

보안 분산 객체지향 데이터베이스 스키마의 통합

박 우 근[†] 노 봉 남^{††}

요 약

분산 DBMS는 네트워크의 각 사이트에서 서로 다른 사용자에게 의해 독립적으로 설계, 관리, 유지보수되고 있는 지역 스키마들을 통합하여 전역 가상 스키마를 제공하며, 특정 사이트의 사용자가 다른 사이트의 지역 데이터베이스를 투명하게 이용할 수 있는 환경을 지원한다. 또한 각 지역 스키마에 부여된 스키마 구성 엔티티들의 보안성질이 통합된 스키마에서도 유지되도록 해야한다. 그러나 분산 DBMS에서 지역 스키마의 보안성질을 유지할 수 있는 통합에 대한 연구는 거의 이루어지지 않았다. 본 논문은 분산 DBMS 환경에서 각 사이트의 지역 스키마 정의를 위한 모델로서 객체지향 모델을 확장한 다단계 보안 객체지향 데이터베이스 모델을 사용하였으며, 지역 스키마를 통합하는데 있어서 본래의 보안성질을 유지할 수 있는 통합 방법을 객체클래스, 객체클래스 사이의 관계를 중심으로 각각 8가지로 구분하여 제안하였다.

Integration of Secure Distributed Object-Oriented Database Schemas

Woo Geun Park[†] and Bong Nam Noh^{††}

ABSTRACT

A distributed DBMS integrates local schemas that are independently designed, maintained, and managed by different users at each site providing a global virtual schema. This global schema supports users at a specific site to transparently utilize local databases at different sites. The security features of the local schemas should also not be changed in the global schema integrating security features of each local schema. Researches on the integration of security features into local schemas in distributed DBMS environment, however, are very rare. This paper using the multilevel secure object-oriented database model(as the model for the definition of a local schema in distributed environment) which is an extension of the object-oriented models. It also suggests eight integration methods that can maintain the security features of local schemas. The eight methods are classified by the object classes and by the relationships among them.

1. 서 론

데이터베이스 스키마의 통합은 유사한 응용 문제에 대하여 서로 다른 사용자가 생성한 스키마이거나, 새로 생성한 스키마들을 전역적이고 통합된 스키마로 통합하는 행위이다[1]. 스키마 통합은 일반적으로 (그림 1)과 같은 Batini의 4 단계 통합과정을 기본으로 하고 있으며, 데이터베이스 설계 단계에서 서로 다른 사용자에게 의해 제안된 사용자 뷰를 전역 개념 스키마로 통합하

는 뷰 통합(View Integration)(그림 2)과 분산 데이터베이스 환경에서 각 지역 데이터베이스 스키마를 전역 가상 스키마로 통합하는 데이터베이스 통합(Database Integration)(그림 3)의 두 분야에서 많은 연구가 진행되어 왔다[1, 2, 5, 6, 9, 10].

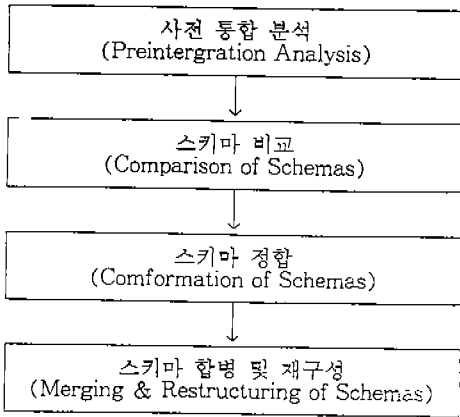
또한 분산 데이터베이스 환경에서의 데이터베이스 보안은 대단히 중요한 문제 중의 하나이다. 데이터베이스 보안의 목적은 권한이 없는 사용자의 접근을 제어하여 정보의 불법적인 접근, 고의적인 파괴 및 변경을 방지하고 우발적인 사고로부터 정보를 보호하는데 있다.

전통적인 보안 정책은 사용자에게 일정한 접근

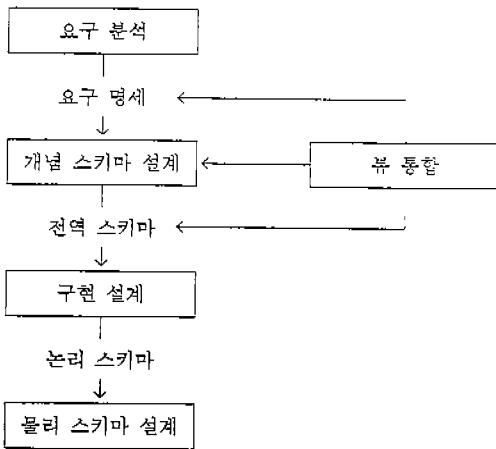
[†] 정 회 원 : 광주대학교 전자계산학과 부교수.

^{††} 종신회원 : 전남대학교 전산학과 교수.

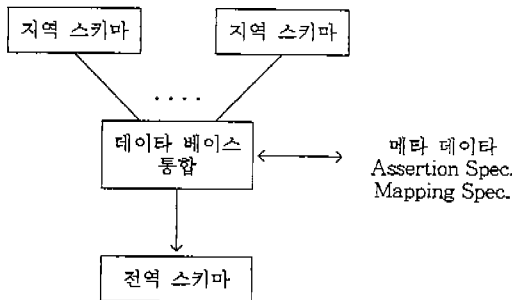
논문접수 : 1995년 7월 3일, 심사완료 : 1995년 9월 21일.



