

동적 객체지향 모델에서 무결성과 임무 분리 메카니즘

서재현* · 김영균** · 노봉남**

* 송원전문대학 전자계산학과

** 전남대학교 전산학과

A Mechanism of Integrity and Separation of Duty in an Active Object-Oriented Data Model

Jaihyun Seo*, Youngkyun Kim** and Bongnam Noh**

* Dept. of Computer Science, Songwon Junior College

** Dept. of Computer Science, Chonnam National University

요 약

객체지향 데이터베이스 시스템에서 무결성은 중요한 개념이다. 이 무결성이 객체의 내부 메소드로 정의되고 구현하면 응용에서의 무결성 의미 파악이 어렵고, 또한 객체의 기본적인 메소드와 함께 모형화되기 때문에 데이터베이스 설계 작업이 복잡해진다. 본 논문에서는 무결성을 메소드로 처리할 때의 문제점을 극복하기 위하여 동적 데이터베이스의 동적 규칙을 이용하여 무결성 제약조건을 정의하고 유지시키는 객체지향 데이터베이스에서 무결성과 임무 분리를 유지시키는 기법을 제안한다.

1. 서론

객체지향 프로그래밍에서 시작되는 객체지향 패러다임(paradigm)은 수년동안 데이터베이스, 소프트웨어 공학, 인공지능, CAD/CAM 등과 같은 분야에서 활발하게 연구되었고, 최근에는 데이터베이스 분야에서 표준화된 객체지향 데이터 모델이 제시되고 있는 실정이다. 객체지향 시스템에서 기본적인 요소인 객체는 현실 세계의 개체, 관련성, 현상들을 표현하고 있으며, 개체의 상태와 행위를 모두 캡슐화하고 있는 특징을 갖는다. 객체지향 데이터베이스에서 캡슐화가 갖는 의미는 객체의 상태에 대한 변경이 객체가 포함하고 있는 행위 즉, 메소드라는 인터페이스를 통해서만 가능하기 때문에 객체지향 시스템에서 무결성은 중요한 구성요소이다. 즉, 객체의 무결성 제약조건은 메소드를 통해서만 속성값에 적용될 수 있다.

무결성은 데이터가 비일관된 상태로 변경되는 것을 방지하며, 또한 데이터 내용이 불법적으로 노출되는 것을 보호하는 개념이다. 그러므로 무결성은 데이터의 일관성을 유지할 뿐만 아니라 데이터 비밀성도 포함하는 광의의 개념으로 정의될 수 있다. 일반적으로, 데이터베이스 시스템에서 무결성 제약조건은 사용자에 의해서 정의되고, 객체를 정의하는 클래스를 생성할 때 이 무결

성이 적용된다[5].

객체지향 데이터베이스에서 무결성 제약조건은 상당히 중요한 개념이지만, 현존하는 무결성 모델들 즉, Biba 무결성 모델과 Clark/Wilson 모델을 객체지향 시스템으로 적용한 연구는 거의 없는 실정이며, 또한 무결성 제약조건을 정의하고 이것을 객체에 적용하는 메카니즘을 포함하는 객체지향 데이터베이스 관리 시스템은 아직까지 개발되지 않았다. 단지, Clark/Wilson의 무결성 모델을 객체지향 환경으로 적용한 연구가 Herndon에 의해서 수행되었다[11].

Herndon은 Clark과 Wilson이 제시한 무결성 유지 메카니즘에 사용된 개념들을 객체지향 데이터베이스 환경으로 적용하기 위하여 제한 클래스(restricted class) 개념을 정의하였고, 이 클래스들에서 무결성을 유지하기 위한 수단으로 권한부여 릴레이션(authorization relation)을 이용하였다[11]. 특히, 객체지향 데이터베이스 시스템에서 임무 분리가 유지되는 권한부여 릴레이션의 예를 제시하였다. 이러한 권한부여 릴레이션에는 사용자가 접근할 객체 집합에 대한 연산이 그 객체의 내부 메소드인 변환 프로시저어를 의미한다. 따라서 변환 프로시저어에 의해서 객체의 무결성은 유지된다.

