

개념적 모델링을 위한 관계 표현의 확장

강병영

동의대학교 상경대학 경영정보학과

요약

데이터베이스 설계에서 개념적 모델링 단계에서는 논리적 설계가 효율적으로 수행되도록 하기 위하여 현실 세계를 보다 정확하게 표현할 수 있도록 다양한 의미 표현 도구를 제공해야 한다. 본 연구에서는 데이터베이스 개념 설계에 가장 널리 사용되는 실체 관계 모델(ERD)을 기반으로 데이터의 표현 능력을 향상시키고자 한다. 특히 데이터의 표현을 확장하려는 연구들을 정리하고 통합적인 데이터 모델링 방법의 개발 필요성을 제시한다.

1. 서론

데이터베이스 개념 설계에 가장 널리 이용되는 실체 관계 모델은 일반화, 분류화 등의 추상화 개념을 제대로 표현해 주지 못하는 단점이 있다. 따라서 의미 데이터 모델의 추상화 개념을 추가하여 현실 세계의 정적 성질을 정확히 표현할 수 있도록 할 필요가 있다. 또한 데이터 모델링에 정적인 성질 외에 오퍼레이션과 같은 동적인 성질이 추가될 필요가 있다. 따라서 데이터 추상화 개념에 중점을 두는 데이터 모델과 오퍼레이션-동적 특성을 모델화한 추상화의 통합으로 데이터 모델링 능력을 향상시키는 연구가 요구되어 왔다.

2. 의미 데이터 모델(Semantic Data Model)

의미 데이터 모델의 두 가지 주요 목적은 첫

째는 데이터베이스의 물리적 구조에 얽매이지 않고 개념적 데이터 모델을 개발할 수 있도록 하는 데이터 독립성(data independence)이며 둘째는 데이터 모델링 과정에 부가적 의미를 부여하여 함수적 종속(functional dependencies), 관계와 연관화(association), 일반화와 집단화(generalization & aggregation), 분류화(classification), 상위-하위 클래스 개념, 속성 상속(attribute inheritance) 등을 명확히 표현하도록 하는 것이다.

개념적 설계를 위한 의미 데이터 모델에는 다음 특성들이 포함되어야 한다.¹⁾

- 표현력 : 데이터, 관계, 제약조건의 여러 유형을 충분히 구분할 것.
- 단순성 : 사용자가 이해하고 사용하기 쉽도록 단순할 것(그래픽 표현).
- 최소성 : 모델의 기본 개념이 적을 것.
- 형식성 : 스키마의 타당성을 위한 기준을 제시할 것.
- 고유한 해석 : 모델링 구조에 완전하고 명확한 의미를 부여할 것.

전통적 모델과 달리, 대부분 의미 데이터 모델의 구조는 하향식(top-down), 스키마의 모듈

1) C. Batini, S. Ceri, and S.B.Navathe, Conceptual Database Design: An Entity-Relationship Approach, Benjamin Cummings Redwood City, Calif., 1992.

뷰(modular view)를 유지하며 스키마 설계와 데이터베이스 사용을 단순화한다.

의미 데이터 모델의 기본이 되는 특성들은 다음과 같으며 이는 비교의 기준이 되어진다.

- a. 비구조적 객체의 표현
- b. 관계 표현
- c. 표준화된 추상화(예: 분류화, 일반화, 집단화, 연관화 등)
- d. 관계의 계층 또는 네트워크 (예 : IS-A 다이어그램)
- e. 유도/상속 - 다중상속, 명칭 충돌 문제 처리 포함
- f. 삽입/삭제/수정 제약 조건
- g. 관계 의미의 표현 정도 - 사상수(cardinality), 널 값(null values), 역 관계, 유도, 속성의 표현
- h. 동적 모델링 - 의미 데이터 모델은 정적인 면을 강조하지만 삽입/삭제 제약, 예러 복구, 트랜잭션 정의 등의 동적 모델링 부분이 있다.

이들 특성들을 가지고 대표적인 의미 데이터 모델들을 살펴본다.

① 함수적 데이터 모델(Functional Data Model, FDM)

실체를 형성하는 속성들의 집단화를 정의하기 위해 함수가 이용되는 함수적 데이터 모델은 1976년 Kerschberg와 Pacneco에 의해 제시되어 1981년 Shipman의 데이터 언어 DAPLEX에 도입되었다. FDM에서 속성은 단일값 또는 다중 값을 가지며 ISA 관계를 표현한다. 관계는 함수적으로 표현된다.

