

보안 객체-관련성 모델을 위한 편집기

정 언 철, 이 영 특, 노 봉 남
전남대학교 전산학과

A Graphic Editor for Secure Object-Relationship Model

Y. C. Jeong, Y. R. Lee, B. N. Noh
Dept. of Computer Science,
Chonnam National University

요 약

실세계를 모델링하기 위한 많은 데이터 모델링 자동화 도구가 연구되어 왔다. 본 논문에서는 보안 객체-관련성 모델을 표현하기 위한 그래픽 편집기를 설계 구현하였다. 보안 객체-관련성 모델은 객체 데이터 모델링 기법(OMT)에 기초하고 이에 보안성을 확장하여 설계된 모델이다. 그래픽 사용자 인터페이스(GUI)인 SODE(Security Object Diagram Editor) 편집기는 유닉스 시스템에서 X 윈도우 시스템을 사용하여 구현하였다.

1. 서 론

컴퓨터 기술의 발전으로 인해 하드웨어 가격이 저렴해지고 컴퓨터 응용범위가 확대됨에 따라 기존의 컴퓨터에 대한 인식을 바꿔 놓았다. 즉 컴퓨터 전문가에 의해 사용되던 특수개념에서 사무실, 은행, 일반 가정에 이르는 일상적인 개념으로 변화시켜 놓았다. 컴퓨터 응용의 확대는 다양한 사용자의 많은 요구사항을 포함하게 되었다. 이러한 컴퓨터 확대에 병행한 큰 변화는 일반 사용자들이 사용하기 편리하게 하는 인터페이스의 발전이다. 인터페이스는 컴퓨터 중심의 작업이 아닌 사용자 중심의 형태로 제공된다. 인터페이스는 전문지식을 갖지 않은 일반 사용자에게 기존의 명령어 입력과 같은 복잡한 방식을 피하고 마우스와 같은 단순한 입력장치를 조작함으로써 다양한 형태의 작업을 쉽게 할 수 있게 하여 컴퓨터에 대한 거부감을 해소시켜 준다[8].

사용자 인터페이스는 사용자가 시스템과 대화 형식에 대한 개념으로 문자형태의 명령어 입력 방식과 메뉴 사용 방식으로 크게 구분된다. 일반적으로 인터페이스라 함은 사용자 중심의 작업 형태로 명령어 사용보다 메뉴와 같은 직관적이고 단순한 형태의 작업을 가리킨다. 이러한 방식은 사용자의 필요에 따라 개발되기도 하고 사용하는 개발용 시스템에서 제공하기도 한다.

사용자 인터페이스의 구성요소는 윈도우(window), 메뉴(menu), 대화상자(dialog box), 아이콘(icon)과 같은 그래픽 형태로 사용자에게 제공된다. 그래픽 사용자 인터페이스(GUI)는 점과 클릭(point and click)에 의한 접근 방식을 취한 애플의 매킨토시가 시초였다. 이와 유사한 개념으로 만들어져 광범위한 데이터베이스 응용에 사용되는 X-Window가 있는데, 툴킷을 이용한 구현을 제공한다. 인터페이스 개발은 SRI, Xerox사에서 시작되어 매킨토시의 출현과 현재의 개인용 컴퓨터환경에서 사용자에게 보다 친숙한 형태로 발전하고 있다. 또한 UNIX를 운영체제로 하는 워크스테이션에서 이용되는 X/Open, APP나 OSF 등 국제 기구에서 표준으로 굳어진 X 윈도우 시스템이 있다.

이러한 사용자 인터페이스 추세는 80년대의 관계형 데이터베이스에서도 발전하여 90년대의 새로운 방식의 접근과 사용자 인터페이스 기술을 요구하게 되었다. 1980년대 후반과 1990년대 초기의 관계형 데이터베이스의 빠른 발전은 새로운 디스

플레이와 사용자 인터페이스 기술로 모든 컴퓨터 자료의 저장 및 복귀를 기존의 문서 형태에서 그래픽 형태로 정보를 제공하는 형태로 변환되었다.

관계형 데이터베이스의 구현과 관리를 위한 관리자는 다른 시스템과의 통신에 제약을 받게 되는데 이는 인터페이스 이식성의 부족에서 발생한다. 이러한 환경에서 서로 다른 시스템 사이에는 상호간에 구축된 데이터베이스 시스템의 정의된 많은 자료를 이용할 수 없다. 관계형에서 제공하는 테이블 형태의 표현에 비해 그래픽 표현 형태를 사용한 객체의 조작은 객체를 생성하고 수정하며 디스플레이를 쉽게 한다.

