로 표현한다.

보안 객체-관련성 모델에서 연관성을 표현하는 방법은 대응수(cardinality)와 존재성(existence)의 관점에서 정의된다. 대응수는 연관성에서 객체의 사상 관계를 나타낸다. 그리고 어떤 객체 클라스가 그와 관련된 객체 클라스의 한 객체에 대해 얼마만큼 대응하는가를 보여준다. 각 객체 클라스의 대응수는 일(one) 또는 다(many)의 값을 갖는다. 존재성은 한 객체의 존재가 다른 객체의 존재에 종속되는 관계를 나타낸다. 즉, 객체의 존재 종속성을 표현하는 방법이다. 존재성은 필수(mandatory)와 선택(optional)으로 구분한다[노92].

보안 객체-관련성 모델에서 보안 등급은 U(unclassified), C(confidential), S(security), TS(top security) 등이 있는데 이들의 등급 관계는 U < C < S < TS 이다. 보안성 의미의 표현은 클라스나 속성 그리고 연관성 등에 굵은 실선으로 표현한다. 예를 들면 속성의 보안등급이 [U,S]로 표기되어 있으면 이 속성의 보안등급은 U, C, S중의 하나를 갖는다는 것을 의미한다.

4. 보안 관계 스키마 변환

이 장에서는 개념적 설계 과정동안 만들어진 보안 객체-관련성 스키마를 보안 관계 스키마로 변환하는 방법을 설명한다. 보안 관계 스키마로의 변환을 수행하기 위해서는 객체나 관련성을 변환 과정 동안 객체의 무결성과 보안성이 보장되도록 변환 규칙이 정의되어야 한다. 따라서 여기서는 이들 변환 규칙들을 정의하고, 변환 알고리즘을 제시한다. 변환 결과로는 관계형 데이타베이스의 데이타 정의 언어인 SQL을 확장한 보안 SQL을 생성한다.

4.1 보안 SQL 형식

관계형 스키마는 현실 세계를 표현하는데 있어 간단하며, 강력하고, 정형화된 모델이다. 또한 데이타베이스 설계시 발생하는 어떤 문제나 중복성, 분산 등의 데이타베이스 관리에 관계되는 많은 문제들을 이론적으로 분석할 수 있는 안정된 수학적 기초를 제공한다. 보안 SQL의 일반적인 형식은 다음과 같다.

< 보안 SQL 형식 >

```
CREATE TABLE  secure_object_class_name : [ object_class_security_level ]
                  attribute_name : type :[attribute_security_level]:
                                        [attribute_relationship_security_level]
                                .
                                .
   PRIMARY KEY  attribute_name     .
   FOREGIN KEY  attribute_name REFERENCE TABLE secure_object_class_name
```

4.2 변환 규칙

본 논문에서는 보안 관계 스키마 변환 규칙을 각 관련성의 종류에 따라서 정의한다. 즉, 일항 연관성, 이항 연관성, 삼항 연관성, 일반화나 집단화 관련성 각각에 따라 변환 규칙을 정의한다. 그리고 각 관련성에서는 참여하는 보안 객체 클라스가 갖는 대응수와 존재성의 종류에 따라 변환하는 규칙이 다르기 때문에 각각의 경우에 따라서 무결성을 보장하고 보안 성질을 일관되게 유지하는 변환 알고리즘을 제시한다.

[보안성 변환 규칙]

(1) 보안 객체 클라스의 등급과 속성들의 보안등급은 개념적 스키마에 정의된 보안등급을 그대로 유지시키며 변환한다.

(2) 일항 혹은 이항 연관성에 보안등급이 부여되었을 경우 보안성을 일관되게 유지하기 위하여 각 보안 객체 클라스의 외래키 속성의 보안등급이 연관성 보안등급에 따라서 변환되어야 한다.