(그림 1) 스키마 통합단계
(Fig. 1) Level of Schema Integration



(그림 2) 뷰 통합
(Fig. 2) View Integration



(그림 3) 데이터베이스 통합
(Fig. 3) Database Integration

권한을 부여한 다음 사용자의 권한만을 검사하여 정보에 접근을 허용하는 임의적 보안(discretionary security)정책이었다. 그러나 이 정책은 사용자의 권한에 대한 비밀 유지가 어려운 단점이 있다. 최근의 보안 문제에 대한 연구는 사용자에게 인가등급을 부여하고 데이터베이스의 각 구성 엔티티에 대해 보안등급(classification)을 주어서 사용자 인가등급이 엔티티의 보안등급을 지배할 때 접근을 허용하는 Bell-Lapadula의 강제 보안(mandatory security)정책에 기반을 두고 있다. 특히 객체 지향 데이터베이스의 보안에 대한 연구는 Thuraisingham, Lunt에 의해 다단계 보안 개념으로 발전되어 연구되어지고 있다[4, 7, 8, 13, 14, 16].

본 논문에서는 분산 데이터베이스 환경에서 강제 보안정책과 객체지향 모델에 기초한 다단계 보안 객체지향 데이터베이스 모델로서 정의된 각 사이트의 지역 스키마를 통합하여 사용자에게 전역 가상 스키마를 제공하는데 있어서 본래의 보안성질을 유지할 수 있는 통합 방법을 다루었다.

통합 방법은 객체지향 모델의 모든 구성 요소(객체 클래스, 클래스 속성과 메소드, 객체 클래스 사이의 관계, 제약 조건 등)에 대한 의미적 관계에 의하여 통합 가능한 경우를 완전하게 다루어야 한다. 본 논문에서는 스키마 구성의 기본 요소인 객체 클래스와 객체 클래스 사이의 관계(Relationship)만을 중심으로 각 구성 요소가 서로 다른 스키마 사이에서 통합 가능한 경우를 각각 8 가지로 구분하였다[3, 10, 12].

2. 다단계 보안 객체지향 데이터베이스 모델

객체지향 모델은 실세계 객체들이 갖는 공통적인 특성을 정의한 클래스(Class)를 기본으로 한 데이터 모델링 기법이다. 클래스 구조는 공통적인 특성을 갖는 유사 객체의 속성(attribute)과 행위(behavior)를 캡슐화한 추상화 단위이다. 객체는 객체클래스의 인스턴스로서 객체의 상태는 속성으로 표현되며, 속성은 원시 데이터 형이거나 또 다른 객체일 수 있다. 객체의 행위는 메소드(method)라고 하며 속성을 조작하는 루틴들이다. 객체지향 모델은 상위 클래스와 하위 클래스

로 구성되는 클래스 계층 구조를 갖게 되며, 이 구조에서 하위 클래스는 상위 클래스의 속성과 행위를 상속(inheritance)받거나 자신의 고유한 속성과 행위를 가질 수 있다[15].

다단계 보안 객체지향 모델은 클래스 계층 구조로 모델링된 스키마의 각 엔티티-즉 클래스, 속성, 메소드, 클래스 사이의 관계-에 보안등급을 부여할 수 있도록 객체지향 모델을 확장한 것이며 각각의 구성 요소에 부여된 보안등급은 강제 보안 정책에 의한 다단계 보안성질을 따른다 [4, 16, 17].

2.1 모델 구성 요소

클래스 계층 구조로 모델링된 다단계 객체지향 데이터베이스 스키마의 객체 클래스와 객체 클래스의 구성 요소, 즉 속성, 메소드, 클래스 사이의 관계들을 정의하면 다음과 같다.

[정의 1] 객체 클래스

$C=(ID, A, M, R) [: L]$
 ID : 객체 클래스 이름
 $A=\{A_1, A_2, \dots, A_n\}$ /객체 클래스의 속성 집합/
 $M=\{M_1, M_2, \dots, M_n\}$ /객체 클래스의 메소드 집합/
 $R=\{R_1, R_2, \dots, R_n\}$ /객체 클래스 사이의 관계 집합/
 L : 객체 클래스의 보안등급

[정의 2] 객체 클래스의 속성

$A_i=(ID : Ta) [: L]$
 ID : 속성 이름
 Ta : 속성 형 C_p /원시 객체 클래스/
 C_s /단순 객체 클래스/
 C_c /복합 객체 클래스/
 L : 속성의 보안등급

[정의 3] 객체 클래스의 메소드

$M_i=(ID, Arg, Tm, P) [: L]$
 ID : 메소드 이름
 Arg : 메소드의 매개변수 집합
 Tm : 메소드 실행 결과의 형
 P : 메소드의 구현 코드
 L : 메소드의 보안등급

[정의 4] 객체 클래스 사이의 일반화(Generalization)

$R=C_i \text{ IS_A } C_j$
 C_i : 하위 클래스
 C_j : 상위 클래스

[정의 5] 객체 클래스 사이의 집단화(Aggregation)

$R=C_i \text{ IS_PART_OF } C_j$
 C_i : 부품 클래스
 C_j : 집합 클래스

[정의 6] 객체 클래스의 연관성(Association)

$R=(Name, C\text{-seq}, A, M) [: L]$
 Name : 연관성 이름
 $C\text{-seq}=\{C_1 : card_1, C_2 : card_2, \dots, C_k : card_k\}$, $k \geq 1$
 /R에 관계된 객체 클래스 집합/
 k : R의 항수(Degree)
 $card_i$: R에 관계된 객체 클래스 C_i 의 대응수(Cadinality)
 L : 연관성의 보안등급
 $card_i$ 는 $\langle l, u \rangle$ 로 표현하며 l 은 대응수의 하한값, u 는 대응수의 상한 값을 나타낸다.

