보안이 적용된 객체모델의 객체지향 스키마로의 변환 방법

김 정 종 박 운 재 송 호 영 김 재 영°
경남대학교 컴퓨터공학과

The Transformation of an Object Model Adopting Securities into an Object-Oriented Schema

Jung-Jong Kim Woon-Jai Park Ho-Young Song Jae-Young Kim°
Dept. of Computer Science & Engineering, Kyungnam University

요 약

객체지향 방법으로 시스템을 개발할 때 분석단계의 객체모델을 객체지향 스키마로 변환을 용이하도록 하기 위하여 분석단계의 객체모델을 정제할 필요가 있다. 따라서 본 논문에서는 분석단계의 객체모델을 정제하여 다단계 데이터베이스 어플리케이션으로 설계하는 방법을 제시한다. 또한 자료의 잘못된 유출이나 수정을 예방하고 모호성을 제거하기 위하여 보안을 적용한다. 보안을 적용한 분석단계의 객체모델을 다단계 데이터베이스 어플리케이션으로 설계할 때 이 보안이 다단계 데이터베이스 어플리케이션의 설계에서 적용되는 방법을 제시한다.

1. 서 론

객체지향 방법으로 시스템을 개발할 때 먼저 의뢰자의 요구사항을 분석하여 분석단계의 모델을 만들게 된다. 그러나 이 모델은 객체지향 스키마로 변환하기 어렵다. 이와 같은 문제를 해결하기 위해 객체지향 스키마를 표현하는 다단계 데이터베이스 어플리케이션을 설계하기 위한 많은 연구들이 있었고 그 대부분은 다단계 데이터베이스 어플리케이션을 표현하기 위해 개체-관련성 모델이나 의미 모델의 변화를 이용해서 연구되어 왔다[1][2][3][4]. 이 모델들은 단지 어플리케이션의 정적 요구를 표현하는데 초점을 맞추고 있다. Pernul의 연구에서는 개체-관련성 모델과 데이터 흐름 다이어그램을 이용하여 다단계 데이터 베이스 어플리케이션의 정적, 기능적 요구를 파악한다. Serll은 다단계 데이터베이스 어플리케이션 설계를 위한 MOMT (Multilevel Object Modelling Technique)를 제안하지만 이 MOMT는 분석단계에 초점을 맞추고 있다.

본 논문은 정적인 분석단계의 객체모델을 다단계 데이터베이스 어플리케이션으로 설계할 때 클래스와 일반화를 정제하는 방법을 설명하는데 클래스가 동적 모델과 관련되는 연산을 가지지 않는다는 가정하에서 OMT 방법론을 이용하여 설명한다.

본 논문에서는 객체지향 모델 중 자료의 잘못된 유출을 예방하고 모호성을 제거하기 위해 보안개념을 적용한 객체모델을 쉽게 객체지향 스키마로 변환할 수 있는 다단계 데이터베이스 어플리케이션으로 설계하는 방법을 제시한다.

2. 기본 개념

보안은 권한이 없는 사용자에 의한 데이터 접근을 제한하여 정보의 유출, 수정, 파손 등을 방지하고 추론과 같은 간접수단을 통한 정보유출을 방지하는 것으로 시스템이 다양한 보안등급을 가진 정보를 포함하고 시스템에 포함된 어느 높은 보안등급의 정보에 접근해서는 안될 사용자가 있을 때 다단계 보안이 필요하고 보안등급은 Unclassified(U) < Confidential(C) < Secret(S) < TopSecret(TS)로 나누어진다.

객체지향 모델에서 보안등급은 객체 식별자 그리고 객체의 속성, 속성 값, 연산과 같은 구성요소와 관련된다. 따라서 객체는 객체뿐만 아니라 클래스 이름, 속성, 연산 등의 객체의 구성요소들도 서로 다른 보안등급을 가질 수 있다는 것이다. 이와 같이 객체가 다단계 보안등급을 가질 때 지켜져야 하는 몇 가지 기본 규칙이 있다.

[규칙 1] 객체의 보안등급이 L이라면 그 객체의 객체식별자의 보안등급이 L이고 이 객체는 L이상의 보안등급에서만 볼 수 있다.