모든 워크스테이션 소프트웨어는 아이콘, 마우스 중심의 사용자 인터페이스를 기본으로 한다. 비트맵 그래픽은 워크스테이션에서 다중 문자형태(font)와 스프레드시트, 플롯(plot)과 같은 고수준 그래픽을 허용한다. 또한 윈도우 메카니즘은 다중 도구를 사용할 수 있고 서로 다른 윈도우 사이에 자료를 전송할 수 있다. 윈도우 메뉴는 각 도구에 접근하는 명령어로 볼 수 있고 다른 보조형태의 메뉴도 제공할 수 있다[1]. 인터페이스가 성공하기 위해서는 직관적이고 접근이 쉬워야 하며 다른 시스템과의 프로토콜을 가져야 한다[6].

본 논문은 객체 지향 데이터 모델에 보안성을 결합한 보안 객체 관련성 모델링을 지원하기 위한 사용자 인터페이스의 설계와 구현을 목적으로 하였다.

본 논문에서는 2장에서 자동화 도구와 관련된 지금까지의 동향을 소개하고 3장에서 보안 객체 관련성 모델에 대해 기술하며 4장에서 편집기의 설계와 구현에 대해 설명하고 간단한 예를 들었다. 결론에서는 앞으로의 연구방향을 제시하였다.

2. 관련 연구

개념적 설계 단계는 효과적인 데이터베이스를 구현하기 위해 중요하다. 개념적 단계의 설계를 위한 효율적인 방법이 제시되지 못했을 때 설계자는 부주의한 작업에서 발생될 수 있는 오류로 인해 결과적으로 데이터베이스를 재구성 하여야 하는 결과를 초래한다. 이러한 오류를 줄이기 위한 개념적 스키마 설계 도구가 제안되고 구현되었다. 또한 고수준의 개념적 자료 모델은 실세계 객체와 개념적으로 설계된 객체와의 의미적 차이를 줄여야 한다. 이러한 개념적 단계의 설계 도구인 편집기의 설계는 Chen에 의해 제시된 도식적 관계형 모델에 기초하여 수행되어져 왔다[6].

관계형 데이터베이스의 사용자 인터페이스는 마우스와 비트맵 그래픽을 사용하여 데이터베이스 디스플레이가 이루어진다. Schemadesign 시스템[7]의 그래픽 표기는 기존의 관계형 시스템의 테이블 형태의 리스트보다 이해하기 쉽다. 또한 Schemadesign은 데이터베이스 관리자에 의해 설계되는 도구로 메시지 윈도우, 명령어 윈도우 등으로 되어있고, 삽입, 삭제, 수정 되었을 때 무결성을 유지할 수 있다. Schemadesign 도구는 객체를 정의하고, 각 객체에 대한 테이블을 생성하며 관련성을 정의하는 순서로 이루어진다[7].

시각 스키마 설계 도구 GS Designer[4]는 그래픽 객체 모델인 클래스와 관련성을 대화적으로 정의할 수 있는 도구이다. GS Designer는 객체지향 데이터베이스를 지원하고 마우스와 키보드 인터페이스, 비트맵을 윈도우 환경에서 사용하여 클래스를 정의하고 지을 수 있게 하였다.

GS Designer의 주요 구성 요소는 클래스 그래프이다. 클래스 그래프는 다양한 관련성에 의해 결합된 관련 클래스의 집합이다. 특별한 클래스는 한개 이상의 클래스 그래프에 포함되고 스키마는 많은 클래스 그래프를 포함한다. 연관성 표현은 일반화 관련성에 중점을 두고 구현되었다. 클래스 그래프는 응용 스키마의 모든 클래스를 논리적 조각으로 분해한다. 클래스 그래프에서 연관성은 선택적으로 디스플레이 되고 숨겨진다.

GS Designer의 구현 원리중 중요한 점은 타당치 않은 클래스 그래프를 생성할 수 없다는 것이다. 모든 클래스 그래프는 데이터베이스가 일관성을 유지할 수 있도록

특 표현되고 사용자는 부적합한 정보로 스키마를 그릴 수 없다.

GS Designer의 인터페이스 구성은 스키마 윈도우, 클래스 그래프 윈도우, 클래스 형식 윈도우로 구성된다. 스키마 윈도우는 스키마에서 표현되는 모든 클래스 그래프의 아이콘을 포함한다. 모든 스키마는 클래스 그래프에서도 표현될 수 있다. 클래스 그래프 윈도우는 사각형으로 클래스로 표현하고 화살표로 관련성을 표현한다. 굵은 단순 화살표와 이중 화살표는 클래스의 인스턴스 변수를 표현하고 일반성은 IS-A로 레이블된 화살표로 표현된다. 클래스 형식 윈도우는 문자적으로 클래스를 정의하는 윈도우로 클래스 이름과 인스턴스 변수, 변수 타입 등을 정의한다. GS Designer에서 지원하는 인터페이스 조작 방법은 클래스 그래프의 생성, 지우기, 데이터베이스에 스키마 저장, 스키마를 파일로 호출하기 등으로 구분된다.