② 의미 데이터베이스 모델(SDM)

설계자의 개념을 직접적으로 표현하기 위해 "클래스(class)"라는 추상화를 사용하는 SDM은 1981년 Hammer와 McLeod에 의해 제시되었다. SDM은 계층구조나 클래스 사이의 상호 관계 표현을 사용하지 않고, 클래스 추상화로 상속성, 제약조건, 유도 옵션의 풍부한 셀을 제공한다.

다.

SDM에서는 집단화, 일반화보다 분류화, 연관화가 강조되며 일반화, 연관화 추상화 개념을 다루기 위해 하위유형(subtype)과 그룹핑 연결이 사용된다. SDM은 데이터의 사용자 관점을 표현하는데 효과적이며 다른 의미 모델을 위한 원형(prototype)시스템으로서 공헌한다. ER 모델과 달리 SDM은 속성에 중점을 두며 FDM이나 ER 보다 풍부한 표현력을 가진다.

③ RM/T

Codd의 관계형 모델의 기본적 특성을 유지하면서 의미적 표현력을 향상시킨 RM/T 모델은 1979년 Codd에 의해 제시되었다. RM/T 모델은 실체와 관계유형을 규정한다. 실체유형은 유형에 하나만 존재하는 E-relations에 의해 정의되며 유형의 특성(속성)은 P-relation으로 정의된다.

M:M 관계에는 연관(associative) 실체유형, M:1 관계에는 지정(designative) 실체유형을 사용하여 표현한다. ISA 관계는 하위유형의 어떤 객체는 상위유형의 멤버이어야 하는 무조건적 일반화와 유형 집합의 부분집합을 형성하는데 사용되는 선택적(조건적) 일반화로 구분된다.

④ 의미 연관 모델(A Semantic Association Model, SAM')

비전통적 객체유형과 관계를 지원하는 특징의 SAM' 모델의 일반적 조직은 원자와 비원자 개념(또는 객체)의 네트워크이다. 1986년 Su는 CIM 환경의 요구 분석의 결과로 SAM'에 다음의 유형과 오퍼레이션을 추가했다.

- a) 셀(sets)
- b) 벡터와 행렬
- c) 타임과 타임 시리즈
- d) 텍스트

e) G-relations(일반화된 관계)

SAM'의 G-relation은 속성들이 관계를 포함하여 어떤 유효한 SAM' 유형이 될 수 있는 확장된 관계이다.

Su는 또한 7개의 관계(SAM'에서는 연관화로 표현)로 물리적, 추상적 객체를 표현했다. 전형적 4개의 관계(membership, aggregation, interaction, generalization)에다가 통계적 응용에 적합하며 셸의 집단 특성에 초점을 맞춘 Composition("is part of"), Cross Product(유형의 cartesian product), Summarization(통계적 집산화와 비집단화를 지원) 관계를 도입했다. SAM'은 비전통적 데이터를 포함한 통계적 데이터베이스와 CAD/CAM 환경과 같은 응용에 의미를 표현하는데 유용함을 보여준다.

⑤ 동적 의미 모델(Dynamic Semantic Model)

정적 의미 모델은 관계형 데이터베이스 개념과 의미네트(semantic nets)로 부터 온 네 가지 중요한 관련성, 즉 분류화, 집산화, 일반화, 연관화를 통합한 형태로 스키마는 객체의 집산화, 일반화, 연관화의 계층 구조로 표현되어진다.

동적 의미 모델은 정적 모델링 개념에 동적 특성을 부여함으로써 ADT(abstract data types), Class, Strong typing, Polymorphic type과 같은 진보된 데이터 유형 개념을 사용할 수 있게 한다. 동적 의미 모델 중 대표적인 모델은 다음과 같다.

- a) 확장된 의미 계층 모델(SHM'; Brodie 1984, Brodie and Ridjanovic 1984)
- b) TAXIS(Borgida, Mylopoulous and Wong 1984)
- c) Event Model(King and McLeod 1984, 1986)

SHM'는 집산화, 그룹핑, 유형유형으로 정적 데이터 표현을 트랜잭션의 통제 흐름을 나타내는 행위 스키마로 동적 데이터 모델링이 가능하다. TAXIS는 분류화, 일반화 계층을 강조하고 세분화를 사용하여 데이터베이스의 동적 부분을 프로그래밍 언어와 데이터베이스 이론을 결합한 데이터베이스 시스템 설계를 위한 언어이다. Event Model은 먼저 단계적 정제(stepwise refinement)의 프로세스가 함수를 정의하고 정보의 흐름을 나타내는 경로를 연결하여 데이터베이스의 동적 측면을 고려한 모델이다.

