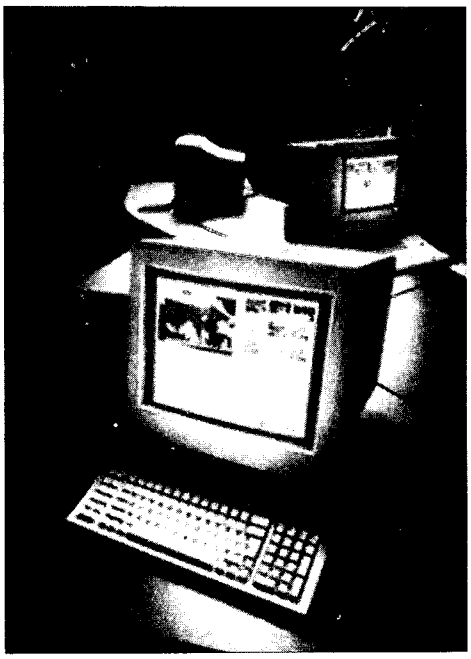
시스템 통합 환경을 고려한 논리적 데이터베이스 설계 방법론

정인기, 백두권
고려대학교 전산과학과

A Logical Design Methodology for Database in System Integration Environments

Jeong, In-Kee Baik, Doo-Kwon
Computer Science Department, Korea University



1. 서론

제2의 산업 혁명이라 불리는 정보 혁명은 컴퓨터 기술의 발달과 통신 기술의 발달에 힘입은 바 크다. 이러한 두 개의 기술은 정보의 처리 및 전달을 위하여 발달된 기술로 정보 혁명의 발달을 가속화시켜 왔으나 지금까지는 서로 독립적으로 발전하여 왔기 때문에 상대편의 기술을 이용할 뿐 두 가지를 모두 고려하여 발전되어 오지 못하였다.

그러나 현대에 와서는 두 가지 기술이 서로 조합되어 발달되고 있으며, 지금까지는 하나의 컴퓨터 시스템에서의 정보 처리를 염두에 두고 개발하던 것들도 많은 지역에 있는 컴퓨터 시스템들의 존재를 인식하고 이를 효과적으로 운용할 수 있는 많은 방법들이 연구되어 이제는 많은 정보 관리 시스템들을 하나로 통합한 환경에서 정보 시스템들을 개발하고 운용하는 시스템 통합 환경(system integration

environment)으로 발전하게 되었다.

시스템 통합 환경의 구축은 여러 지역에 존재하는 데이터베이스들이 서로 유기적인 관계를 가지면서 데이터 및 처리의 중복 요소들을 최대한 제거하고, 데이터의 무결성 및 일관성들을 유지시킬 수 있도록 한다.

그러나 지금까지의 시스템 통합 환경의 구축은 기존의 DBMS들을 통합하는 것에 불과하였으며 시스템 통합 환경에서 데이터베이스의 설계는 기존의 방법론들을 그대로 고수하면서 사람의 경험에 의존하여 타 시스템과의 중복성들을 제거하고 효율적으로 데이터를 분산 배치하는 등의 방법을 사용하여 왔다.

그러나 이러한 방법들은 기존의 단독 시스템에서의 데이터베이스 설계 방법들의 단순한 확장일 뿐 시스템 통합 환경이라는 새로운 환경에 적합한 것은 아니다. 따라서 시스템 통합 환경에 적합한 데이터베이스 설계 방법 및 데이터 모델이 개발되어야 한다.

시스템 통합 환경에 적합한 데이터베이스 설계 방법 및 데이터 모델은 다음과 같은 특징을 가지고 있어야 한다.

첫째, 다른 데이터베이스와의 유기적인 관계를 필요로 한다. 따라서 단독 시스템에서의 데이터베이스의 설계와는 다른 면을 가지고 있어야 한다.

둘째, 시스템 통합 환경에서는 공유 정보들을 저장하고 있는 데이터 저장소(data repository)의 존재가 필요하며, 시스템 통합 환경에서의 새로운 지역 데이터베이스의 구축은 이러한 데이터 저장소의 존재를 인식하고, 참조하는 형태가 되어야 한다.

뿐만 아니라 데이터 저장소 내의 정보들은 체계화되어 있어야 하고, 메타 정보를 가지고 있어야 한다.

셋째, 시스템 통합 환경이 새로운 실행 환경이지만 기존의 데이터베이스 설계 방법과 너무 상이한 방법이어서는 안된다.

따라서 본 논문에서는 Entity-Relationship(ER) 데이터 모델을 확장한 Entity-Structure- Relationship(ESR) 데이터 모델과 ESR 데이터 모델을 관계형 데이터 모델로 변환하는 규칙을 제안한다.

2. Entity-Structure-Relationship 데이터 모델

ER 데이터 모델은 Chen이 1976년에 처음 제안한 데이터 모델로서 실세계의 객체들을 개체(entity)로서 표현하고 개체들 간의 관계(relationship)를 나타내기에 편리하도록 만들어진 데이터 모델이다. 이 모델은 간단하며, 또한 정형화가 가능하여 데이터베이스를 설계하는 데 적당하기 때문에 현재 가장 널리 보급되어 있는 데이터 모델 중의 하나이다.