3. 보안 객체 관련성 모델

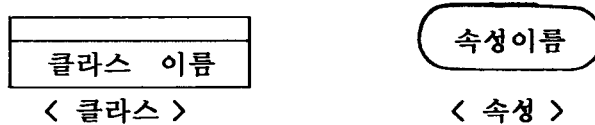
데이터 모델이란 데이터베이스 관리자가 사용하는 물리적 자료 구조가 아닌 데이터베이스 사용자의 추상 자료 구조이다. 이를 일반 사용자가 접근하고 조작할 때는 전문적인 프로그램의 도움이 필요없다. 초기 자료 모델은 자료 저장의 개념으로 화일이라는 용어를 사용했으며 관련성은 화일간의 접근 경로를 나타내었다. 후에 화일은 객체와 속성으로 세분화되어 유연성을 갖게 되었다[5].

보안 객체 관련성 모델(Secure Object-Relationship Model:SOREM)은 클래스와 속성(attribute)을 기본 객체로 하는 객체 데이터 모델(OMT)에 보안성(security)을 확장시킨 모델로 데이터의 무결성과 보안성을 동시에 표현할 수 있다.

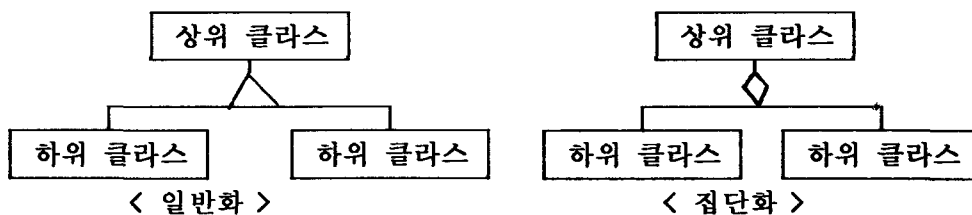
3.1 기본 보안 객체 모델의 표기법

보안 객체 모델에서 사용하는 기본 객체 요소의 표기법은 <그림 3.1>과 같다. 기본 객체인 클래스는 실세계 객체를 이해하는데 도움을 주고 실제 구현에 사용되는 단위로 속성과 트랜잭션(연산)을 포함한다. 이와 같은 클래스의 표현은 사각형으로 나타낸다. 속성은 클래스의 자료값을 표현하고, 속성의 표현은 타원으로 나타낸다. 객체 모델에서 클래스는 속성과 트랜잭션을 동시에 표현하지만 속성에 의미를 부여하기 위해 보안 객체 모델에서는 분류하여 표현한다.

일반화는 상속성(inheritance) 표현을 나타낸다. 일반화는 상위 클래스의 정보를 포함하는 하위 클래스에 대한 관계로 일반적으로 IS-A 관계로 표현한다. 집단화는 연관성의 특수한 형태이다. 하위 클래스는 상위 클래스의 구성 요소이다. 즉 하위 클래스가 상위 클래스를 구성하는 관계로 일반적으로 Part-of로 표현된다. <그림 3.2>에서 일반화 관계는 삼각형을 기호로 사용하여 표현하고 집단화는 마름모를 기호로 사용하여 도식적으로 나타내었다.



< 그림 3.1 > 기본 객체의 표현



< 그림 3.2 > 일반화와 집단화의 그래픽 표현

3.2 연관 무결성

연관 무결성(association integrity)를 표현하는 방법을 대응수(cardinality)와 존재성(existence)으로 구분하여 정의할 수 있다. 대응수는 어떤 객체가 다른 객체에 대해 대응하는 객체의 사상관계를 표현하는 것으로 일(one) 또는 다(many)의 값을 갖는다. 존재성은 한 객체가 다른 객체의 존재에 종속되는 관계를 나타낸다. 존재성은 선택(option)과 필수(mandatory)로 구분한다. <그림 3.3>은 이와같은 대응수와 존재성을 표현할 때 사용되는 표기법을 나타낸다.