객체들은 동일한 구조를 가진 객체 집합으로 모을 수 있는데 이를 클래스라고 하고 클래스의 보안등급은 그 보안등급 이상의 객체들을 포함한다는 의미이다. 객체는 객체의 상태를 나타내는 속성과 이런 속성을 조작하는 행위를 가진다.

[규칙 2] 다단계 객체지향 모델에서의 속성이 각각의 보안등급을 가질 수 있고 속성의 보안등급은 클래스 보안

등급 이상으로 해야 안전하다. 다시 말해 클래스 보안등급 이상으로 해야 허락되지 않은 정보의 유출을 막을 수 있다.

일반화는 한 클래스와 그 클래스의 정제된 버전과의 관련성으로 기존의 클래스를 상위클래스의 정제된 버전을 하위클래스라 하고 이것은 "IS-A"관계의 계층으로 정의할 수 있다.

[규칙 3] 이와 같이 클래스가 상위와 하위로 나누어 질 때 하위클래스는 상위클래스 이상의 보안등급을 가진다. 즉 하위클래스는 상위클래스의 특수화이므로 더 세밀한 정보의 보안이 필요하다.

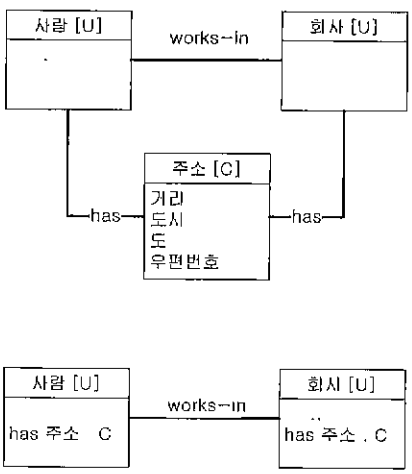
만일 상위클래스로부터 상속받는 속성의 보안등급이 하위클래스의 보안등급에 지배받을 때는 하위클래스에 있는 상속받은 속성의 보안등급을 그 속성이 속해 있는 하위클래스의 보안등급으로 변환한다.

3. 설계모델로의 변환

3.1 클래스 설계

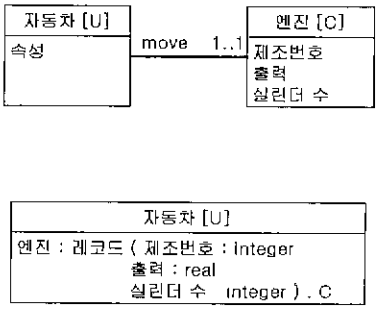
먼저 객체모델에서 클래스가 있을 때 설계자는 그것을 클래스로 표현하는지 아니면 값 타입으로 표현해야 하는지를 결정해야 한다. 타입은 값을 서술하고 클래스는 객체를 서술한다. 값은 정체성이 없다. 값은 객체의 속성 내에서만 존재하거나 연산의 매개변수로서 존재하는 기호를 나타낸다. 반대로 클래스는 정체성을 가진다. 클래스는 확장을 가지며, 클래스에 점차적으로 도입될 모든 인스턴스로 확장을 표시한다. 클래스는 속성 뿐만 아니라 연산, 제약 조건, 그리고 트리거를 가진다.

예를 들어 [그림1]에서 두 개의 클래스 '사람'과 '회사'가 참조하는 객체 클래스 '주소'는 정적 상태를 가지지만 연산은 없다. 따라서 이 '주소'는 하나의 타입으로 변환될 수 있다. 이런 타입 변환이 일어나면 각각의 클래스에 '주소'를 속성으로 추가하고 하나의 타입으로 사용한다. 이때 보안은 '주소' 클래스의 원래 보안등급 [C]를 그대로 가져와 타입의 보안등급으로 쓴다. 만일 이때 '주소' 클래스의 보안등급이 알려져 있지 않고 속성들의 보안등급만 알려져 있다면 그 속성 중 가장 낮은 보안등급을 '주소' 타입의 보안등급으로 한다. 또 '주소' 클래스의 속성을 타입이 아니라 각 클래스의 속성으로 바로 써줄 수 있는데 이때 보안등급은 [규칙 2]의 보안 규칙에 맞게 쓴다.