그러나 객체에 대한 무결성이 그 객체의 내부 무결성 메소드로 정의되어 구현된다면, 실제로 응용에 대해서 정의된 무결성의 의미를 파악하기 어려울 뿐만 아니라 객체의 기본적인 메소드와 함께 모형화하기 때문에 데이터베이스의 설계 작업이 복잡해진다. 또한 정의된 무결성들을 변경할 때에도 무결성 메소드들을 변경하므로써 데이터베이스의 비일관된 상태를 발생시킬 수 있는 문제점들이 존재한다.

따라서 본 논문에서는 이러한 문제점들을 극복할 수 있는 방안으로서, 동적 데이터베이스 시스템의 동적 규칙 개념을 이용하여 무결성 제약조건을 정의하고, 유지시키는 메카니즘을 제시하였다. 제안된 무결성 기법은 무결성을 유지시키기 위한 메소드가 필요치 않으며, 대신에 무결성 메소드를 객체의 기본적인 메소드와 분리하여 하나의 독립적인 규칙 객체로 처리하므로써 무결성의 정의를 그래픽 다이어그램을 이용하여 쉽게 수행하고, 또한 동적 데이터베이스 시스템의 기능을 이용하여 무결성과 임무 분리를 자동적으로 유지할 수 있는 장점을 갖는다.

2장에서는 Clark/Wilson의 무결성 모델을 간략히 설명하였고, Herndon의 연구를 살펴보았다. 3장에서는 본 논문에서 이용하는 동적 객체지향 모델을 정의하였고, 4장에서는 무결성과 임무 분리를 유지시키는 무결성 유지 기법을 설명하며 예를 제시하였다. 마지막으로 5장에서는 본 논문의 결론과 앞으로의 연구 방향을 언급하였다.

2. 관련 연구

2.1 Clark/Wilson 무결성 모델

상용 시스템에서 무결성을 유지하기 위한 모델을 1987년도에 Clark과 Wilson이 소개하였다 [3]. 이들이 제시한 무결성 모델에서는 가장 중요한 두가지의 기본적인 개념들 즉, 정형화 트랜잭션(well-formed transaction)과 임무 분리(seperation of duty)를 정의하였다[3, 6, 11]. 일반적인 임무 분리의 개념은 특정한 응용 트랜잭션이 수행될 때, 이 트랜잭션에서 한명의 사용자가 두가지 이상의 역할을 수행하지 못하도록 한다는 의미를 갖는다. 즉, 임무 분리를 Clark/Wilson 모델에서의 정형화 트랜잭션에 적용할 때 정형화 트랜잭션을 정의한 사용자는 이 트랜잭션을 수행할 수 없는 개념이 바로 임무 분리에 해당한다.

Clark/Wilson의 무결성을 지원하는 시스템에서 모든 데이터는 두가지의 범주 즉, 제약 데이

타 항목과 비제약 데이터 항목으로 구분된다. 제약 데이터는 무결성 정책이 적용되는 데이터이며, 이 데이터에 대해 수행되는 정형화 트랜잭션이 적용될 수 있는 특징을 갖는다. 반면에, 비제약 데이터는 무결성에 의해서 통제되지 않는 데이터들을 의미한다.

그리고 이 모델에서는 제약 데이터의 무결성이 유지되는 것을 보장하기 위하여 인가(certification)와 시행(enforcement) 기법을 이용하였다. 인가는 정형화 트랜잭션에 적용되며, 그림 1의 (a)에서와 같이 변환 프로시저어와 제약 데이터 항목으로 구성된 릴레이션이다. 변환 프로시저어는 제약 데이터를 초기 상태에서 최종 상태까지 무결성이 위반되지 않게 유지시킨다. 대부분, 보안 관리자나 대행자에 의해서 인가 프로세스가 수행되며, 인가 릴레이션이 생성되고, 이때에 임무 분리의 개념이 변환 프로시저어에 적용된다.

무결성 시행은 권한이 주어진 사용자와 인가 릴레이션에서 정의된 변환 프로시저어를 결합시키는 절차이다. 따라서 Clark/Wilson 모델이 적용된 시스템에서는 그림 1의 (b)와 같은 릴레이션이 필요하다.