(3) 삼항 연관성에서 대응수가 일대일대일, 일대일대다 그리고 다대다대다의 경우에는 보안성이 보장되기 위하여 주키의 집합을 이루는 속성들의 등급이 모두 동일해야 된다. 속성들의 보안등급이 서로 다른 경우에는 속성들 중 가장 높은 등급을 갖는 속성의 보안등급이 새로 생성되는 릴레이션의 속성들의 보안등급으로 모두 동일하게 유지되어야 한다. 그러나 연관성에 부여된 보안등급이 속성들의 보안등급보다 더 높게 부여되어 있다면 새로 생성되는 릴레이션의 모든 속성들의 보안등급이 동일하게 연관성에 부여된 보안등급을 따라야 한다. 그리고 대응수가 일대다대다인 경우에는 다쪽의 보안 객체 클라스들의 주키 등급은 동일해야 한다. 일쪽의 보안 객체 클라스의 연관성 보안등급이 같거나 높을 경우가 있으므로 새로 생성되는 릴레이션의 주키를 이루는 속성들의 보안등급은 동일하게 유지한다. 그리고 나머지 속성은 연관성에 부여된 보안등급과 같게 유지시키며 변환한다.

(4) 일반화의 보안성 변환은 상위 보안 객체 클라스의 보안등급이 하위 보안

객체 클라스의 보안등급보다 낮게 부여되어야 한다. 그래서 하위 보안 객체 클라스의 주키나 외래키로 포함되는 상위 보안 객체 클라스의 주키의 보안등급은 하위 보안 객체 클라스의 보안등급에 따라서 변환된다.

(5) 집단화의 보안성 변환은 이항 연관성에서의 경우와 비슷하게 집단화 관련성에 부여된 각각의 보안등급에 따라서 각각의 부품 보안 객체 클라스의 외래키 속성의 보안등급이 관련성 보안등급을 따른다.

4.2.1 일항 연관성

연관성에 참여하는 보안 객체 클라스의 수가 오직 하나이어야 하고, 또한 존재성은 모두 필수이거나 선택이어야 한다. 따라서 존재성, 대응수, 보안성의 종류에 따라 24가지 경우가 발생한다. 이들의 변환 알고리즘은 다음과 같다.

< 일항 변환 알고리즘 >

```
/* CaEx는 클라스가 갖는 대응수와 존재성 */
/* p_k는 클라스의 주키 */
/* f_k는 클라스의 외래키 */
/* CM는 대응수는 일이고 존재성은 필수 */
/* CO는 대응수는 일이고 존재성은 선택 */
/* NCM는 대응수는 다이고 존재성은 필수 */
/* NCO는 대응수는 다이고 존재성은 선택 */
/* L(R)는 관계성에 부여된 보안등급 */
/* level -> security_level */

Unary_Map()
{
                /* 클라스의 대응수가 모두 일인 경우 */
   if ( all CaEx of a class = CM || CO )
   {
          get(role_name);
          include(role_name);
          if ( exist L(R) )
               role_name.level := L(R);
          else
               get(p_k.level);
               role_name.level := p_k.level;
   }
                /* 클라스의 대응수가 모두 다인 경우 */
   else if ( all CaEx of a class = NCM || NCO )
        {
               get(p_k);
               get(role_name);
             /* 새로운 릴레이션 생성 */
               create_new_relation(p_k, role_name);
               if ( exist L(R) )
                   p_k.level := L(R);
                   role_name.level := L(R);
               else
                   role_name.level := p_k.level;
        }
               /* 한쪽의 대응수는 일이고, 다른쪽 대응수가 다인 경우 */
        else if ( one CaEx = CM && the other CaEx = NCM )
             {
                   get(role_name with NCM);
                   include(role_name);
                   if ( exist L(R) )
                       role_name.level := L(R);
                   else
                       role_name.level := p_k.level;
             }
}
```

4.2.2 이항 연관성

이항 연관성은 객체 인스탄스들 사이의 일반 관련성으로 본질적으로 양방향의 성질을 가지고 있다. 그리고, 연관성도 하나의 객체 클라스로 자신의 연관성 객체 속성들을 가질 수 있다. 이항 연관성의 변환은 각 보안 객체 클라스의 대응수와 존재성 그리고 보안성 변환 규칙에 따라서 수행한다. 변환 알고리즘은 다음과 같다.