2.2 다단계 보안성질

다단계 보안 객체지향 데이터베이스 모델에서는 객체 클래스와 객체 클래스의 구성 요소, 즉 속성, 메소드, 객체 클래스 사이의 관계 등이 서로 다른 보안등급을 가질 수 있으며, 보안등급, 보안등급 사이의 관계 및 보안등급 부여 함수의 정의는 다음과 같다[16, 17].

[정의 7] 보안등급

$L(X)=\langle l, h \rangle \subseteq \{U, C, S, TS\}$, $X \in \{C, R, A, M\}$

보안등급의 전체 집합은 $L=\{U, C, S, TS\}$ 이며 전체 순서 $U(\text{Unclassified}) < C(\text{Confidential}) < S(\text{Secret}) < TS(\text{Top Secret})$ 를 갖는다. 보안등급 $L(X)$ 는 $\langle l, h \rangle$ 로 표현되며, L 의 부분 순서 집합이다. X 는 보안등급이 부여될 수 있는 다단계 보안 객체지향 데이터베이스 모델의 모든

엔티티, 즉 객체클래스, 속성, 메소드, 클래스 사이의 관계 등이며, l 은 X 의 보안등급의 하한($LB(X)$)이고 h 는 X 의 보안등급의 상한($UB(X)$)을 나타낸다.

[정의 8] 최대하계

A, B 가 보안등급의 전체 집합 L 의 부분 순서 집합이고, A 의 모든 원소 a 와 B 의 모든 원소 b 사이에 $a \leq b$ 이면 A 의 원소 a 를 B 의 하계(lower bound)라고 하며, B 의 모든 하계 중에서 가장 큰 원소를 최대하계(Greatest Lower Bound : GLB)라고 한다. 그리고 보안등급의 집합 A 에 대한 최대하계를 찾는 함수를 $GLB(A)$ 로 정의한다.

[정의 9] 최소상계

A, B 가 보안등급의 전체 집합 L 의 부분 순서 집합이고, A 의 모든 원소 a 와 B 의 모든 원소 b 사이에 $b \leq a$ 이면 A 의 원소 a 를 B 의 상계(upper bound)라고 하며, B 의 모든 상계 중에서 가장 작은 원소를 최소상계(Least Upper Bound : LUB)라고 한다. 그리고 보안등급의 집합 A 에 대한 최소상계를 찾는 함수를 $LUB(A)$ 로 정의한다.

[정의 10] 보안지배

엔티티 X 와 Y 의 보안등급을 각각 $L(X)$, $L(Y)$ 라고 할 때 $LB(X) \geq LB(Y)$ 이고 $UB(X) \geq UB(Y)$ 이면 X 가 Y 를 보안지배한다고 하며, $L(X)$ 가 $L(Y)$ 보다 크거나 같다는 의미를 갖는다.

[정의 11] 보안등급 부여 함수

지역 스키마의 엔티티 X_1, X_2, \dots, X_n 으로 부터 스키마 통합에 의해 생성된 엔티티 X 의 보안등급을 정하는 함수를 보안등급 부여 함수(Classification Grant Function)라고 하며, 다음과 같이 정의한다.

■ 상위 보안등급 부여 함수(UCGF)

$UCGF(X_1, X_2, \dots, X_n) = (LUB(LB(X_1), LB(X_2), \dots, LB(X_n)), LUB(UB(X_1), UB(X_2), \dots, UB(X_n)))$

■ 하위 보안등급 부여 함수(LCGF)

$LCGF(X_1, X_2, \dots, X_n) = (GLB(LB(X_1), LB(X_2), \dots, LB(X_n)), GLB(UB(X_1), UB(X_2), \dots, UB(X_n)))$

다단계 보안 객체지향 데이터베이스 모델의 객체클래스, 객체클래스의 속성과 메소드, 객체클래스 사이의 관계들의 보안등급은 다음과 같은 보안성질을 갖는다.

[성질 1] 원시 클래스의 보안성질

객체지향 데이터베이스 시스템에서는 원시 클래스(시스템 정의 클래스)가 존재한다. 이 클래스는 모든 사용자(주체)가 사용 가능해야 하기 때문에 원시 클래스의 보안등급은 시스템에서 정의할 수 있는 가장 낮은 보안등급인 "system-low"를 갖는다.