그러나 초기의 ER 데이터 모델은 개체와 개체들의 관계만을 주로 표현함으로서 개체 자체의 추상화 개념과 관계의 여러 종류를 나타내는 표현력이 부족하였다. Chen의 ER 모델은 데이터의 특성 중에서 개체들 간의 수평적 관계를 잘 표현해 주는 데이터 모델이지만 개체가 내부적으로 가지는 계층 구조 및 추상화를 표현하는데 미흡하였기 때문에 많은 사람들이 개체의 추상화 및 계층 구조를 ER 데이터 모델 내에서 표현하기 위하여 연구하여 왔다. 이러한 연구 결과들은 여러 추상화 개념들을 만들어 왔는데 일반화 (generalization), 집단화 (aggregation) 개념 등이 바로 그것이다. 또한 데이터 계층 사이의 상속 관계를 유지하고자 ER 데이터 모델에 객체 지향 개념을 도입한 경우도 나타났다. Batini, Elmasri, Teorey등은 ER 데이터 모델에 추상화의 개념을 추가하였고, Martin은 관계의 종류를 세분하였는데 이러한 개념의 추가는 ER 데이터 모델의 표현력을 좀 더 풍부하게 해주었다.

그러나 확장된 ER 데이터 모델들도 문제점을

가지고 있다. 즉, 데이터의 수직적 계층 구조와 수평적 관계를 동일한 평면에 표현함으로써 수직적 계층 구조에서 나타나는 추상화의 개념을 자연스럽게 표현하고 있지 못하다는 점이다. 따라서 확장된 ER 데이터 모델은 개체를 추상화된 개체로 대치할 수 있는 개념을 올바르게 표현하기에는 미흡한 점이 있다. 즉, 추상화된 개체라면 추상화된 개체를 존재하는 개체의 위치에 그대로 대치시킬 수 있을 때 그 의미를 좀 더 확실하게 전달할 수 있을 것이다. 일반적으로 실세계의 객체들은 개체로 표현되는데 이것의 내부 속성은 계층 구조를 가지고 있다.

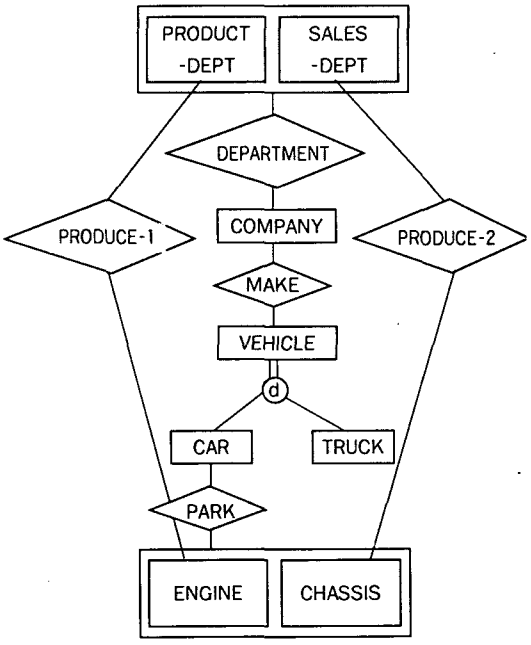
또한 개체들은 다른 개체들과의 유기적인 관계를 가지며 여러 정보들을 표현하게 된다. 실세계의 정보들을 올바르게 표현하려면 내부적으로 나타나는 개체의 수직적 계층 구조와 외부적으로 나타나는 수평적 관계를 별도의 평면에 표현하는 것이 좋을 것이다.

따라서 본 논문에서는 개체의 수직적 계층 구조와 수평적 관계를 별도로 표시하여 나타내는 방법으로 ESR 데이터 모델을 제안한다. 제안한 데이터 모델의 설명을 위하여 데이터베이스 설계 요구 사항과 Elmasri의 EER 도표를 예로 들면 〈그림 2-1〉과 같다.

〈설계 요구 사항〉

COMPANY는 VEHICLE을 만든다.
COMPANY는 PRODUCT-DEPT와 SALES-DEPT로 이루어진다.
VEHICLE은 CAR와 TRUCK으로 나누어진다.
CAR는 ENGINE과 CHASSIS로 이루어진다.
PRODUCT-DEPT에서 ENGINE을 생산한다.
PRODUCT-DEPT에서 CHASSIS를 생산한다.