대응수	존재성	필수	선택
일(one)			
다(many)			

< 그림 3.3 > 연관성 표현

3.3 보안성 표기법

보안성이 갖는 의미는 정보의 무분별한 접근으로 인해 발생하는 데이터베이스의 훼손을 방지하고 동시에 효율적인 정보의 관리를 목적으로 도입된다. 그러므로 기본 객체인 클래스와 속성을 포함하여 클래스와 속성간의 관계성도 보안성이 주어지고 클래스와 클래스간의 연관성 표현도 보안성을 부여하게 된다. 일반적으로 보안정책에서는 각 객체가 지니고 있는 정보의 비밀정보에 따라 비밀구분(security classification)이 부여되고 이러한 비밀 구분은 보안등급으로 언급되기도 한다. 또한 각 클래스는 어떤 범주(category)에 속해 있다. 이러한 보안등급의 부여는 데이터베이스 관리자에 의해 부여되고 일반 사용자는 보안등급을 부여할 수 없으며 사용자에게 부여된 보안등급에 한정해서 정보를 접근할 수 있다.

데이터베이스 관리자에 의해 부여되는 보안수준은 4단계로 U, C, S, TS가 있다. 여기에서 U(unclassified)는 보안의 필요성이 없음을 나타낸다. C(confidential)는 불법유출시 불리할 수 있음을 의미하고 S(security)는 불법유출시 피해가 있음을 뜻하며 TS(top security)는 불법유출시 심각한 피해가 있음을 나타낸다.

보안성의 표기 방법은 굵은선과 가는선을 이용하는데 U 등급은 가는선으로 표현하고 나머지 보안성을 갖는 등급은 굵은선으로 표현한다. 보안성을 갖게 되는 클래스와 속성 및 관련성의 표현은 <그림 3.4>에 나타내었다.

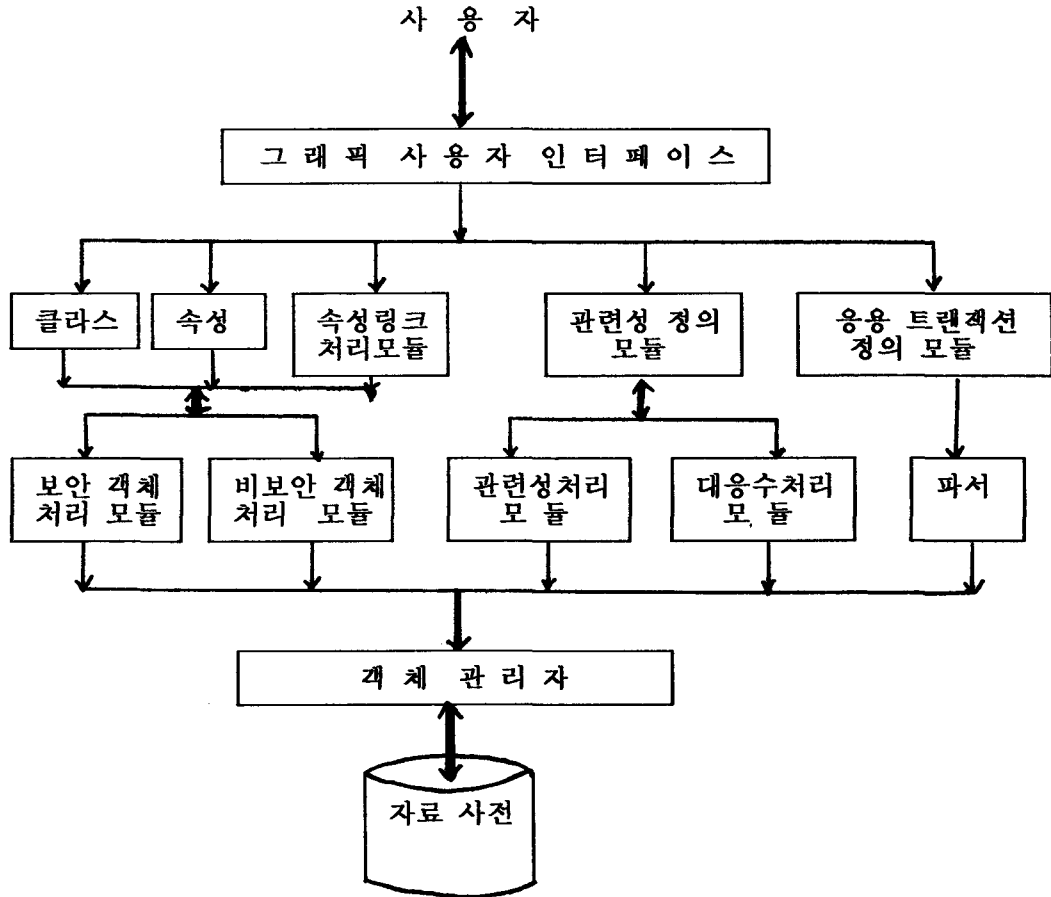
종류	보안 미분류	보안 분류
클래스		
속성		
관련성		

< 그림 3.4 > 보안성 표현

4. 보안 객체 관련성 모델 편집기

4.1 보안 객체 시스템의 구성도

본 논문에서 사용하는 시스템의 구성도는 <그림 4.1>과 같다. 객체 클래스와 속성은 처리 모듈에서 보안 객체와 비보안 객체 처리 모듈에서 처리되어 객체 관리자에 의해 데이터 사전에 저장된다. 저장된 정보는 수정과 첨가등의 명령을 행할 때에 새로운 정보로 갱신된다.