⑥ ER 모델

ER 모델은 실체, 속성, 관계로 이루어진다. 먼저 실체란 시스템 내에 존재하는 객체들의 집합을 의미한다. 실체는 강(strong) 실체와 약(weak) 실체로 구분한다. 강 실체란 개별 실체를 유일하게 결정해 주는 식별자를 그 실체의 속성으로 포함하는 경우이고, 약 실체란 다른 실체의 속성을 이용하여 개별 실체를 구별하는 경우이다.

관계는 실체들이 어떤 연관을 맺고 있는지를 속성을 사용하여 표현한다. 관계는 실체와의 대응도를 1:1, 1:N, N:M의 관계로 구분하며, 실체의 관계 참여가 절대적인가 선택적인가로 구분한다.

ER 모델은 ISA 관계로 일반화를 나타내며 세분화, 집산화가 표현될 수 있도록 확장되었으며(확장 실체-관계 모델: EER 모델), 관계형 모델과 비교하여 EER 모델이 표현력이 뛰어나며 초보자에게 성과가 있음이 나타났다.²⁾ ER 모델은 간단한 데이터베이스 구조는 명료하게 묘

2) D.Batra, J.A.Hoffer, and R.P.Bostrom, "Comparing Representations with Relational and EER Models," Com. of the ACM, Vol. 33, No. 2, Feb. 1990, pp. 26-38.

사할 수 있으나, 엔지니어링, 영상, 음성 그래픽 등의 복잡한 자료 구조와 의미를 표현하지는 못한다.

3. ER 모델의 확장3)

데이터베이스의 개념적 설계에 가장 폭넓게 사용되는 ER 모델은 ECR 모델, EER 모델, EEER 모델 등으로 확장되었다.

① ECR 모델

ECR(Entity-Category-Relationship) 모델은 ER 모델을 여러 가지 측면에서 확장한 것이다. ECR 모델은 실체들을 두 가지 방법으로 분류하는데, 첫째는 특성들의 유사성에 따라 둘째는 실체가 관계에서 하는 역할에 따라 분류한다. 그리고 ECR 모델은 일반화와 같은 고수준의 관계와 복잡성을 다룰 수 있다.

실체들은 수행하는 역할에 의하여 범주로 분류된다. 범주는 서로 다른 둘 이상의 상위유형의 합집합과 하나의 상위/하위 관계를 가지는 하위유형이다.

ECR 모델은 다음과 같은 장점을 가진다. ECR 스키마는 네트워크나 관계형으로 변환이 용이하며, 복잡한 관계를 표현하는데 필요한 의미론적 표현 방법을 제공하고, Weeldreyers의 ECR calculus와 1차 술어 논리로 완전하게 정의된다. 그러나 ECR 모델은 기본적으로 문자 데이터를 다루는 것으로 엔지니어링 데이터를 다루기 위해서는 객체지향의 개념 등으로 확장해야 한다.

범주를 관계형으로 전환하는 과정에서 주의해야 할 점은 여러 상위유형의 기본키가 다를

수 있다는 점이다.

② EER 모델

EER(Extended Entity-Relationship) 모델은 앞에서 소개된 것과 같이 ER 모델을 확장하여, 현실 세계의 의미를 보다 정확하게 나타나게 한다. EER 모델이 지원하는 추상화 개념은 다음과 같다.

첫째, 일반화는 실체 유형들의 상위유형을 결정하는 작업으로, 둘 이상의 실체 유형으로부터 공통의 속성을 포착하여 새로운 상위유형을 만드는 것이다.

EER 모델은 실체들 사이의 계층적 관계를 잘 표현할 수는 있으나, 관계에서 수행하는 역할에 근거한 실체들의 관계를 표현할 수 없다.

③ EEER 모델

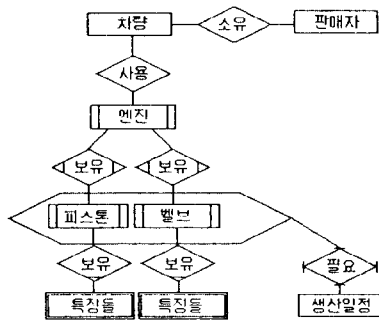
EEER(Extended Entity-Category-Relationship) 모델은 실체들로 구성되는 복합객체와 다양한 데이터를 표현하기 위하여, 4개의 실체 클래스와 4개의 관계 클래스를 활용하고 있다.

실체 클래스의 유형인 정규(regular) 실체는 비계층적 객체와 개념을 표현하며, 범주(category) 실체는 관계 속에서 수행하는 역할에 따라 분류한다. 범주는 단 하나의 실체만으로도 구성될 수 있다. 복합(kernel) 실체는 한 단위로 참조될 수 있는 복합객체를 확인하기 위한 것이다. 약(owned) 실체는 엔지니어링 객체의 계층적 구조를 표현하기 위한 것으로 상위 실체의 존재에 달려 있다.