(TP_i, CDI₁, CDI₂, ..., CDI_n) ; TP_i : 변환 프로시저어
 CDI_n : 제약 데이터 항목

(a) 인가 릴레이션

(userID, TP_i, CDI₁, CDI₂, ..., CDI_n) ; userID : 사용자
 (b) 시행 릴레이션

<그림 1> Clark/Wilson 모델의 무결성을 위한 릴레이션

2.2 Herndon의 연구

Herndon은 Clark/Wilson 모델에서 무결성을 위한 개념들을 동일한 기능들을 수행할 수 있는 객체지향 데이터베이스의 구조들로 변환시켰다. 먼저, Clark/Wilson 모델에서의 제약 데이터 개념을 유지시키기 위하여 제한 클래스를 정의하였는데, 이 클래스는 무결성이 유지되는 클래스를 의미한다. 즉, 제한 클래스의 모든 메소드들은 이 클래스의 객체 인스턴스의 무결성을 보장하도록 인가된다. 그리고 무결성 시행 릴레이션은 Clark/Wilson의 권한부여 릴레이션을 그대로 사용하였다.

제한 클래스는 클래스 계층구조에서 상위 제한 클래스에 정의된 속성과 메소드를 상속받고, 제한되지 않은 상위 클래스에서 보내진 메시지에 대해서는 결코 반응하지 않는다. 또한 상위 클래스가 제한되지 않을 경우는 그 클래스의 메소드를 호출하지 않는 특징들을 갖는다. 그리고 인가 릴레이션에서의 변환 프로시저어는 제한 클래스의 메소드 개념으로 대체되며, 무결성을 유지시키는 메소드는 다른 메소드들과 구분하여 독립적으로 정의하였다.

이 연구에서는 클래스를 생성할 때, 세가지의 클래스 생성 방법을 아래와 같이 제안하였는데 이러한 클래스 생성 규칙은 제한 클래스를 활용하기 위한 기본적인 바탕을 이루고 있다.

< 클래스 생성 방법 >

[CCT 1] 상위 제한 클래스의 무결성 메소드와 하위 클래스의 무결성 메소드는 분리되어 정의한다.

[CCT 2] 제한 클래스에서 제한되지 않은 하위 클래스를 생성할 때, 제한 클래스를 복제하여 이 클래스를 상속받도록 한다.

[CCT 3] 무결성 제약조건을 검사하는 메소드는 객체의 상태를 변경시키는 메소드와 구분되어야 하고, 그들 자신의 메소드로 구현한다.

Herndon이 제시한 클래스 생성 방법을 고려할 때, 가장 중요한 문제점은 무결성을 유지시키기 위해서 클래스내에 무결성 메소드로 정의한다는 것이다. 이러한 방법은 무결성 의미를 파악하기 어려울 뿐만 아니라 정의된 무결성을 변경할 때 그 작업이 복잡해진다. 따라서 무결성을 객체의 메소드로 정의 또는 구현하지 않고, 선언적 제약조건 형식을 사용하며 시스템에서 자동적으로 유지될 수 있도록 동적 데이터베이스 시스템의 기능을 이용하는 것이 바람직하다. 따라서 본 논문은 동적 데이터베이스를 이용하여 Clark/Wilson 모델의 무결성을 정의하고, 유지시키는 방법을 제시한다.

3. 동적 객체지향 데이터 모델

실시간 응용 분야들 즉, 통합 생산 자동화, 분산 네트워크 관리, 항공 교통 제어 시스템 등에서는 중요한 상황이 발생할 때, 시기적절하게 대처할 수 있는 기능이 필수적으로 제공해야 한다. 따라서 실시간 응용들에서는 데이터베이스 상태를 감독하고, 정의된 조건을 만족시키는 연산이 수행되었을 때 이에 대한 적절한 행위를 수행하는 능동적 기능을 포함하는 동적 데이터베이스 시스템이 필요하다[1, 4, 7, 8].