< 이항 변환 알고리즘 >

```
Binary_Map()
{
                /* 한 클라스의 대응수가 다일 경우 */
   if ( one of class's CaEx = NCM || NCO )
      /* CaEx = CM || CO 인 클라스에 대해 */
        get(class.p_k);
      /* CaEx = NCM || NCO 인 클라스에 대해 */
        class.f_k := class.p_k;
        call relationship_security_map();

              /* 한쪽 클라스의 대응수가 일이면서 존재성이 필수이고
                 다른쪽 클라스의 대응수가 일이면서 존재성이 선택인 경우 */
        else if ( one class's CaEx = CM && the other class's CaEx = CO )
             /* CaEx 가 CM인 클라스에 대해 */
              get(class.p_k);
             /* CaEx 가 CO인 클라스에 대해 */
              class.f_k := class.p_k;
              call relationship_security_map();

              /* 모든 클라스의 대응수가 일이면서 존재성이 선택인 경우 */
            else if ( all classes's CaEx = CM || CO )
                 get(class1.p_k);
                 class2.f_k := class1.p_k;
                 call relationship_security_map();

              /* 모든 클라스의 대응수가 다인 경우 */
                else if ( all classes's CaEx = NCM || NCO )
                {
                       get(class1.p_k);
                       get(class2.p_k);
                    /* 새로운 릴레이션 생성 */
                      create_new_relation( class1.p_k, class2.p_k );
                      if ( exist L(R) )
                            get(L(R));
                            classes.p_k.level := L(R);
                }
}

/* 연관성에 보안등급이 부여되어 있을 경우 보안등급 변환 */
relationship_security_map()
{
  if ( exist L(R) )
        get(L(R));
        class.f_k.level := L(R);
  else
        get(p_k.level);
        class.f_k.level := p_k.level;
}
```

4.2.3 삼항 연관성

삼항 연관성의 보안 관계 스키마 변환은 연관성에 참여하는 보안 객체 클라스들의 대응수와 존재성의 종류에 관계없이 항상 연관성에 참여하는 각 보안 객체 클라스들의 주키들로 이루어진 새로운 연관성 보안 객체 클라스를 만든다. 그리고 각각 보안 객체 클라스를 독립적인 릴레이션으로 변환을 수행하고 연관성 릴레이

션에 대한 변환을 수행한다. 그러나, 보안 객체 클라스의 대응수와 존재성의 종류에 따라 주키의 범위가 제약을 받는다.

< 삼항 변환 알고리즘 >

```
/* LUB -> Least Upper Bound */

Ternary_Map()
{
    /* 세개의 클라스들 모두가 대응수가 일이거나 다인 경우 */
  if ( all of classes'CaEx = CM || NCM )
  {
     get(p_k1, p_k2, p_k3);
     create_new_relation(p_k1, p_k2, p_k3);
       /* 새로 생성된 릴레이션에 대해 */
     p_k := (p_k1, p_k2, p_k3);
     if ( exist L(R) )
         all attrs.level := L(R);
     else ( not same of all p_k.level )
          all attrs.level := LUB(p_k1, p_k2, p_k3);
  }
    /* 한 클라스만 대응수가 다이고 나머지 클라스들의 대응수는 일인 경우 */
  else if ( one CaEx = NCM && others of CaEx = CM )
       {
          get(p_k1, p_k2, p_k3);
          create_new_relation(p_k1, p_k2, p_k3);
            /* 새로 생성된 릴레이션에 대해 */
          p_k := (a pair of p_k with CM);
          if ( exist L(R) )
              p_k.level := L(R);
              attr.level := L(R);
          else ( not exist && not same of all p_k.level )
               p_k.level := LUB(p_k1, p_k2, p_k3);
               attr.level := p_k.level;
       }
    /* 한 클라스만 대응수가 일이고 나머지 클라스들의 대응수가 다인 경우 */
      else if ( one CaEx = CM && others of CaEx = NCM )
           {
              get(p_k1, p_k2, p_k3);
              create_new_relation(p_k1, p_k2, p_k3);
                 /* 새로 생성된 릴레이션에 대해 */
              p_k := (a pair of p_k with NCM);
                 /* 관계성에 부여된 보안등급이 동일할 경우 */
              if ( exist L(Ri) && (L(Ri) = L(Rj) )
                  p_k.level := L(Ri);
                  attr.level := L(Rj);
                /* 관계성에 부여된 보안등급이 같지 않을 경우 */
              else ( exist L(R) && (L(Ri) ≠ L(Rj) )
                    p_k.level := L(Ri);
                    attr.level := L(Rj);
           }
}
```