관계 클래스의 유형인 약(weak) 관계는 복합 실체와 약 실체 사이 또는 두 개의 약 실체 사이의 1:N의 이진 관계를 나타낸다. 이 관계에서 하위 실체의 존재는 상위 실체에 달려 있다. 참조(reference) 관계는 M:N 관계로서 복합

3) ER 모델, ECR 모델, EER 모델, EEER 모델은 발표된 순서에 따라 요약 정리한 것

실체를 하나 이상의 상위 복합 실체와 연결하는데 사용된다. 제한(limited) 관계는 범주나 정규 실체가 계층구조에서 최상위인 경우의 1:n 관계를 표현하는데 사용된다. 일반(complete) 관계는 일반적인 복합, 정규, 범주 실체의 M:N 관계를 표현한다.



< 그림 1 > EECR 모델 스키마의 예4)

<그림 1>은 자동차 회사에서 제조되는 차량을 나타내었다. 한 차량은 엔진(복합 실체)을 사용하며, 엔진은 피스톤, 밸브 등의 복합 요소들로 구성된다. 이들 각 요소는 흠, 구멍과 같은 다양한 특질(소유 실체)을 가진다. 생산 일정이라는 관계를 위하여 피스톤과 밸브는 부품이라는 범주 실체로 정의될 수 있다. EECR 모델은 ER 모델을 확장하여 분산 환경하의 CAD/CAM 데이터를 개념적 모델링을 할 수 있음을 보여준다. EECR 모델을 기존의 모델로 매핑하는 것이 요구된다.

4. 의미 데이터 모델의 평가와 전망

대부분의 의미 데이터 모델들은 추상화 방법으로 일반화와 집산화 개념을 활용하고 있다.

4) H.K. Jain and M.I.Bu-Hulaiga, "E-R Approach to Distributed Heterogeneous Database Systems for Integrated Manufacturing," Journal of Database Administration, Summer 1992. pp. 21-29.

그렇지만 모델링 프로세스의 핵심적 부분으로서의 동적 모델링은 극소수 모델들에서만 나타난다. SAM, SDM은 삽입/삭제의 제약조건은 관계 정의의 일부분으로 규정하나, 함수적 데이터 모델이나 ER 모델에서는 사용자가 규정하도록 한다. 유도(derivation) 방법에도 차이가 있어 대부분의 모델들은 유도의 유일한 형태로 일반화 관계를 통한 상속을 활용하고 있지만, SDM과 같은 모델은 다양한 방법을 제시한다.

의미 데이터 모델링으로 분류될 수 있는 개념적 모델링 접근법은 다양하지만, 이들 모두는 데이터 객체 사이의 관계와 이들 관계에 추가적인 의미를 부여하여 데이터 모델링을 용이하게 하고자 하는 목적은 같다.

의미 데이터 모델을 사용함으로써 얻는 이점으로는 첫째, 표현의 경제성으로 개발자가 같은 정보를 정확하게 추출할 뿐 아니라 쉽게 얻을 수 있다. 둘째, 무결성 유지로서 사용자가 물리적 수준을 고려하지 않게 되며 무결성 제약 조건의 정리에 대한 메카니즘을 제공한다. 셋째, 모델링 유연성과 관련하여 추상화 개념의 사용은 데이터를 여러 수준에서 조명할 수 있게 한다. 넷째, 모델링 효율성으로 대부분 의미 모델들이 기본적인 오퍼레이션과 제약 조건을 고정시켜 사용함으로써 설계자가 시간 절약을 할 수 있다.

그리고 의미 데이터 모델에서 데이터 조작 언어는 전통적 레코드-지향 모델의 조작 언어와는 달리 추상 유형을 질의할 수 있어야 하며, 속성들을 참조하거나 조작하기 위한 설비를 제공해야 하고, 기존(하위)유형, 함수로 구성된 하위유형 그리고 함수의 형태에서 유도된 데이터를 사용자가 다룰 수 있도록 해야 한다.

대부분의 의미 데이터 모델의 연구에서 나타

난 바와 같이 모델의 객체, 오퍼레이션 관계의 다양한 의미를 개발하는 것이 중요하다. 그리고 비전통적 데이터를 지원하는 SAM 모델과 같이 특수한 유형의 설계를 위하여 텍스트, 음성, 이미지와 같은 멀티미디어 정보를 처리하기 위한 접근법이 요구된다.

또한 대부분 의미 데이터 모델이 정적 모델링을 지원하며 동적 특성을 다루고 있으나 동적, 일시적인 모델링에 대한 연구로 사건의 표현뿐만 아니라 행위적 측면을 보다 효과적으로 다룰 필요가 있다. 그리고 시간 표현에 대한 다양한 방법들이 제시되어 왔지만 일시적 데이터에 대한 제한적인 결과로 지속적인 연구가 필요하다.