< 그림 4.1 > 시스템의 구성도

4.2 보안 객체 다이어그램 인터페이스

본 논문에서 사용하는 인터페이스는 보안객체를 모델링하기 위한 편집기로 이름을 SODE(Secure Object Diagram Editor)라 한다.

편집기의 구성은 다이어그램을 생성하고 관리하는 메뉴 영역과 클래스와 속성, 속성링크, 관련성 정의 방법 도구를 제공하는 도구 버튼 영역이 있다. 모델링된 다이어그램은 캔버스 영역에 그려진다. 캔버스 영역은 크기가 큰 다이어그램을 편집할 수 있게 스크롤 기능을 가지고 있다. 메시지 윈도우는 다이어그램을 수행하는 과정이나 오류 정보를 문자 정보로 나타내어 준다.

본 편집기의 기능을 설명하면 다음과 같다.

가. 객체 생성 기능

기본 객체인 클래스와 속성의 정의는 다음과 같다.

Editing 메뉴에서 **create** 버튼을 선택하면 도구 버튼 영역에 정의 버튼이 나타난다. 도구버튼에서 클래스 정의 버튼을 선택하면 <그림 4.2>와 같은 대화 윈도우가 생성된다. 클래스의 이름을 입력하고 클래스가 속한 범주에 대한 정보를 입력한다. 범주에 관한 정보는 반복적으로 입력될 수 있다. 정보 입력이 끝났음을 Done 버튼으로 확인하고 원하는 그리기 영역을 선택하여 심볼을 그리면 클래스 정의가 끝난다. 이때 심볼의 크기는 일정한 크기로 고정되지 않고 클래스 이름의 길이에 따라 사용자가 임의로 조정할 수 있다. 범주에 관한 정보는 테이블로 저장되어 시스템에서 주어지는 사용자 보안등급과 범주에 관한 정보와 함께 사용자가 정보 접근을 시도할 때 정보 허가의 평가 기준이 된다.

Class definition	
Class name	<input type="text"/>
Category	<input type="text"/>
<div style="display: flex; justify-content: space-around;"> Done Cancel </div>	

< 그림 4.2 > 클래스 정의 대화상자

이와 같은 객체를 정의하는 과정에서 메시지 영역에서는 다음에 행해져야 할 작업을 문자 정보로 보여주어 현재 진행과정을 알수 있게 한다.

속성 정의 단계는 도구 버튼에서 속성 정의 버튼을 선택하면 <그림 4.3>과 같은 대화 윈도우가 나타난다. 클래스 인덱스 정보는 이 속성이 속하는 클래스의 이름이고 속성의 이름과 속성이 갖는 보안등급을 부여하여 속성을 정의하고 승인 버튼을 선택하고 원하는 위치에 속성을 그려준다. 보안 분류를 정의하는 입력의 종류는 Top security인 경우 t를 Security는 s를 그리고 각각 Certified는 c를 Uncertified는 u를 입력하게 된다. 이러한 정보는 클래스와 동일하게 메시지 영역에 나타난다. 입력된 보안 분류에 따라 u(비보안 분류)는 가는선으로 표현되고 나머지 t, s, c(보안 분류)는 굵은선으로 표현된다. 속성 심볼도 클래스 심볼과 동일하게 가변적인 크기를 가진다. 또한 속성 심볼은 보안등급이 U인 경우 가는선으로 나타나고 보안 등급이 부여 되었을 때 굵은 선으로 나타난다.

Attribute definition	
index class	<input type="text"/>
Attribute name	<input type="text"/>
Secure Level	<input type="text"/>
<div style="display: flex; justify-content: space-around;"> DONE CANCEL </div>	

< 그림 4.3 > 속성 정의 대화상자

나. 속성링크의 정의

속성링크의 정의는 클래스와 속성간의 관계성을 표현하는 것으로 역시 보안 분류와 비보안 분류로 정의 된다. 버튼 영역에서 속성 링크 버튼을 선택하면 <그림 4.4>와 같은 링크 정의 대화상자가 나타난다. 이때 입력되는 정보는 보안 분류의 정의로 속성의 보안 분류 정의와 같은 방법으로 입력하면 된다. 또한 속성링크의 표현도 속성의 표현과 동일하다.