또한 동적 데이터베이스는 데이터 생성 규칙이나 무결성, 그리고 보안성 등에서도 자동적으로 유지시키는 기능들을 지원할 수 있다. 이 장에서는 데이터의 무결성을 정의하고, 유지시키는 기법의 기본적인 틀로서 동적 객체지향 데이터 모델을 소개한다.

먼저, 동적 객체지향 데이터 모델은 크게 데이터 구조를 표현하는 정적인 측면과 데이터의 연산 또는 무결성 제약조건을 표현하는 동적 규칙으로 구성된다. 정적인 측면은 객체 클래스와 관련성 등과 같은 데이터의 추상화 구조를 포함하고 있으며, 특히 상속성과 집단화 구조가 정의된다. 본 논문에서는 데이터의 구조에 대한 사항은 고려하지 않고, 무결성 제약조건을 정의하고 유지시키는 기법에 관심을 갖기 때문에 동적 규칙의 구성요소들을 간략히 설명한다.

3.1 동적 규칙의 구성

제시된 동적 객체지향 모델의 동적 규칙은 HiPAC 시스템[10]의 사건-조건-행위 규칙을 기반으로 하며, 동적 규칙의 구성요소를 사건과 규칙으로 분리하여 처리한다. 따라서 객체지향 환경에 적합하도록 사건과 규칙을 각각의 독립된 객체로 취급한다.

동적 규칙의 첫번째 구성요소인 사건은 데이터베이스 상태를 변경시키는 데이터 갱신 연산과 데이터베이스 외부에서 발생하는 외부 사건으로 구분하지만, 데이터베이스의 무결성을 고려할 때는 데이터의 갱신 연산만을 사건으로 고려한다. 대부분의 관계형 데이터베이스 시스템에서 갱신 사건은 삽입, 삭제, 변경 연산으로 표현되지만 객체지향 데이터베이스에서는 객체 클래스에 정의되어 있는 메소드에 의해서 객체의 상태가 변경되므로 메소드의 실행을 변경 사건으로 정의한다. 따라서 동적 규칙의 사건 객체를 다음과 같이 정의한다.

【정의 1】 사건 객체 클래스

사건 클래스를 C_E 라고 표시할 때, 사건 객체 클래스를 다음과 같이 정의한다.

$C_E = (E_{id}, D, N_C, N_M, \{M_E^i\}), i = \{ 1, 2, \dots, N \}.$

E_{id} : 사건 클래스 식별자, D: BEFORE, AFTER 구분 지시자

N_C : 클래스 이름, N_M : 메소드 이름, $\{M_E^i\}$: 사건 클래스의 메소드 모임

두번째 요소인 규칙 객체는 사건이 발생했음을 탐지하고, 규칙에 정의된 조건이 만족되면 데이터베이스에 대한 연산을 실행하는 기능을 갖는다. 규칙 객체는 사건, 조건, 행위, 그리고 결합형태로 구성된다. 사건은 규칙에 영향을 미치는 변경 사건을 의미하고, 조건은 무결성 또는 보안성, 그리고 상업적인 규칙에 대한 제약조건을 정의하고, 행위는 데이터에 해당하는 객체와 객체의 속성 그리고 관련성에 대해 수행되는 일련의 연산인 메소드로 구성된다. 그리고 결합 형태는 규칙이 실행되는 환경을 정의하는 부분으로서 즉시 수행, 나중 수행, 그리고 독립 수행이 있으며, 규칙의 수행 모델과 밀접하게 관계되어 있다.

【정의 2】 규칙 객체 클래스

규칙 객체를 C_R 로 표기할 때, 규칙 객체를 다음과 같이 정의한다.