의미 데이터 모델이 데이터베이스의 설계에 대한 실무적 면에서 한계가 있고 상업적 데이터 처리에 중점을 두어 관계형 모델 사용을 뛰어 넘지 못하고 있어 데이터의 행위적 측면과 프로그래밍 언어와 인공지능 영역으로부터의 아이디어를 결합하여 객체지향 데이터 모델이 출현하게 되었다고 할 수 있다.

5. 관계 표현의 확장

1) EER 모델과 객체지향 모델의 관계성 비교

의미 데이터 모델에서의 관계 추상화와 객체지향 모델에서의 관계성을 비교하여 통합 OER 모델에서 이러한 개념을 사용한다.

먼저, 연관화 개념으로 EER 모델에서의 연관화는 실체들간의 연결(link)을 의미한다. 즉, 한 실체와 다른 실체간의 연결을 나타낸 것으로 연결상의 정도는 1:1 연관화, 1:0,1 연관화, 1:M 연관화, M:N 연관화의 4가지로 구분된다. 1:0 연관화는 연결되는 실체가 실제로 존재할 수도

혹은 안할 수도 있다는 것을 나타내는데 사용한다. EER 모델에서 연관화란 객체들 간에 어떤 의미나 목적을 가지고 연결시키는 것을 말하는데, 객체지향 모델에서의 연관화도 객체들 간의 어떤 의미나 목적을 가지고 연결시키는 것을 의미한다.

집단화와 일반화 개념을 비교하면 EER 모델에서의 집단화는 실체를 구성할 때 여러 개의 실체나 항목들이 합성되어 한 실체로 되는 것을 말한다. 객체지향 모델에서도 단일 객체로 집단화 관계가 동일하게 쓰인다. EER 모델에서의 일반화란 별개의 실체들이 소유한 값들은 다르지만, 어떤 공통된 속성을 가진 것들로 구성된 것이거나 일반 실체에 부속화된 것을 말한다. 그러나 객체지향 모델에서의 일반화란 개념적 모델링과 구현 모두를 수행하기 위한 유용한 개념이다. 일반화란 클래스들의 개별적인 특성을 유지시키면서 유사한 특성을 공유할 수 있도록 해주는 강력한 추상화 기법이다. EER 모델과 객체지향 모델에서의 일반화는 유사하나, EER 모델에서는 계속적으로 상위 클래스의 식별자를 하위 클래스가 가지고 있는 반면 객체지향 모델에서는 그러하지 않다.

상속성을 처리하는데는 조금의 차이가 있다. EER 모델의 일반화는 개념적 모델링에 치중하지만 실제 테이블이나 관계형 언어로 구현할 때 키(key)로 상속이 구현된다. EER 모델에서는 단일 상속이나 다중 상속이라는 말이 없다. 그리고 EER 모델은 상속을 표현할 때 객체지향 모델의 상속처럼 상속되는 모든 특성을 보여 주지는 못한다. 즉, 객체지향 모델은 상속되는 특성을 직접 기술해 주지만, EER 모델에서는 키로 상속하기에 직접 상속하는 특성을 기술해 주지는 않는다.

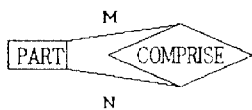
EER 모델과 객체지향 모델에서의 분류화

개념은 비슷하나 EER 모델에서는 실체들을 구조의 유사한 특성에 따라 분류하지만 객체지향 모델에서는 행위의 유사성을 고려하여 분류하는 것이 다르다.

2) 관계의 확장

가) 회선(involved) 관계

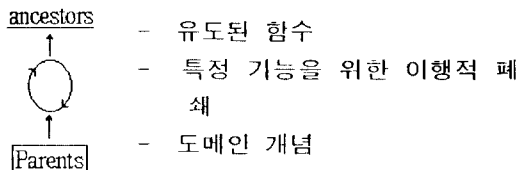
제품 객체에서 나타나는 연관관계를 표현하기 위해 회선 관계와 이행적 관계가 필수적이다. 회선 관계는 1:1 회선 관계, 1: N 회선 관계, M:N 회선 관계로 구분할 수 있으며 M:N 회선 관계는 <그림 2>와 같다. 실체 타입 PART의 인스턴스는 다른 부품들로 구성될 수 있다. 또한 어떤 부품은 많은 다른 부품들의 요소가 될 수 있으며, 제품-부품의 관계에도 적용된다.