Linking definition	
Linking secure level <input style="width: 80%;" type="text"/>	
Done	Cancel

< 그림 4.4 > 연관성 정의 대화상자

다. 관련성 정의

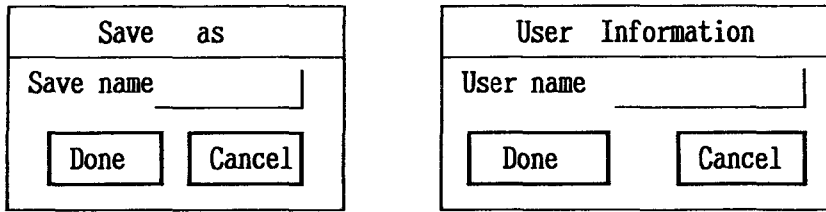
관련성 정의는 클래스간의 관련 정보를 나타낸다. 관련성의 형태는 일항, 이항, 삼항 연관성과 일반화, 집단화 등이 있다. 관련성 정의 버튼이 선택되면 <그림 4.5>와 같은 관련성 정의 대화상자가 나타난다. 정의 박스에 정의하고자 하는 관련성 정보를 입력하면 된다. 관련성 정의 형태도 보안등급의 정의와 마찬가지로 일항인 경우 대문자 U를 이항인 경우 B, 삼항은 T, 일반화 관련성은 G, 집단화 관련성은 A를 입력하는데 이에 대한 문자 정보가 메시지 영역에 표시된다.

Relationship definition	
Relationship Type <input style="width: 80%;" type="text"/>	
Done	Cancel

< 그림 4.5 > 관련성 정의 대화상자

관련성 정의는 관련성에 참여하는 클래스와 대응수 정의를 포함한다. 이와같은 작업은 관련성 타입의 정의후에 클래스 인덱스를 구하고 대응수를 정의하는 순서로 이루어진다. 클래스 인덱스의 정의는 그리기 영역에 정의된 클래스를 마우스로 클릭하면 시스템에 의해 주어진 클래스 인덱스 정보가 얻어진다. 선택된 클래스의 대응수의 정의는 원하는 대응수 심볼 이름을 대화 윈도우를 통해 입력하고 원하는 위치에 심볼을 클릭한다. 대응수 심볼의 입력형태는 대응수가 일이고 존재성이 필수일 때는 CM을 입력하고, 대응수가 일이고 존재성이 선택일 때는 CO를 입력한다. 또한 대응수가 다이고 존재성이 필수일 때 NCO를 입력하고 대응수가 다이고 존재성이 필수일 때 NCO를 입력한다. 대응수 선택 대화상자는 관련성 정의 대화상자와 동일한 형태이다. 마지막으로 관련성 심볼을 원하는 위치에 클릭하면 관련성 정의가 끝난다.

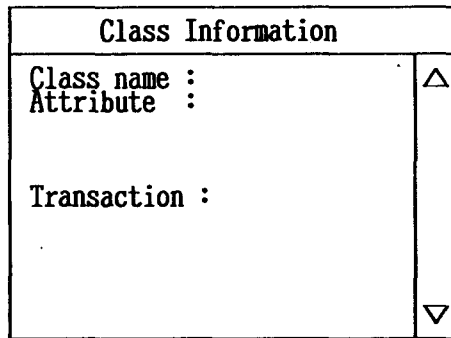
System View 메뉴에서 실행되는 작업은 편집기를 통해 작성된 그래픽 다이어그램을 시스템에 저장하는 저장 메뉴와 사용자가 원하는 정보를 얻기 위해 접근하는 사용자 정보 메뉴의 두가지로 <그림 4.6>과 같은 두가지 중속 메뉴로 구성된다.



< 그림 4.6 > System view의 종속 메뉴

User Information 메뉴의 사용은 먼저 모델링 정보를 원하는 사용자가 사용자 이름을 입력한다. 입력된 사용자 이름은 데이터베이스 관리자에 의해 주어진 사용자의 보안등급과 범주(category)를 이용하여 클래스 정의때 주어진 클래스의 범주와 각 클래스의 보안등급에 따라 사용자에게 허용된 정보를 다이어그램으로 그리기 영역에 보여준다.

Information 메뉴는 정의된 클래스의 속성과 트랜잭션에 대한 정보를 얻기 위해 사용되는 메뉴로 사용자가 입력한 클래스 이름과 해당 클래스에 대한 속성과 트랜잭션을 보여준다. <그림 4.7>과 같은 메뉴를 통해 입력된 클래스 속성과 트랜잭션에 대한 정보를 볼수 있다.