$C_R = (R_{id}, A_R, C, B, CM).$


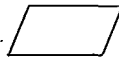

R_{id} : 규칙 객체 식별자, A_R : 규칙 객체의 속성

C: 규칙의 조건, B: 규칙 메소드, CM: 결합 형태

3.2 동적 규칙의 그래픽 표현과 서술 구조

제시된 데이터 모델은 데이터베이스의 동적 규칙을 사용자가 쉽게 정의할 수 있도록 동적 규칙의 그래픽 표기와 동적 규칙의 서술 구조를 제공한다. 기존의 관계형 모델에서 참조 무결성을 유지하기 위해서는 질의어 또는 응용 트랜잭션에 프로그래밍 언어를 이용하여 사용자가 정의하기 때문에 무결성 의미가 쉽게 이해되지 않았다. 이러한 문제점은 무결성에 해당하는 동적 규칙을 그래픽 표기를 이용하여 개념적으로 정의함으로써 해결될 수 있다.

동적 규칙의 그래픽 다이어그램은 그림 2에서와 같이 사건 클래스는 원으로, 규칙 클래스는 평행사변형으로 표현하고, 사건과 규칙 클래스사이에서 사건과 규칙의 관련성을 단방향 화살표로 표현한다. 이 화살표는 단방향 화살표의 뒤쪽에 위치한 사건 객체는 앞쪽에 위치한 규칙 객체를 활성화시킨다는 것을 의미한다. 그러므로 사건 객체는 적어도 하나 이상의 규칙 객체와 서로 연결되어야 한다.

종 류	그래픽 표기
사 건	
규 칙	
사건과 규칙 연 관 성	

<그림 2> 동적 규칙의 그래픽 표기법

동적 규칙의 그래픽 다이어그램은 개념적 또는 직관적으로 모형화된 규칙의 의미를 파악할

수 있으나, 규칙의 정확하고 세밀한 정보는 표현하지 못한다. 따라서 동적 규칙의 서술 구조가 지원되므로써 완벽하게 동적 규칙이 정의된다. 그러므로 이 서술 구조는 사건과 규칙 객체를 통합하여 전체적인 동적 규칙의 의미가 표현하는 구조를 갖으며, 서술 구조에 명세되는 자세한 내용은 수학적인 술어 형태로 서술된다.

ACTIVE RULE	rule_identifier
EVENT	class.method_name;
CONDITION	condition predicate;
ACTION	(raise reject permit) class.method;
COUPLING	(immediate deferred seperate);

<그림 3> 동적 규칙의 서술 구조

4. 무결성 유지 기법

[3, 11]에서 무결성의 정의는 인가 릴레이션에 의해서 정의되고 그리고 무결성 시행은 권한부여 릴레이션을 이용한 시스템에 의해서 수행되었으나, 본 논문에서의 무결성 정의와 시행은 하나의 동적 규칙으로 처리된다.

먼저, Clark/Wilson의 무결성 모델의 기본적인 인가 릴레이션에 있어서, 인가 릴레이션은 제약 데이터 항목에 대해서 적용되는 변환 프로시저어를 정의한 릴레이션이다. 즉, 무결성 제약사항을 정의한 릴레이션이므로 이 릴레이션은 3장에서 제시한 동적 객체지향 모델의 동적 규칙으로 정의할 수 있다. 본 논문에서는 제약 데이터 항목에 대한 개념이 필요치 않으며, 모든 객체 클래스를 동일하게 취급한다. 따라서 인가 릴레이션의 제약 데이터 항목은 데이터베이스의 어떠한 객체도 될 수 있다.

Clark/Wilson의 인가 릴레이션과 동일한 의미를 갖는 무결성은 동적 규칙으로 정의하는 것은 다음과 같은 방법에 따라서 수행된다.

[무결성 정의 방법]

- (1) 객체의 생성과 변경, 메소드 실행 또는 인가 릴레이션의 메소드 이름과 같은 무결성을 유지시키는 조건을 사건으로 정의한다.
- (2) 인가 릴레이션에서 변환 프로시저어인 무결성 메소드의 내용을 규칙 객체의 조건절에 선언적 제약조건으로 서술한다.
- (3) 규칙 객체의 행위 부분에 무결성을 유지되면, 메소드의 실행을 허용하거나 또는 이 메소드에 의해 무결성이 유지되지 않으면, 메소드의 실행을 취소하는 연산을 정의한다.