[그림 1] 클래스의 타입화

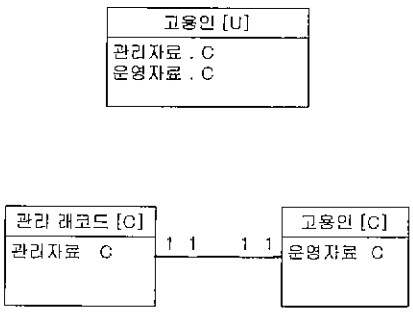
두 번째로 클래스를 조합할 수 있는지 아닌지를 결정한다. 이런 조합될 클래스 후보는 하나의 관련성만 있는 클래스이다. 만약 두 개 이상의 관련성이 있다면 조합될 수 없다



[그림 2] 클래스 조합

[그림 2]의 두 클래스 자동차와 엔진을 살펴보자. 여기서 엔진에 대한 관련은 자동차 하나뿐이고 자동차는 엔진이외의 여러 가지 관련을 가진다고 하면 엔진을 자동차의 구성품으로 한다. 만일 여기서 "자동차의 부품이 아닌 엔진"이 존재한다면 엔진에 또 다른 관련이 존재하게 되고 이 '엔진' 클래스는 독립적 클래스가 된다. 만일 "자동차의 부품이 아닌 엔진"이 존재하지 않으면 '자동차'의 속성으로 '엔진'을 모델하는 것이 편리하다. 그리고 이 엔진을 타입으로 사용할 수도 있다. 이 경우의 보안도 '엔진'의 원래의 보안등급을 가져와 '자동차'의 속성으로 만든 '엔진'의 보안등급으로 사용한다.

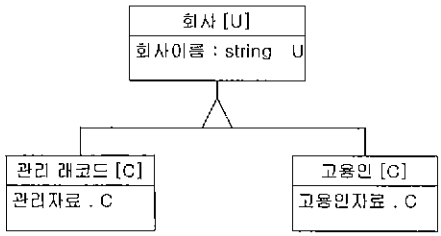
세 번째는 클래스 중 어떤 것을 두 개의 클래스로 분할할 수 있는지 아닌지를 결정하는 것이다. 이런 분할할 수 있는 클래스는 매우 많은 수의 속성과 연산을 가진다. 이런 변환은 클래스가 어떻게 접근하는지에 대한 자세한 이해가 필요하다. 이런 분할 때 속성들이 두 개의 속성 부분집합으로 중복없이 나누어지면 클래스 분할시 편리하다. 이러한 특징들은 먼저 그 클래스를 목표로 하는 연산과 제약에 대하여 검증되어야 한다.



[그림 3] 클래스 분할

[그림 3]를 보면 고용인의 속성이 관리 자료와 운영 자료로 되어있다. 이 자료는 속성의 집합으로 생각할 수 있다. 이때 어떤 회사내의 어플리케이션이 관리와 운영이라고 하면 클래스 '고용인'을 [그림 3]의 두 번째와 같이 분할한다. 이때 새로운 클래스의 보안등급은 속성 집

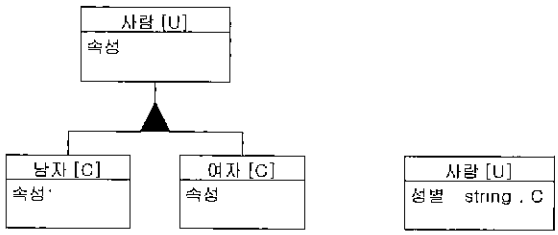
합내의 최소의 등급으로 정의된다. 이것은 2장의 기본 개념에서 모든 속성의 보안등급은 그 클래스의 보안등급 이상이라는 규칙에 만족한다. 또 두 개의 속성 부분집합의 중복되는 '회사이름'이라는 속성이 포함되면 [그림 4]와 같이 분할되며 새로운 상위클래스는 '회사이름' 속성의 보안등급 이하의 보안등급을 가진다.