[성질 2] 일반화의 보안성질

서브클래스의 정의는 슈퍼클래스의 정의로부터 유도될 수 있다. 따라서 서브클래스의 보안등급은 슈퍼클래스의 보안등급을 지배해야 한다.

$$L(C_2) \geq L(C_1) / C_2 \text{는 } C_1 \text{의 서브클래스}$$

[성질 3] 집단화의 보안성질

객체 클래스 C 가 객체 클래스 C_1, C_2, \dots, C_n 의 집단화 클래스라면 C 의 보안등급은 부품클래스들의 보안등급을 지배해야 한다.

$$L(C) \geq LUB(L(C_1), L(C_2), \dots, L(C_n))$$

[성질 4] 구조적 보안성질

객체 클래스의 구성요소 V (속성, 메소드, 제한조건)의 보안등급은 객체 클래스 C 의 보안등급을 지배해야 한다.

$$L(V) \geq L(C)$$

[성질 5] 상속의 보안성질

객체 클래스 C_1 으로부터 상속한 C_2 의 구성요소인 V 의 보안등급은 C_1 의 V 보다 높아야 한다.

$$L(V, V \in C_2) \geq L(V, V \in C_1)$$

[성질 6] 연관성의 보안성질

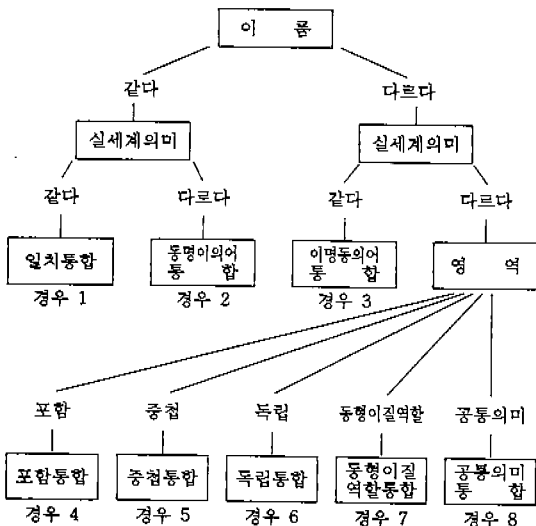
객체 클래스 R이 객체 클래스 C_1, C_2, \dots, C_k 의 관계를 정의한 연관성 객체라면 R의 보안등급은 C_1, C_2, \dots, C_k 의 보안등급을 지배해야 한다.

$$L(R) \geq \text{LUB}(L(C_1), L(C_2), \dots, L(C_k))$$

3.스키마 통합

3.1 객체 클래스의 통합

다단계 보안 객체 지향 모델에서 가장 기본적인 추상화 단계는 객체 클래스이다. 클래스는 실세계의 모델과 데이터베이스 설계자의 논리적 인식을 가장 자연스럽게 반영한다. 객체 클래스의 통합은 객체의 이름, 실세계 의미 및 객체 인스턴스의 영역을 비교함으로써 수행하며, 통합이 가능한 경우의 정의는 (그림 4)와 같다. 각각의 경우에 대해서 통합이 이루어질 때 통합된 스키마에 새로 생성된 클래스의 보안등급은 원래의 보안등급과 보안등급의 기본 성질을 만족해야 한다. 보안등급의 충돌 문제를 해결하기 위한 정책은 데이터 보안의 목적과 보안등급의 기본 성질을 만족하는 높은 등급 우선 정책을 기본으로 한



(그림 4) 객체 클래스의 통합 분류

(Fig. 4) classify the integration of object classes

다. 다음은 이들 각 경우의 통합에 대해서 통합 연산과 보안성질의 유지 규칙을 정의한 것이다. 통합은 편의상 임의의 객체 클래스

$$C_1=(ID, A, M, R) : L, C_2=(ID, A, M, R) : L$$

사이에서 발생 가능한 통합을 다루며, 이는 간단히 n개의 클래스에 대한 통합으로 확장될 수 있다. 통합될 객체 클래스 C_1 과 C_2 의 보안등급은 각각 $L(C_1), L(C_2)$ 로 표현한다.

[경우 1] 일치통합(Identical Integration)

클래스 C_1 과 C_2 가 같은 의미를 표현하고 동일한 영역(Domain)을 가질 때, 하나의 클래스 C로 통합될 수 있으며 통합 연산과 클래스 C의 보안등급은 다음과 같이 정의된다.

Combine(C_1, C_2) /결과클래스 C/
 $ID(C)=ID(C_1)=ID(C_2)$
 $Dom(\dot{C})=Dom(C_1)=Dom(C_2)$
 $A(C)=A(C_1) \cup A(C_2),$
 $M(C)=M(C_1) \cup M(C_2)$
 $L(C)=UCGF(C_1, C_2)$

[경우 2] 동명이의어 통합(Homonym Integration)

클래스 C_1 과 C_2 가 이름은 같지만 실세계 의미가 전혀 다를 경우이며, 통합 스키마에는 서로 이름이 다른 새로운 클래스가 추가되어야 한다. 이 경우의 통합 연산과 보안등급은 다음과 같다.

Rename(C_1) /결과 클래스 C_1' /
 $L(C_1')=L(C_1)$

[경우 3] 이명동의어 통합(Synonym Integration)

클래스 C_1 과 C_2 가 이름은 다르지만 실세계 의미가 동일한 경우이며, 통합 스키마에는 C_1 과 C_2 의 이름을 반영한 객체 클래스 C가 정의 된다. 통합 연산과 스키마의 보안등급은 다음과 같다.