<그림 2> M:N 회선 관계

나) 이행적 폐쇄(transitive closure) 관계

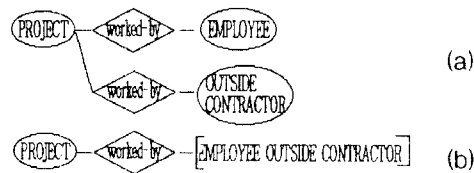
이행적 폐쇄 관계는 객체가 결정될 수 있을 때까지 함수를 재적용하는 반복적 오퍼레이션이다. 이행적 폐쇄의 가장 널리 사용되는 응용에는 자재명세서에서의 부품전개이다. 여기서 이행적 폐쇄는 부품(PART) 객체를 조립부품(subassembly), 조립부품의 조립부품을 결정하는데 적용한다.



<그림 3> 이행적 폐쇄

다) 중첩(overloading) 관계

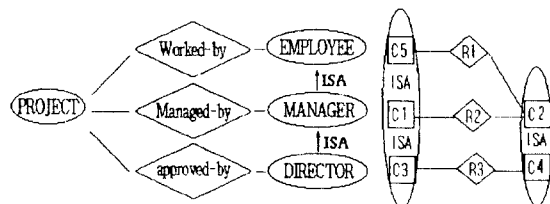
CIM 객체의 중복 관계로 인한 객체의 이중 표현은, 같은 명칭을 가진 관계의 중첩과 다중 관계를 허용함으로써 해결할 수 있다. 즉, <그림 4>에서 EMPLOYEE 클래스와 OUTSIDE CONTRACTOR 클래스는 PROJECT 클래스에 참여함을 나타낸다(a). 관계의 중첩 허용은 EMPLOYEE와 OUTSIDE CONTRACTOR로 구성되는 새로운 클래스를 만들 필요가 없어진다(b).



<그림 4> 중첩 관계

라) 다중 관계(multiple relationship)와 관계의 포함(subsumption)

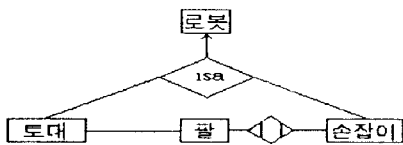
한 객체는 클래스의 다른 객체와 여러 관계로 관련될 수 있다. 클래스 PROJECT은 EMPLOYEE, MANAGER, DIRECTOR와의 다중 관계가 존재함을 알 수 있다. 또한 관계와 관계 사이에 일반화/세분화가 있을 경우 관계의 포함이 발생한다. <그림 5>에서 관계 R2는 관계 R1을 포함하고, 관계 R3은 관계 R2를 포함한다. 클래스 C5가 클래스 C1의 일반화이며, 관계 R1이 관계 R2의 일반화일 때 관계 R2는 관계 R1을 포함하는 것이다.



<그림 5> 다중 관계와 관계의 포함

마) 동시 관계

로봇 객체는 토대, 팔, 손잡이 세부분으로 이루어진다. 토대, 팔, 손잡이 객체의 데이터 값은 위치를 나타내는 벡터 값으로 구성된다. 여기서 팔의 위치가 변하면 자동적으로 손잡이 위치가 변하게 된다. 토대는 고정되어 있어 팔과 손잡이의 위치에 변화를 주지 않는다. 그리고 손잡이의 움직임으로 값이 변하더라도 팔의 값이 변하지는 않는다. 여기서 팔과 손잡이 객체의 관계 즉, 한 객체의 데이터 값이 변하면 자동적으로 다른 객체의 값이 변하는 관계를 동시관계로 규정하고 <그림 6>에 다이어그램으로 표현하고 있다.

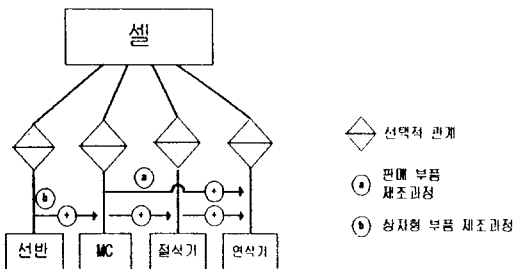


<그림 6> 동시관계의 예

바) 선택적 관계

CIM의 제조 공정 과정을 모델링하기 위해서는 실행 절차가 표현되어야 한다. <그림 7>의 데이터 모델에서 기계 가공은 셀(cell)에서 이루어지고 셀은 선반, 머시닝센터(MC), 절삭기, 연삭기로 구성되고 있음을 보여준다.

공작기계 제품의 종류에 따라 사용되는 부품이 다르고 그 부품을 셀에서 제조된다.