[그림 4] 클래스 분할

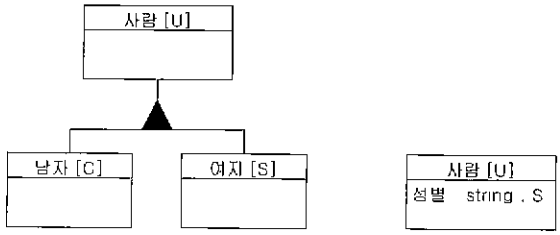
3.2 일반화 설계

객체모델에서 일반화 계층 구조에 대한 초기의 정의가 유효한가를 검증하기 위해 일반화 계층구조를 조사한다. 일반화 계층구조를 위해 어떤 기준이 주어져야 한다. 일반화의 하위클래스는 상위클래스의 정적 상태에 대하여 부가적인 정보를 저장할 수 있도록 확장된 정적 상태를 가지거나 상위클래스가 지원하지 못하는 연산이나 제약조건이 있어야 한다. 이런 조건들을 하나도 갖추지 않으면 하위클래스는 연관된 의미가 없고 하위클래스에 부가적인 구조적 정보가 없기 때문에 하위클래스를 제거할 수 있다. 그러나 보안에 대해서는 유효하다. 따라서 하위클래스의 보안등급이 같을 때 하위클래스들은 상위클래스에 속성을 추가함으로 제거될 수 있다.



[그림 5] 일반화 제거

[그림 5]을 보면 '사람' 클래스의 하위클래스로 '남자'와 '여자'가 있다. 여기서 정적 구조에 영향을 주는 속성이나 '사람' 클래스가 지원하지 못하는 연산이나 제약조건이 없다고 하면 하위클래스 '남자'와 '여자'를 상위클래스에 성별이란 속성을 추가하여 제거할 수 있다. 단



[그림 6] 일반화의 제거

보안등급 관점에서는 유효하기 때문에 하위클래스의 보안등급이 동일할 때 제거 할 수 있다.

그리고 [그림 6]과 같이 보안등급이 다를 때는 상위클래스 속성의 보안등급을 하위클래스 중 높은 쪽을 보안등급으로 하여 하위클래스를 제거하는데 그 이유는 비록 요구하는 자료를 제공하지 못할 수도 있지만 보안의 주목적은 자료의 잘못된 유출과 불법적 수정을 막는 것이기 때문이다. 결국 이때 '보안등급이 C인 사용자의 남자에 대한 자료 요구는 포기한다' 라는 문제가 생긴다.

4. 결론

본 논문에서는 분석단계의 객체모델을 정제하여 다단계 데이터베이스 어플리케이션으로 설계하는 방법을 제시하였다. 또한 자료의 잘못된 유출이나 수정을 예방하고 모호성을 제거하기 위하여 보안을 적용한 분석단계의 객체모델을 다단계 데이터베이스 어플리케이션으로 설계할 때 이 보안이 다단계 데이터베이스 이플리케이션의 설계에서 적용되는 방법을 제시하였다. 이와 같이 설계된 데이터베이스 어플리케이션은 적절한 의미언어를 사용하여 객체지향 스키마로 변환할 수 있다.

본 논문은 객체의 연산에 대해 거의 언급하지 않았는데 연산의 구조가 바뀌게 되면 그에 따라 동적 성질이 바뀌기 때문이다 따라서 추후의 연구과제로는 동적 모델이나 기능 모델 등 객체지향 모델의 다른 모델에의 변환에 대한 연구도 함께 되어야 하고 정적 모델의 관련성의 변환에 대해서도 연구되어야 한다.

참고문헌

[1] Burns. R., "ER Approach to Multilev지 Database Design" Presented at the 1st RADC Database security Workshop Menlopark. CA. May 1988.

[2] Smith. G., "Modelling Security Relevant Data Semantics", Proceedings of the IEEE Symposium on Security and Privacy. Oakland. CA. 1990

[3] Wiseman, S., "Abstract and concrete Models for secure Database Application". Proceedings of the 5th IFIP Working Conference in Database Security. Shepherstown. W. VA. November 1991.

[4] Sell. P., "The Spear Data Design Methodology". Proceedings of the 6th IFIP Database Security Conference. Vancouver, BC, August 1992.

[5] G. Pernul. "Security Constraint Processing During Multilevel Secure Database Design". proc. 8th Annual Computer Security Applications Conference". IEEE Computer Society Associations. 1992. pp75-84

[6] P. J. Serll and B. M. Thuraisingham. "Applying OMT for Designing Multilevel Database Application". DATABASE SECURITY Ⅷ: Status and Prospects. North-Holland. pp.41-64. 1994.

[7] 노 봉남. "다단계 객체지향 계층구조에서 상속의 보안 성질". 한국정보과학회 논문지 '93. 9. Vol.20, No. 9, September pp1346-1352.