Concate(C_1, C_2) /결과 클래스 C/
 $ID(C)=SS(ID(C_1), ID(C_2))/ID_1$ 과 ID_2 의 의미 합(Semantic Sum)/
 $Dom(C)=Dom(C_1)=Dom(C_2)$

$$\begin{aligned} A(C) &= A(C_1) \cup A(C_2), \\ M(C) &= M(C_1) \cup M(C_2) \\ L(C) &= UCGF(C_1, C_2) \end{aligned}$$

[경우 4] 포함 통합(Containment Integration)

클래스 C1의 영역이 클래스 C2의 영역에 포함 되었을 때 ($Dom(C_1) \subseteq Dom(C_2)$), 통합 스키다는 새로운 클래스 C'와 C2'를 갖게 되고 C'는 C2'의 하위클래스가 되며, 통합 연산과 보안등급은 다음과 같다.

Generalize(C1, C2) /결과 클래스 C', C2' /

$$\begin{aligned} ID(C_1') &= ID(C_1) & ID(C_2') &= ID(C_2) \\ Dom(C_1') &= Dom(C_1) & Dom(C_2') &= Dom(C_2) \\ A(C_1') &= A(C_1) - A(C_2) & A(C_2') &= A(C_2) \\ M(C_1') &= M(C_1) - M(C_2) & M(C_2') &= M(C_2) \end{aligned}$$

If $L(C_1) \geq L(C_2)$
 then { $L(C_1') = L(C_1)$; $L(C_2') = L(C_2)$ }
 else { $L(C_1') = UCGF(C_2)$; $L(C_2') = L(C_2)$ }

[경우 5] 중첩 통합(Overlapping Integration)

클래스 C1의 영역과 클래스 C2의 영역이 중첩 되었을 때 ($Dom(C_1) \cap Dom(C_2) \neq \emptyset$), 통합 스키다는 새로운 클래스 C', C2' 그리고 C를 갖게 되고 C', C2'는 C의 하위클래스가 되며, 통합 연산과 보안등급은 다음과 같다.

Specialize(C1, C2, C) /결과 클래스 C, C', C2' /

$$\begin{aligned} ID(C) &= SS(ID(C_1), ID(C_2)) \\ ID(C_1') &= ID(C_1) \\ ID(C_2') &= ID(C_2) \\ Dom(C) &= Dom(C_1) \cup Dom(C_2) \\ Dom(C_1') &= Dom(C_1) \\ Dom(C_2') &= Dom(C_2) \\ A(C) &= A(C_1) \cap A(C_2) \\ A(C_1') &= A(C_1) - A(C) \\ A(C_2') &= A(C_2) - A(C) \\ M(C) &= M(C_1) \cap M(C_2) \\ M(C_1') &= M(C_1) - M(C) \\ M(C_2') &= M(C_2) - M(C) \\ L(C) &= LCGF(C_1, C_2) \\ L(C_1') &= L(C_1) \\ L(C_2') &= L(C_2) \end{aligned}$$

[경우 6] 독립 통합(Disjoint Integration)

이 경우는 객체 클래스가 유사하게 정의되어 있더라도 공통적인 영역을 갖지 않은 경우이며 ($Dom(C_1) \cap Dom(C_2) = \emptyset$), 클래스 C1과 C2가 같은 속성이나 메소드를 가지고 있더라도 실세계 의미가 다르기 때문에 통합이 될 수 없으며, 통합 스키마에 그대로 포함되게 된다. 예를 들면 다음의 두 클래스

$$\begin{aligned} &Full-Professor(dep, name, salary) \\ &Graduate-Student(dep, name, salary, service) \end{aligned}$$

는 상당히 많은 속성을 공유함에도 불구하고 이들이 표현하는 실세계 의미가 다르기 때문에 통합이 될 수 없다.

[경우 7] 동형이질 역할 통합(Isomorphic-Role Integration)

클래스 C1과 C2가 다른 이름과 다른 실세계 의미를 가지면서도 동형이질적 역할 관계가 있고 통합 가능성을 지닌 경우이며, 통합 스키마는 새로운 두 개의 가상 클래스와 C', C2'를 갖게 된다. 하나는 C'와 C2'의 일반화 클래스 GC이며 GC의 생성은 경우 5와 같다. 다른 하나는 C'와 C2'의 다중 상속 클래스 MC이다. MC의 생성을 위한 통합 연산자와 보안등급은 다음과 같다.

M-inherit(C1, C2, C) /결과클래스 C, C1, C2' /

$$\begin{aligned} ID(C) &= SS(ID(C_1), ID(C_2)) \\ Dom(C) &= Dom(C_1) \cup Dom(C_2) \\ A(C) &= A(C_1) \cup A(C_2) \\ M(C) &= M(C_1) \cup M(C_2) \\ L(C) &= UCGF(C_1, C_2) \end{aligned}$$

그리고 C1과 C2의 동형이질 역할 통합은 다음과 같이 구현될 수 있다.

$$\begin{aligned} &Specialize(C_1, C_2, GC) \\ &M-inherit(C_1', C_2', MC) \end{aligned}$$

[경우 8] 공통의미 통합(Common Semantic Integration)

클래스 C1과 C2의 영역이 다르고 실세계 의미가 다르더라도 두 클래스가 공유 속성을 가지고 있

고 통합이 가능하다면 [경우 5]와 같이 통합될 수 있다. 그러나 클래스 C_1 과 C_2 의 영역은 여전히 독립되어 있다. 예를 들면 다음의 두 클래스

```
Secretary(emp-no, sal, name, grade)
Engineer(emp-no, sal, name, engineer-type)
```

는 "Employee(emp-no,sal,name)"로 통합될 수 있다.