즉, 인가 릴레이션을 동적 규칙으로 정의할 때, 인가 릴레이션에서 무결성을 검사하는 변환 프로시저어는 필요치 않고, 단지 무결성을 유지시키는 조건 즉, 객체가 새롭게 생성되거나 또는 기존의 객체의 내용을 변경하는 메소드를 동적 규칙의 사건 객체로 정의한다. 특별한 경우에는 인가 릴레이션의 변환 프로시저어의 이름이 사건 객체로 표현될 수 있다. 무결성의 조건은 규칙 객체의 조건절에 선언적으로 서술되며, 무결성의 조건은 실제로 인가 릴레이션에서 메소드에 구현된 정보와 동일한 의미를 표현한다.

그리고 상위 클래스와 연관된 무결성 동적 규칙은 클래스 계층구조에서 하위 클래스에도 상

속되어 적용되며, Herndon의 상위 클래스의 무결성 메소드를 상속받고, 또한 자신의 무결성 메소드를 정의한 방법과 같이 하위 클래스는 자신만의 무결성 동적 규칙과 결합될 수 있다.

한편, Clark/Wilson 무결성 모델의 두번째 요소인 권한부여 릴레이션은 특정한 권한부여 클래스로 정의하여 유지시킨다. 이 권한부여 클래스는 각각의 사용자가 수행할 수 있는 메소드들의 리스트를 유지한다. 따라서 이 클래스는 무결성 동적 규칙이 실제로 수행될 때, 사용자가 실행한 메소드가 정의된 무결성을 유지시키거나 또는 임무 분리를 유지하는가를 검사하는 기본적인 클래스의 역할을 담당한다.

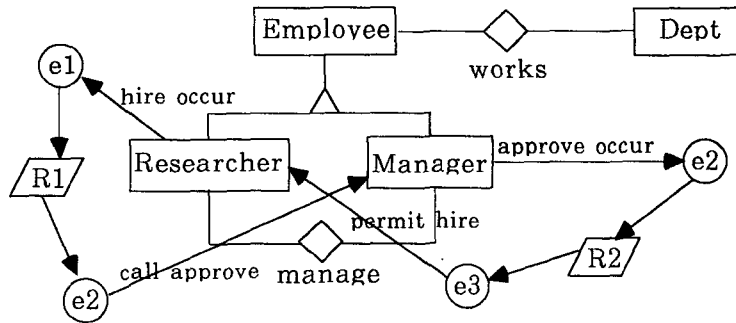
동적 규칙을 이용한 무결성의 예를 설명하기 위해서 그림 4 (a)의 연구소 데이터베이스 스키마를 정의한다. 이 스키마는 연구실의 구성원은 실장과 연구원으로 구성되고, 구성원들은 해당부서에 소속되어 있으며, 실장은 연구원들을 관리한다는 의미를 표현한다.

4.1 임무 분리를 유지하는 동적 규칙의 예

이 절에서는 임무 분리의 일반적인 경우에 해당하는 예를 들고, 임무 분리를 동적 규칙의 그래픽 표기법과 서술 구조로 정의한다. 먼저, 그림 4 (a)의 스키마에서 다음과 같은 임무 분리 무결성의 예를 고려해 보자.

[예제 1] 연구소에서 새로운 연구원을 채용한다고 할 때, 3명 이상의 연구실장들의 동의가 있어야 하며, 또한 세명의 실장중에는 신입 연구원이 근무할 연구실장이 동의해야 한다.

위의 예제를 동적 규칙의 다이어그램과 서술구조로 나타낸 것이 그림 4의 (a)에 제시되어 있다. 이 동적 규칙에서, researcher가 채용된다는 것은 researcher 클래스의 hire 메소드가 실행되어야 하고, 이 메소드가 실행될 때 예제에 서술된 무결성이 만족되는가를 검사한다. 따라서 세명의 승인이 있어야 하고, 반드시 한명은 소속실장이어야 한다는 임무 분리의 개념은 그림 4의 (b)에서와 같이 규칙의 조건절에 선언적으로 정의된다.