< 그림 4.7 > 클래스 정보 윈도우

Utility 메뉴는 객체 재정의하기, 지우기, 이동, 복사하기, 관련성 재정의하기를 할 수 있다. 이 경우 속성에 대해 지우기가 행해졌을 때는 속성과 속성 링크가 같이 지워지고 클래스에 대한 지우기가 실행되면 그에 따르는 속성도 모두 지워진다. 또한 다이어그램시 객체의 크기를 확대하고 축소할 필요가 있을 때의 기능도 포함한다.

4.3 객체 관리자

객체 관리자는 각 정의 모듈에서 정의된 객체의 보안등급과 관련 정보를 관리하는데, 클래스 정의 모듈에서 정의된 클래스 보안등급과 범주에 관한 정보를 관리한다. 클래스와 속성의 보안등급은 구간으로 관리한다. 임의의 사용자가 객체 관리자에 의해 관리되는 정보 접근을 시도하였을 때, 사용자의 보안등급과 범주 정보를 비교하여 접근이 허용된 정보만을 제공한다.

4.4 보안 객체 모델의 데이터 사전

하나의 객체가 생성될 때마다 하나의 자료구조가 생성한다. 이러한 자료구조는

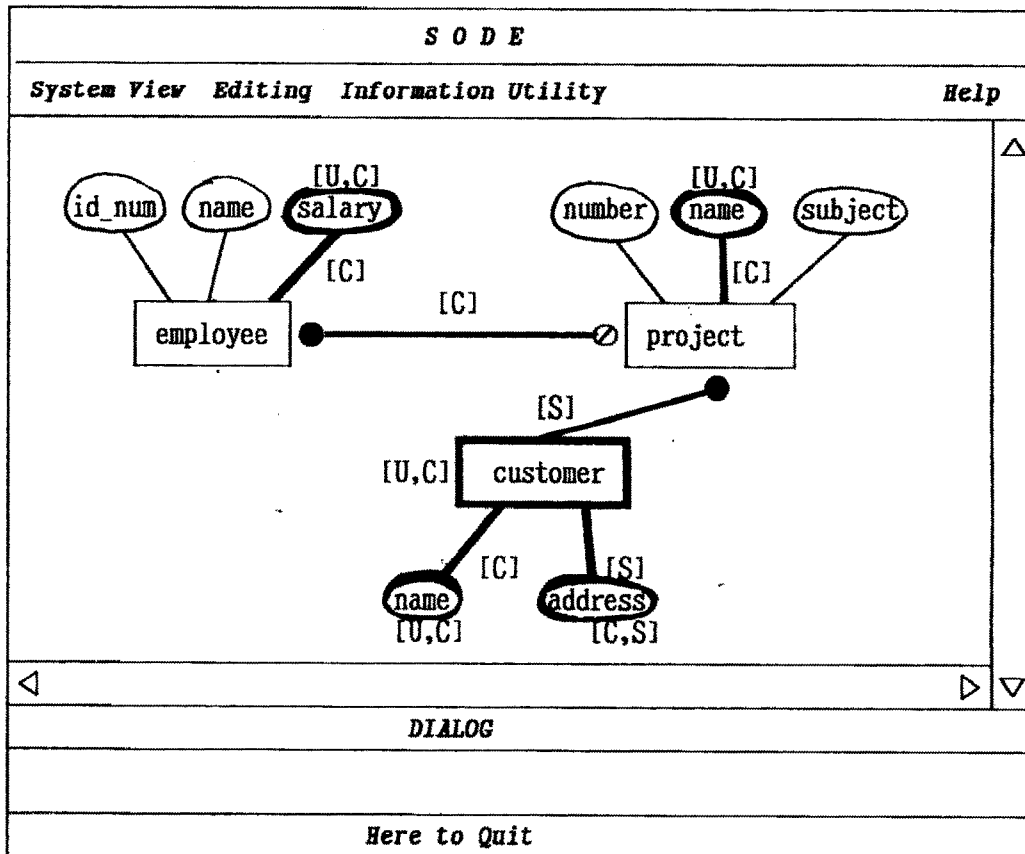
연결구조(linked list) 형태로 구성되었다. 시스템에서 사용되는 데이터 사전의 구성은 자료의 형태에 따라 크게 클래스, 속성, 속성 링크, 관련성으로 구분된다. 클래스와 속성의 자료구조는 보안등급 정보와 이름을 갖고 그래픽 정보를 제공하기 위한 위치를 텍스트 정보로 갖게 된다. 속성 링크의 자료구조는 보안 등급을 가지고 속성 자료구조에 연결되어 있다. 관련성 자료구조의 구성은 관련성 형태에 따라 일항, 이항, 일반화, 집단화, 삼항으로 구분되고 각각의 관련성 정보는 대응수와 대응되는 클래스의 인덱스를 갖고 있다.

4.5 보안 계층 모델링의 예

<그림 4.8>은 위에서 설명한 기본 도구를 사용하여 모델링 한 예를 보여준다.