3.2 관계의 통합

관계의 통합은 객체 클래스의 통합시 통합된 객체 클래스들에 관련된 관계의 통합으로서 스키마 S_1 의 객체 클래스 C_1 이 스키마 S_2 의 객체 클래스 C_2 와 유사하거나 일치될 때, C_1 에 관련된 모든 관계와 C_2 에 관련된 모든 관계에 대해서 통합이 고려된다. 그러나 관계의 실제적인 합병은 관련된 객체 클래스의 역할(Role)과 관계의 항수(Degree)와 대응수(Cardinality)조건에 의한 충분한 의미 정보가 설계자에 의해 제공되어야 하며 사전 통합 분석시 정확한 조사가 이루어져야 한다. 본 논문에서는 관계의 의미 정보가 충분히 제공된 환경에서 임의의 통합 가능한 두 개의 이항 관계

```
R1=(ID, C1 : Card1, C2 : Card2, A, M) : L
R2=(ID, C1 : Card1, C2 : Card2, A, M) : L
```

에 대한 통합을 다루었으며, 이는 N개의 N항 관계의 통합으로 쉽게 확장될 수 있다.

관계의 통합 가능한 경우는 (그림 5)와 같이 관계에 대한 다음의 세 가지 특성에 의해 8가지로 분류될 수 있다.

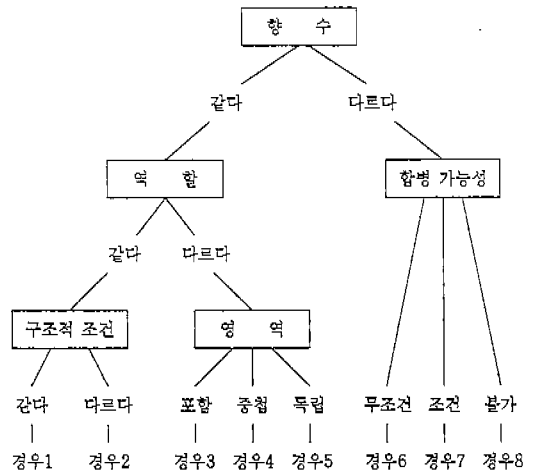
- ① 관계의 항수 : 관계에 관련된 객체클래스의 수이다.
- ② 객체클래스의 역할 : 관계에 관련된 객체클래스의 역할은 역할명으로 정의된다. 만일 역할명과 관계 인스턴스가 서로 다른 스키마의 관계 사이에서 일치되고 동일한 항수를 갖는다면 두 관계는 동일하다.
- ③ 구조적 조건 : 두 객체클래스 사이의 관계는 각각의 객체클래스의 인스턴스 사이의 사상(Mapping)이다. 사상의 조건을 표현하기 위하여 데이터 모델에 의해 지원된 정의 규칙을 구조적

조건이라고 하며, 대응수 조건은 구조적 조건의 일종이다.

다음은 각각의 관계 통합의 경우에 대해서 보안등급을 정하는 규칙을 정의한 것이다.

[경우 1] 관계 R_1 과 R_2 의 항수, 역할, 구조적 조건이 모두 같은 경우이며 통합된 스키마는 하나의 관계 집합 R 을 갖게 되고, 이때 R 의 보안등급은 다음과 같다.

```
Combine(R1, R2) /결과 관련성 R=(ID, C1 :
Card1, C2 : Card2, A, M) : L
C1(R)=Combine(C1(R1), C1(R2))
C2(R)=Combine(C2(R1), C2(R2))/
```



(그림 5) 관계의 통합 분류
(Fig. 5) classify the integration of relationships

[경우 2] 관계 R_1 과 R_2 의 구조적 조건이 다를 경우이며, 다시 말해서 임의의 $C_i(R_1)$ 과 $C_i(R_2)$ 의 영역이 서로 포함관계를 갖는 경우로서 통합된 스키마는 하나의 관계 R 을 갖게 되고 R 의 객체클래스 C_i 는 포함 통합이 이루어진다.

```
If Dom(C1(R1)) ⊆ Dom(C1(R2)) Then
{Generalize(C1(R1), C1(R2)),
Concate(C2(R1), C2(R2))}
Concate(R1, R2) /결과 관계 R=(ID, C1 : Card1,
C2 : Card2, A, M) : L
```

$C_1(R) = \text{Generalize}(C_1(R_1), C_1(R_2))$ 의 하위 클래스

$$C_2(R) = \text{Concate}(C_2(R_1), C_2(R_2)) /$$

[경우 3] 관계 R_1 의 모든 인스턴스가 R_2 의 인스턴스에 포함될 경우, 즉 $\text{Dom}(R_1) \subseteq \text{Dom}(R_2)$ 이며 통합된 스키마는 $R_1' = R_1$ 과 $R_2' = R_2$ 를 갖게 되고 R_1' 는 R_2' 의 서브 관계로서 통합된다.