4.2.4 일반화와 집단화 관련성

일반화 관련성을 보안 관계 스키마로 변환하는 방법에는 4가지가 있으나, 그 중에서 가장 일반적인 방법인 하위 보안 객체 클라스에 상위 보안 객체 클라스의 주키를 포함시켜 각각을 독립적인 릴레이션으로 변환하고, 상위 보안 객체 클라스는 그대로 하나의 릴레이션으로 변환하는 방법으로 변환을 수행한다.

집단화 관련성의 보안 관계 스키마 변환 과정은 일반화 관련성 변환의 경우와 비슷하다. 즉, 집단 보안 객체 클라스는 그대로 릴레이션으로 변환하고, 부품 보안 객체 클라스는 집단 보안 객체 클라스의 주키를 포함시켜 하나의 릴레이션으로 변환시킨다. 보안성의 변환은 집단 보안 객체 클라스의 주키의 등급이 부품 보안 객체 클라스의 외래키로 포함될 때 관련성에 부여된 보안등급에 따라서 부품 보안 객체 클라스에 포함되는 외래키의 보안등급이 관련성 보안등급으로 변환된

다. 집단화 관련성 변환 알고리즘은 다음과 같다.

< 일반화와 집단화 변환 알고리즘 >

```
/* GLB -> Greatest Lower Bound */
get(relationship_type);
if ( relation_type = generalization )
{
        /* 상위 클라스에 대해 */
        create_relation(p_k, attrs);
        if ( exist discriminator )
            include(discriminator);
            discriminator.level := p_k.level;

        /* 하위 클라스에 대해 */
        create_relation(attrs);
        get(super_class.p_k);
        include(p_k);
        p_k.level := GLB(attri, i = 1 to n);
}
else if ( relationship_type = aggregation )
     {
            /* 집단 클라스에 대해 */
            create_relation(p_k, attrs);

            /* 부품 클라스들에 대해 */
            create_relation(p_k, attrs);
            get(aggregate_class.p_k);
            f_k := aggregate.p_k;
            if ( exist L(R) )
            {
                    get(L(R));
                    p_k.level := L(R);
                    if ( attrs.level < L(R) )
                            attrs.level := L(R);
                    else
                            reserve level of attributes;
            }
     }
```

4.3 정규화

제3정규형은 객체-관련성 모델링 방법의 고유한 장점이다. 정규형은 데이타의 무결성을 향상시킨다. 제 1정규형은 복합 객체들을 원자값으로 분해함으로써 얻을 수 있다. 그리고 처음 객체를 실세계에 존재하고 서로 구분할 수 있는 사물로 정의했고 각 객체는 구별할 수 있는 정보가 제공되어지면 유일한 키를 갖기때문에 제 2정규형을 만족한다. 대부분의 제 3정규형을 위반하는 것은 테이블에 관계없는 정보가 표현될 때 발생한다. 릴레이션 테이블은 비현실적인 구성이 가능하고 또한 설계하는 수준이 매우 낮다. 그러나 객체 패러다임은 추상화의 높은 수준에 위치하며 비현실적인 설계가 불가능하다. 그래서 객체-관련성 모델링 방법은 어느 정도의 제 3정규형을 만족한다.