(a) 임무 분리 무결성의 그래픽 표현

```

CLASS Manager
INHERIT Employee
ATTRIBUTE
    emp_no : int; name: string;
    dept : DEPT;
METHOD create(); delete(); approve();
    
```

```

CLASS Researcher
INHERIT Employee
ATTRIBUTE
    SSN : int; name : string;
    advisor : MANAGER;
    hire_count : int;
METHOD create(); delete();
    
```

```

CLASS DEPT
INHERIT null
ATTRIBUTE
    dept_code : int;
    chief_id : MANAGER;
    emp_id : EMPLOYEE;
METHOD create(); delete();
    
```

ACTIVE RULE R1		ACTIVE RULE R2	
EVENT	BEFORE e1: Employee.hire occur;	EVENT	AFTER e2: manager.approve occur;
CONDITION	major == 'computer';	CONDITION	count == 3 \wedge researcher.manager == researcher.dept.manager
ACTION	raise e2: Manager.approve;	ACTION	permit e3: Researcher.hire;
COUPLING	immediate;	COUPLING	immediate;

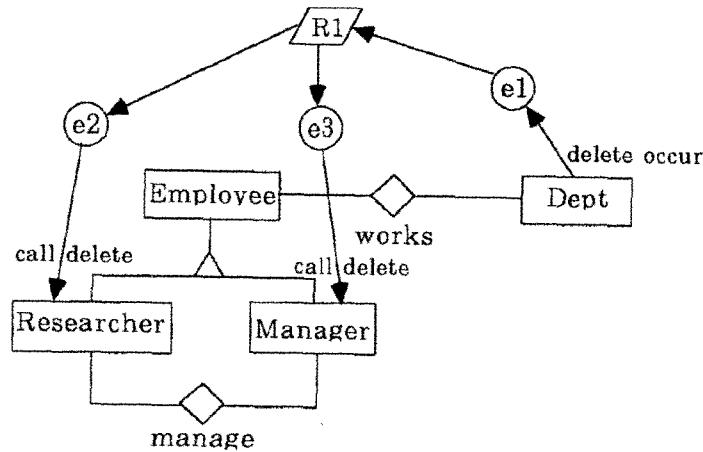
(b) 임무 분리 규칙의 서술 구조

<그림 4> 임무 분리를 유지하는 동적 규칙

그림 4에서는 무결성을 정의하기 위해서 두개의 규칙들이 정의되어 있다. 먼저, hire 메소드가 실행되면, 정의된 규칙 R1이 발화되고, R1은 무결성의 조건을 검사하기 위해서 manager에 정의된 approve 메소드를 실행시켜서 규칙 R2를 발화시킨다. 따라서 R2에 승인 조건이 만족되면 정의된 임무 분리 무결성이 유지되기 때문에 새로운 researcher의 hire 메소드 실행이 허용된다.

4.2 참조 무결성 예

참조 무결성은 관계형 데이터베이스에서 하나의 튜플이 참조하는 다른 튜플들이 존재해야 하는 조건을 의미한다. 따라서 서로 참조되는 튜플들의 내용이 서로 일관되게 유지될 수 있다. 이러한 참조 무결성은 객체지향 데이터베이스에서 의미적 무결성으로 고려할 수 있는데, 아래의 예에서 Dept 클래스에서 delete 메소드가 실행되므로써, 관련된 researcher와 manager 클래스에서 연구원을 삭제하는 연산이 수행되어야 무결성이 일관되게 유지된다.



(a) 갱신 연산시 무결성을 유지하는 동적 규칙 다이어그램

ACTIVE RULE R1	
EVENT	AFTER e1: dept.delete occur;
CONDITION	dept.emp_id == employee.emp_no;
ACTION	raise e2: manager.delete e3: researcher.delete;
COUPLING	immediate;

(b) 참조 무결성에 해당하는 동적 규칙 술어

<그림 5> 참조 무결성의 동적 규칙

5. 결론

데이터베이스 시스템에서 무결성은 데이터의 비밀관련 상태를 방지할 뿐만 아니라 데이터의 불법적인 변경을 통제하고, 데이터의 비밀성을 유지시키는 광범위한 개념이다. 따라서 데이터베이스에서 무결성은 중요한 요소이지만, 기존의 무결성 모델을 데이터베이스에 적용한 연구는 많지 않다. 대표적인 무결성 모델인 Clark/Wilson 모델은 무결성과 임무 분리를 효율적으로 유지시키는 모델로 평가되었고, 이 모델을 활용한 연구들이 많이 수행되었으나 객체지향 데이터베이스 시스템에서 이 모델을 적용하려는 연구는 이제 시작되고 있다.