<그림 7> 제조 과정의 선택적 관계

3) 관계 객체의 동적 관리

CIM 시스템과 같은 다양한 관계성을 표현하고 조작하는 새로운 데이터베이스 응용분야에서는 동적으로 정의되는 다양한 관계성의 지원이 요구된다. 그리고 관계성에 관련된 다양한 무결성이나 일관성 제약조건들도 유지되어야 한다. 관계성을 동적으로 관리하기 위해서는 관계성 클래스를 동적으로 생성, 삭제할 수 있고 관계성에 참여하는 클래스의 정의를 동적으로 갱신할 수 있는 기능이 필요하게 된다. 그리고 이러한 연산들은 관계성에 무관한 참가 객체들의 정보에는 영향을 미치지 않아야 한다.

관계성 클래스는 실제 참가 클래스들을 물리적으로 결합하는 것이 아니다. 관계성 객체의 정의와 같이 관계성에 참가하는 클래스들의 존재는 관계성 클래스의 존재와는 무관하게 된다. 따라서 관계성 클래스는 다른 클래스의 객체에는 영향을 미치지 않고 동적으로 생성되거나 삭제될 수 있다.

관계성 클래스가 새로 생성되면 이 클래스를 도메인으로 갖는 참조 속성이 관계성에 참가하는 각 클래스의 정의에 단순히 참가될 수 있다. 반대로 관계성 클래스가 삭제될 경우에는 그 클래스를 도메인으로 가지는 참조 속성의 삭제를 야기시킨다. 이 외의 다른 일반 클래스의 생성과 삭제는 기존에 연구된 스키마 변경 규칙을 따르면 된다.

관계성은 관계성 객체로 표현되며, 관계성 객체는 관계성에 관련된 정보와 제약조건들을 포함하는 논리적 실체이다. 관계성 R이 클래스 E₁, E₂, ..., E_N 사이의 연관성을 표현할 때, 관계성 클래스 R은 다음과 같이 정의된다.

$$R=(E_1 \times E_2 \times \dots \times E_N \mid C, P^*)$$

C: 관계성 R에 관한 제약조건

E: 관계성 R의 참가 클래스

P*: 관계성 특성의 집합

관계성 클래스는 클래스들 간의 연관성을 나타내므로, 관계성은 $n(>=2)$ 개의 클래스들간의 수학적 관계이다. 따라서 E_1, E_2, \dots, E_N 이 클래스일 때 관계성 클래스 R은 관계성에 참가하는 클래스들의 카티션 프리덕트인 $E_1 \times E_2 \dots \times E_N$ 과 관계성 자체의 특성인 P^* 로 표현된다. 여기서 관계성 클래스 R의 특성은 관계성 자체의 특성뿐만 아니라 참여하는 클래스들의 모든 특성도 함께 포함한다. 그리고 관계성은 카티션 프리덕트의 결과를 제한하는 제약조건 C에 의해 제한되어진다. 따라서 관계성에 관련된 모든 정보가 관계성 클래스에 표현되고 이 관계성에 관한 연산은 모든 참가 클래스들에 동등하게 적용될 수 있다.

관계성 객체는 관계성의 한 인스턴스로, 관계성에 참가하는 각 객체는 다른 참가 객체와 이 관계성 객체를 통해서만 연관되어진다. 관계성 객체는 관련된 여러 객체가 하나의 논리적 실체로 취급되어진다는 측면에서 복합객체와 유사한 특성을 가진다. 그러나 복합객체는 배타적인 요소 객체들로 구성되며 객체간에 "part-of" 관계를 갖기 때문에 요소 객체는 종속객체가 되어 그 존재가 독립객체인 복합객체에 의존하게 되고 반드시 하나의 객체에만 종속되게 된다. 반면, 관계성 객체는 관계성에 참가하는 객체들로 구성되며 여러 객체들 사이의 수평적 연관성 관계를 표현한다. 또 관계성 객체는 각 참가 객체들의 존재에 의존한다. 그러므로 만약 관련된 참가 객체나 연관성 자체가 없어진다면 그 연관성을 나타내는 관계성 객체

도 삭제되게 된다.

4) 예외 관리

데이터베이스에서 스키마는 저장된 데이터가 규정된 카테고리화 제약조건이 확정되는 것으로 가정한다. 그러나 실세계는 이런 고정적(stereotype) 성질을 가지지 않고 새로운 정보들이 수시로 스키마가 확정되지 않도록 한다. 이러한 정보를 예외라 부른다. CIM 데이터는 이러한 예외 관리가 자주 요구된다.

예외는 인스턴스와 스키마 수준에서 발생할 수 있다. 스키마 수준에서 예외는 클래스가 상위클래스를 상속받는 특성을 위반할 때 발생하며 예외 클래스라 한다. 인스턴스 수준에서 예외는 인스턴스 객체가 클래스에 명세된 제약조건을 위반할 때 발생하며 예외 인스턴스 객체(또는 예외 객체)라 한다.