5. 결론

개념적 데이타베이스 설계의 목적은 특정한 데이타베이스 관리 시스템에 독립적인 데이타베이스의 개념적 스키마를 생산한다. 이 개념적 스키마는 물리적인 데이타베이스의 설계를 위한 데이타베이스 모델에 맞는 논리적인 스키마로 바꾸어져야 한다. 관계형 스키마는 다른 스키마에 비해 더 나은 이론적 배경을 제공하며, 상업적 측면에서 주된 관심 대상의 모델이다. 그리고 현실 세계를 표현하는데 정형화되고, 단순하며, 강력한 이론적 배경을 제공한다.

개념적 스키마를 논리적 스키마인 관계 스키마로의 변환 과정은 개체 무결성과 참조 무결성을 보장하도록 변환 규칙이 정의되어야 한다. 또한 보안성질의 유지에 필요한 보안성질 변환 규칙도 정의해야 한다. 그리고 새로운 구조의 데이타베이스를 설계하고 구현하는 과정을 일관성있고 효율적으로 하기 위해 변환 과정을 자동적으로 수행해 주는 자동화 도구가 필요하다. 본 논문에서는 보안 객체-관련성 스키마를 보안 관계 스키마로 변환하기 위해서 무결성을 보장하는 변환 규칙을 정의하였다. 또한 보안성 변환에 필요한 규칙도 정의하였다. 그리고 변환 알고리즘을 제시하고, 변환 과정을 자동적으로 수행해주는 자동화 도구를 설계하고 구현하였다. 앞으로는 정적인 구조를 나타내는 스키마의 변환 뿐만 아니라 개념적 설계 단계에서 연산인 동적인 구조 모델링에 보안성 표현에 대한 연구가 필요하다.

< 참 고 문 헌 >

[Bati92] C. Batini, S. Ceri, Shamkant B. Navathe, Conceptual Database Design, pp. 3-84, pp. 309-346, The Benjamin/Cummings Publishing Co., 1992.

[Deco92] G. Decorte, A. Eiger, D Kroenke, T. Kyte, " An Object-Oriented Model for Capturing Data Semantics", 8th International Conf. on Data Engineering, Feb., 1992, pp. 126-135.

[Dwye87] P. Dwyer, G. Jelatls and B. Thuraisingham, "Multilevel Security in Database Management System", Computers and Security, Vol.6, No.3, June, 1987, pp.252-260.

[Elma89] R. Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, pp. 327-346, pp. 453-485, The Benjamin/Cummings Publishing Co., 1989.

[Gray90] Gary W. Smith, "The Semantic Data Model for Security: Representing the Security Semantics of an Application", Proc. 6th International Conf. on Data Engineering, Feb., 1990, pp. 322-329.

[Mich88] Michael R. Blaha, William J. Premerlani, " Relational Database Design using an Object-Oriented Methodology", Commun. ACM, Vol. 31, No. 4, April, 1988, pp.414-427.

[Smit89] G. Smith, "Multilevel Secure Database Design: A Practical Applica--tion", 5th Computer Security Applications Conf., Dec., 1989, pp. 314-329.

[Smit91] G. Smith, "Modeling Security-Relevant data Semantics", IEEE Trans--actions on Software Engineering, Vol.17, No.11, Nov., 1991, pp. 1195-1203.

[Theo86] T. K. Teorey, D. Yang and J. P. Fry, "A Logical Design Methodology for Relational Databases using the Extended Entity-Relationship Model", Comput. surveys, Vol. 18, No. 2, June, 1986, pp. 197-222.

[김91] 김원중, 객체 데이타 모델을 이용한 스키마와 트랜잭션 모델링에 관한 연구, 전남대학교, 박사학위 논문, 1991.

[노92] 노봉남, 심갑식, "A Secure Data Modelling for Database Design",데이타베이스학회 국제학술대회 논문집, 1992.