Editing 메뉴에서 **Create**를 선택하면 도구 버튼이 나타나고 클래스 정의 버튼을 선택하여 클래스를 정의한다. 각 클래스의 이름과 범주에 관한 정보를 입력하고 그리기 영역에 클래스를 그린다. 클래스 정의가 끝나면 속성을 정의한다. 속성의 클래스 인덱스와 속성 이름, 보안등급을 입력하고 그리기 영역에 그린다. 이때 속성의 보안 등급이 속성 이름과 함께 나타난다. 속성링크의 정의는 속성 링크의 보안 등급을 입력하면 속성과 같은 방법으로 그린다. 속성과 속성링크의 정의가 모두 끝나면 관련성을 정의한다.

예제에서는 이항 관련성을 정의하고 클래스의 인덱스를 선택하여 클래스가 관련성에서 참여하는 대응수를 선택한다. 예제에서는 **employee** 클래스를 선택하고 대응수를 **C0**로 입력하면 <그림 4.8>과 같이 그려진다. 관련성에 참여하는 다른 클래스도 동일한 방법으로 정의한다. 마지막으로 관련성 보안등급을 입력하고 그리면 모델링 작업이 끝난다.



< 그림 4.8 > 모델링의 예

예제가 갖는 보안 성질은 다음과 같다. 트랜잭션은 편의상 생략하였다. 예제에서 객체 클래스는 고용인, 프로젝트, 고객 등 3가지이다. 고용인 객체 클래스는 3가지 속성을 갖고 있다. 고용인 객체 클래스와 속성 봉급 사이의 연관성 보안등급은 C이다. 따라서 어떤 특정 고용인의 봉급을 볼 수 있는 사용자는 보안등급이 C 이상이어야 한다는 것을 의미한다. 속성 봉급의 보안등급이 C이고 객체 클래스 고용인의 보안등급은 U이므로 연관성 보안등급은 두 객체 보안등급 중 큰 것보다 같거나 높아야 한다는 것을 의미한다. 다른 객체 클래스와 속성과의 관계도 같다. 또 객체 클래스 고용인과 프로젝트 사이의 연관성 보안등급은 C이다. 즉, 고용인의 주민등록번호와 프로젝트 번호의 연관성 보안등급이 C라는 의미이다. 여기에서 고용인의 주키는 주민등록번호이고 프로젝트의 주키는 번호이다.

5. 결 론

본 논문에서는 보안 객체 다이어그램으로 현실세계를 모델링하는 그래픽 처리기를 X 윈도우 시스템을 이용하여 구현하였다. 시스템은 마우스를 통한 메뉴 방식의 명령어 처리와 윈도우 처리, 기호 처리를 통해 사용자에게 편의성을 제공하고 이해하기 쉬운 그래픽 사용자 인터페이스로 구현되었다. 또한 모델링된 정보는 보안 등급 관리자에 의해 등급이 관리되고 접근된 사용자는 주어진 권한에서만 정보를 검색하고 수정할 수 있도록 하였다.

본 논문은 모델링을 고수준의 개념적 단계에서 설계하였고, 그래픽 표현을 사용하여 사용자에게 이해하기 쉽고 친숙한 인터페이스를 만들었다. 앞으로 행해져야 할 과제는 사용자가 해야 할 메뉴 방식의 문자 정보 입력을 아이콘을 사용하여 처리하고, 여러 사용자가 동시에 사용할 수 있는 개방형 시스템 환경을 제공하는 것이다.

참고 문헌

- [1] Arthur J. Benjamin, Karl M. Lew, "A visual Tool for managing Relational database", IEEE 2nd Data Engineering, 1986, pp. 661-668.
- [2] D. A. Young, The X Window System Programming and Application with OSF/Motif edition, Prentice-Hall, 1990.
- [3] Jay Almarode, "Issues in the Design and Implementation of a Schema Designer for an OODBMS", ECOOP'91 European Conference on Object-Oriented Programming, pp. 200 - 218.
- [4] Levent ORMAN, "A visual data Model", Data & Knowledge Engineering, 7. 1992, pp 227-238
- [5] Prabhat K. Andleigh, Micheal R. Gretzinger, Object-oriented Distribute Data-System Design, Prentice Hall 1992
- [6] F. H. Lochovsky, Entity - Relationship Approach to Database design and Querying, ERI, 1990.
- [7] T. R. Rogers, R. G. G. Cattell, "Entity - Relationship Database User interface", Data Engineering, June. 1988. VOL. 11 No 2, pp 44 - 53.
- [8] 전길남, 한선영, "객체지향 사용자 인터페이스", 10, 1990, 정보과학회지 제 8권 제 5호, pp 52-60.