Generalize(R_1, R_2) /결과 관련된

$$R_1' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$R_2' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$C_1' = \text{Concate}(C_1(R_1), C_1(R_2))$$

$$C_2' = \text{Concate}(C_2(R_1), C_2(R_2)) /$$

[경우 4] 관계 R_1 과 R_2 에 관련된 임의의 객체 클래스 $C_i(R_1)$ 와 $C_i(R_2)$ 사이에서 중첩 영역이 존재할 때 이들 객체클래스는 중첩 통합이 이루어지며, 아울러 이러한 객체와 관련된 관계도 중첩 통합이 이루어진다. 통합된 스키마에는 새로운 관계 집합 R 과 $R_1' = R_1, R_2' = R_2$ 가 다음과 같이 정의된다.

If $\text{Dom}(C_1(R_1)) \cap \text{Dom}(C_1(R_2)) \neq \emptyset$ then

$$\{\text{Specialize}(C_1(R_1), C_1(R_2), C_1(R)),$$

$$\text{Concate}(C_2(R_1), C_2(R_2))\}$$

Specialize(R_1, R_2, R) /결과 관계

$$R = (\text{ID}, C_1 : 1, C_2 : 1, A, M) : L$$

$$C_1(R) = \text{Specialize}(C_1(R_1), C_1(R_2), C_1)$$

상위 클래스

$$C_2(R) = \text{Concate}(C_2(R_1), C_2(R_2))$$

$$R_1' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$C_1'(R_1') = \text{Specialize}(C_1(R_1), C_1(R_2), C_1)$$

의 하위 클래스

$$C_2'(R_1') = \text{Concate}(C_2(R_1), C_2(R_2))$$

$$R_2' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$C_1'(R_2') = \text{Specialize}(C_1(R_1), C_1(R_2), C_1)$$

의 하위 클래스

$$C_2'(R_2') = \text{Concate}(C_2(R_1), C_2(R_2)) /$$

[경우 5] 관계 R_1 과 R_2 의 모든 객체 클래스 쌍의 영역과 역할이 서로 동일하면서 인스턴스들이 서로 다를 경우이며, 통합된 스키마에는 객체 클

래스 쌍이 이명동의어 통합되고 통합된 객체 클래스들과 관련된 관계 R_1 과 R_2 는 R_1' 와 R_2' 로 재정의 된다.

$$R_1' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$R_2' = (\text{ID}, C_1' : \text{Card}_1, C_2' : \text{Card}_2, A, M) : L$$

$$C_1' = \text{Concate}(C_1(R_1), C_1(R_2))$$

$$C_2' = \text{Concate}(C_2(R_1), C_2(R_2))$$

$$L(R_1') = \text{UCGF}(\text{UCGF}(C_1', C_2'), R_1)$$

$$L(R_2') = \text{UCGF}(\text{UCGF}(C_1', C_2'), R_2)$$

[경우 6] 관계 R_1 과 R_2 의 함수는 다르지만 의미적으로 동일한 정보를 표현하고 있고, R_1 의 모든 인스턴스가 R_2 로부터 조건없이 유도될 수 있을 때 R_1 은 R_2 로 흡수 통합될 수 있다. 이 경우 통합은 객체 클래스 사이의 통합에 유의하여야 하며 통합된 스키마의 관계 집합은 흡수 통합된 관계 집합 R 만 존재 가능하다. 이 때 R 의 보안등급은 R_1 과 R_2 의 보안등급을 반영해야 한다.

$$\text{Concate}(R_1, R_2)$$

[경우 7] 관계 R_1 과 R_2 의 함수는 다르지만 의미적으로 유사한 정보를 표현하고 있고, R_1 의 모든 인스턴스가 R_2 로부터 특정 조건에 의해 유도될 수 있을 때 R_1 은 R_2 로 조건 흡수 통합될 수 있다. 통합의 방법은 [경우 6]과 유사하지만 통합에 사용된 조건이 설계자의 의도와 같아야 하며, 통합된 스키마에 그 정보가 유지되어야 한다.

[경우 8] 관계 R_1 과 R_2 가 서로 완전히 상이하여 통합이 불가능한 경우이며, 통합된 스키마는 이들 관계를 수정없이 그대로 포함하여야 한다.

4. 결 론

80년대에 들어와서부터 데이터베이스 시스템은 분산 환경으로 발전되어지고 있으며, 데이터베이스의 보안 문제가 심각하게 대두되고 있는 실정이다. 따라서 분산 DBMS는 네트워크의 각 사이트에서 서로 다른 사용자에게 의해 독립적으로 설계, 관리, 유지보수되고 있는 지역 스키마들을 통합하여 전역 가상 스키마를 제공하며, 특정 사

이트의 사용자가 다른 사이트의 지역 데이터베이스를 투명적으로 이용할 수 있는 환경을 지원하고, 각 지역 스키마에 부여된 스키마 구성 엔티티들의 보안성질이 통합된 전역 스키마에서도 유지되도록 해야한다.

최근까지 다양한 형태의 분산 DBMS에 대한 스키마 통합의 연구는 활발히 진행되어 왔지만 지역 스키마의 보안성질을 유지할 수 있는 통합에 대한 연구는 거의 이루어지지 않았다.

본 논문에서는 다단계 보안 분산DBMS 환경 하에서 다단계 보안성질을 갖는 각 지역 스키마의 통합이 이루어질 때 본래의 보안등급과 다단계 보안성질을 유지할 수 있도록 하는 통합 방법을 제안하였다. 각 지역 스키마는 다양한 사용자 또는 설계자의 인식, 설계에 사용한 모델, 모델 구성 요소, 무결성 조건 등에 의해서 다양성을 갖는다.