따라서 본 논문에서는 Clark/Wilson 무결성 모델을 객체지향 데이터베이스에 적용시키는 기법을 제시하였다. 제시된 방법은 기존의 Herndon이 제시한 방법과는 달리 무결성을 유지시키기 위한 메소드를 정의하지 않고, 동적 데이터베이스 시스템의 동적 규칙 개념을 이용함으로써 여러 가지 무결성들을 쉽게 정의하고, 임무 분리를 쉽게 유지시킬 수 있는 잇점을 갖는다.

결론적으로, 동적 객체지향 데이터 모델의 그래픽 표기법과 서술 구조를 이용하여 쉽게 무결성의 의미를 파악할 수 있을 뿐만 아니라 정의할 수 있는 무결성 유지 기법을 제시하였다. 동적 규칙으로 무결성을 정의하는 기법은 기존의 응용 데이터베이스를 설계할 때 무결성을 유지시키는 메소드를 다른 메소드들과 동시에 정의하는 방법과 비교하면, 데이터베이스의 설계 작업을 간소화시킬 뿐만 아니라 여러가지의 의미적 무결성들을 변경할 때 유지보수하기가 훨씬 편리하다는 장점을 갖는다.

앞으로는 본 논문에서 미진했던 동적 규칙의 술어를 정형적으로 정의해야 하며, 그리고 이 논문에서는 고려하지 않았던 데이터베이스의 비밀성 요소를 포함할 수 있도록 제시된 기법을 확장하는 연구를 계속해서 수행할 예정이다.

< 참 고 문 헌 >

- [1] A. K. Tanaka, On Conceptual Design of Active Databases, Ph.D Thesis, Georgia Institute of Technology, 1992.
- [2] D.D. Clark, D.R. Wilson, "A Comparison of Commercial and Military Security Policies," Proc. of 1987 IEEE Symposium on Security and Privacy, 1987
- [3] D.J. Thomsen, "Role-Based Application Design and Enforcement," Database Security IV Elsevier Science, 1991, pp151-168
- [4] E. Anwar, L. Maugis, S. Chakravarty, "A New Perspective on Rule Support for Object-Oriented Databases," Proc.1993 ACM SIGMOD Int. Conf. on Management of Data,1993, pp99-108.
- [5] Millen, J.K., T.F. Lunt, "Security for Object-Oriented Database Systems," Proc. of 1992 IEEE Symposium on Security and Privacy, 1992
- [6] M. D. Abrams, E. G. Amoroso, L. J. LaPadula, T.F. Lunt, J.G. Williams, "Report of an integrity research study group," Computer & Security, Vol.12, No.7, 1993, pp679-689
- [7] N. H. Gehani, H. V. Jagadish, "Active Database Facilities in Ode," Data Engineering, Vol.15, No.1-4 1992, pp 19-22.
- [8] S.B. Navathe, A.K.Tanaka, S. Chakravathy, "Active Database Modeling and Design

- Tools: issues Approach, and Architecture," Data Engineering, Vol.15, No.1-4, 1992, pp 6-9.
- [9] U. Dayal, A.P. Buchmann, D.R. McCarthy, "Rules Are Objects Too : A Knowledge Model For An Active, Object-Oriented Database System," Advanced Symposium in Object-Oriented Database Systems, 1988, pp 129-143.
- [10] U. Dayal, B. Blaustein, U. Chakravarthy, "The HiPAC Project : Combining Active Database and Timing Constraints," ACM SIGMOD RECORD, Vol.17, No.1, 1988, pp 51-70.
- [11] William R. Herndon, "An Interpretation of Clark-Wilson for Object-Oriented DBMSs," Database Security, VII, Elsevier Science, 1994, pp65-85