예외는 속성 예외, 관계 예외, 오퍼레이션 예외의 세 타입으로 구분할 수 있으며 스키마와 인스턴스 수준에서 발생할 수 있다.

첫째 속성 예외의 비관련 속성(Irrelevant attribute)의 경우, 인스턴스 수준에서의 예로는 특정 종업원이 자원 봉사자로서 '월급' 속성을 갖지 않는 경우이다. 스키마 수준의 예로는 서브클래스 MANAGER가 상위클래스 EMPLOYEE의 속성 'no-of hours'를 갖지 않는 경우이다. 그리고 속성 범위 위반, 스키마에 규정된 타입 위반, 새로운 속성 추가에서 발생한다.

둘째 관계 예외로는 비관련 관계(inherited but irrelevant relationship), 관련없는 클래스(inherited relationship with unrelated class), 관계 추가(additional relationship)의 경우가 있다.

셋째 오퍼레이션 예외로는 상속되거나 관련없는 오퍼레이션과 상속되며 오퍼레이션 추가의 경우가 있다.

예외의 경우 무결성 제약조건의 완화, 기존 클래스의 변경, 기존 클래스에 새로운 특성 추가, 새로운 서브클래스의 정의 등으로 스키마를 수정해야 한다.

6. 결론

본 연구에서는 의미 데이터 모델인 ER 모델에 추상화 개념과 객체지향 개념을 확장한 OOER(Object-Oriented Entity Relationship) 모델을 이용하였다.

비계층적 관계를 확장하여 회선 관계, 이행적 폐쇄 관계, 관계의 중첩, 다중 관계, 유도 관계, 동시 관계, 선택적 관계 등을 추가하였다. 또한 관계성 객체가 동적으로 생성, 삭제될 수 있도록 하여 관계와 복잡한 엔지니어링 데이터의 관계 처리가 가능하도록 하였다.

위의 결과로서 개념적 모델링 단계에서 데이터의 특성을 다이어그램 표현으로 나타냄으로써 논리적 모델의 실계에 정확성을 가져다준다.

7. 참고문헌

- Batra, D. and J.A.Hoffer, "An ER Based Methodology for Modeling User Views and Detecting Defined Relationships," *Journal of Database Management*, Vol.5, No.1, Winter, 1994, pp.3-16.
- Chen, P.P.S., "The Entity-Relation Model: Toward a Unified View of Data," *ACM Transaction on Database Systems*, Vol.1, No.1, 1976, pp.9-36.
- David, M.Dilts, "Integration of computer Integrated Manufacturing Data Bases Using Artificial Intelligence," *Proc. of the 2nd International Conference on Expert Systems and the Leading Edge in Production Planning and Control*, 1988, South Carolina, pp.277-284.
- Dillon, T. and P.L.Tan, *Object-Oriented Conceptual Modeling*, Prentice-Hall 1993.
- Gorman, K, and J. Choobineh, "The Object-Oriented Entity-Relationship Model (OOERM)," *Journal of MIS*, Vol.7, No.3, Winter, 1990-91, pp.41-65.
- Richter, "Towards a Complete Conceptual Model: Petri nets and Entity-Relationship Diagrams," *Information Systems*, Vol.18, No.5, 1993, pp.275-298.
- Hull, R., and R. King, "Semantic Database Modeling: Survey, Applications and Research Issues," *ACM Computing Surveys*, Vol.19, No.3, Sep. 1987, pp.201-260.
- Jain, H.K. and M.I.Bu-Hulaiga, "E-R Approach to Distributed Heterogeneous Database Systems for Integrated Manufacturing," *Journal of Database Administration*, Summer 1992, pp.21-29.
- Ling, T.W., P.K. Teo and L.L. Yan. "Generating Object-Oriented Views from an ER-Based Conceptual Schema," *Proceedings of The Third International Symposium on Databases for Advanced Applications*, Taejon, Korea, April 6-8, 1993, pp.148-155.
- Navathe, S.B. and M.K. Pillalamarri, "OOER: Toward Making the E-R Approach Object-Oriented," *Entity-Relationship Approach* 1989, pp.185-205.
- Peckman, J, and F. Maryanski, "Semantic Data Models," *ACM Computing Surveys*, Vol.20, No.3, Sep. 1988, pp.153-189.
- Shoval, P. and I. Frumermann, "OO and EER Conceptual Schema: A comparison of User Comprehension," *Journal of Database Management*, Vol.5, No.4, Fall 1994, pp.28-38.
- Su, S.Y.W., "Modeling integrated Manufacturing data with SAM," *Computer*, Vol.19, No.1, 1986, pp.34-49.