본 논문에서는 분산 데이터베이스 스키마 모델로서 객체지향 모델을 확장한 다단계 보안 객체지향 데이터베이스 모델을 사용하였으며, 이를 이용한 통합 연산자 Combine, Rename, Concat, Generalize, Specialize, M-inherit를 정의하였다.

통합 방법은 데이터베이스 스키마의 기본 구성 요소이자 실세계 응용영역의 기본적인 추상화 단위인 객체 클래스와 이들 사이의 관계에 대한 통합을 중심으로 각각 통합이 가능한 경우를 8가지로 구분하여 정의하였다.

본 연구는 분산 DBMS에서 보안성질을 갖는 스키마 통합에 있어서 통합의 기본 방향과 보안성질의 유지 규칙을 제안한 것이며 향후 더욱 연구되어야 하거나 추가로 연구하고자 하는 내용은 다음과 같다.

첫째, 현재 정의한 통합 가능한 경우의 확장 및 형식 정의를 강화하고자 한다.

둘째, 객체 클래스와 클래스 사이의 관계에 대한 실세계 의미 정보의 유지, 검색, 비교 방법 등에 대한 고찰이 필요하다.

셋째, 통합된 스키마의 보안등급과 보안성질의 검증이 필요하다.

넷째, 자동 통합을 위한 알고리즘을 개발하고자 한다.

참 고 문 헌

- [1] C. Batini, M. Lenzerini and S. B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration", ACM Computing Surveys, Vol.18, No.4, pp.323-364, Dec.1986.
- [2] C. Batini, S. Ceri, S. B. Navathe, Conceptual Database Design, The Benjamin/Cummings Publishing Company, pp.119-137, 1992.
- [3] N. B. Idris, W. A. Gray and M. A. Qutaishat, "Integration of Secrecy Features in a Federated Database Environment", IFIP, pp.89-108, 1994.
- [4] T. F. Keefe, W. T. Tsai and M. B. Thuraisingham, "SODA : A Secure Object-Oriented Database System", Computer & Security, No.8, pp.517-533, 1989.
- [5] W. Kim, Modern Database Systems, Addison-Wesley Publishing Company, pp.521-550, 1995.
- [6] J. Larson, S. Navathe and R. Elmasri, "A Theory of Attribute Equivalence in Databases with Application to Schema Integration", IEEE Trans. on Software Engineering, Vol.15, No.4, pp.449-463, Apr.1989.
- [7] T. F. Lunt, "Multilevel Security for Object-Oriented Database Systems", IFIP, pp.199-209, 1990.
- [8] J. K. Millen, T. F. Lunt, "Security for Object-Oriented Database Systems", IEEE Research in Security and Privacy, pp.260-272, May. 1992.
- [9] Amihai Motro, "Superviews : Virtual Integration of Multiple Database", IEEE Trans on Software Engineering, Vol.13, No.7, pp.785-798, Jul.1987.
- [10] S. Navathe, R. Elmasri and J. Larson, "Integrating User Views in Database Design", IEEE COMPUTER Magazine,

pp.50-62, 1986.

[11] M. T. Ozsu, P. Valduriez, "Distributed Database Systems: Where Are We Now?", IEEE computer, Vol.24, No.8, pp.68-78, Aug.1991.

[12] M. A. Qutaishat, N. J. Fiddian and W. A. Gray, "A Schema Meta-Integration System for a Heterogeneous Object-Oriented Database Environment Objectives and Overview," NordDATA '92 Precedings, pp.74-92, 1992.

[13] J. M. Slack, "Security In An Object-Oriented Database", IEEE New Security Paradigms Workshop Proc., pp.155-159, Sep.1992.

[14] F. N. Springsteel, "Object Based Schema Integration for Heterogeneous Databases : A Logical Approach," DEXA'93, pp.166-180, Sep.1993.

[15] T. J. Teorey, Database Modeling and Design, Morgan Kaufmann Publishers, Inc., pp.46-60, 1990.

[16] M. B. Thuraisingham, "Mandatory Security in Object-Oriented Database Systems", OOPSLA '89 Proceedings, pp.203-210, Oct.1989.

[17] 노봉남, 김용성, 장옥배, "다단계 객체지향 계층구조에서 상속의 보안성질", 한국정보과학회 논문지, Vol.20, No.9, pp.1346-1352, Sep.1993.



박우근

1983년 전남대학교 계산통계학과 졸업(학사)
 1985년 전남대학교 대학원 계산통계학과 전산학 전공(석사)
 1993년 전남대학교 대학원 전산통계학과 박사과정 수료
 1986년~현재 광주대학교 전자

계산학과 부교수
 관심분야: 데이터베이스 설계 및 모델링, 분산 시스템, 객체지향 시스템, 퍼지 시스템, 정보 보안 등



노봉남

1978년 전남대학교 수학교육과 이학사
 1982년 한국과학기술원 전산학과 공학석사
 1994년 전북대학교 대학원 전산통계학과 이학박사
 1983년~현재 전남대학교 전산

학과 교수
 관심분야: 객체지향 시스템, 통신망 관리, 정보 보안, 컴퓨터와 정보사회 등