(a)



(b)

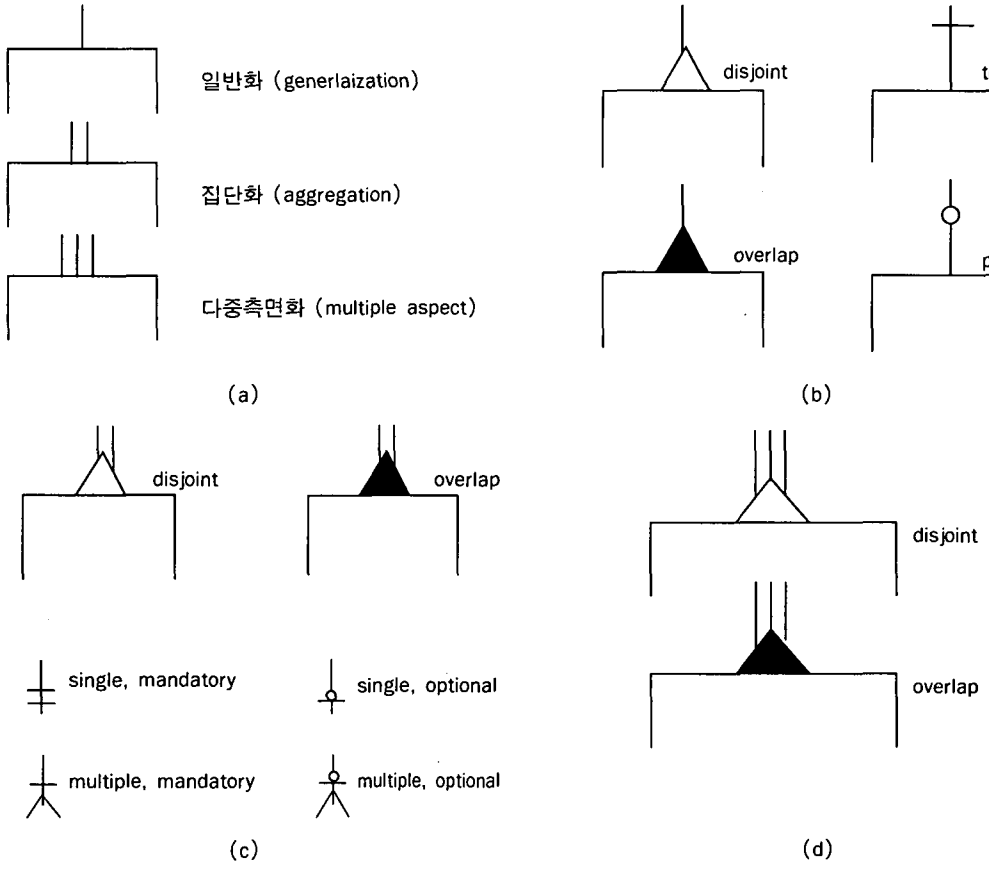
<그림 2-1> Elmasri의 EER diagram의 예

2-1 Entity-Structure 데이터 모델

ES 데이터 모델은 데이터의 수직적 계층 구조를 표현하기 위한 모델이다. 데이터의 수직적 계층 구조는 여러 추상화 개념으로 표현할 수 있는데 ES 데이터 모델에서는 일반화 (generalization), 집단화(aggregation), 다중측면화 (multiple aspect)의 세 가지 추상화 개념을 표현한다. ES 모델에서의 추상화 표현 기호는 〈그림 2-2 (a)〉와 같다.

일반화는 여러 개체들을 하나의 개체로 일반화하는 개념으로 하위 클래스의 여러 개체들을

상위클래스의 하나의 개체로 표현할 수 있도록 하는 추상화 개념이다. 일반화 개념은 IS-A 개념으로 표현하기도 한다. 일반화 계층에 있어서 disjointness 제한은 disjoint와 overlap 관계로 나눌 수 있고, coverage 제한은 total과 partial 관계로 나누어 볼 수 있다. disjoint 관계는 하위 계층의 개체들 사이에 인스턴스들이 상호배타적 관계에 있는 경우를 말하며, overlap 관계는 하나의 인스턴스가 동시에 두 개 이상의 하위 계층의 개체에 속할 수 있는 경우를 말한다. 또한 상위 계층의 개체의 모든 인스턴스가 모두 하위계층의 인스턴스로 나타날 때, 즉, 상위계층의 인스턴스와 하위계층



<그림 2-2> ES 모델에서 사용하는 기호

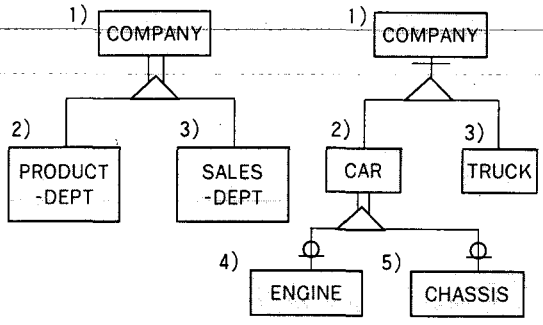
의 인스턴스의 합집합이 같을 때 total 관계라고 하며, 그렇지 않은 경우를 partial 관계라고 한다. 이러한 disjoint/overlap 제한 및 total/partial 제한은 서로 조합으로 나타날 수 있다. 따라서 일반화 관계는 4 가지의 제한 관계로 나타날 수 있다. 그림 2-2 (b)는 일반화 계층을 표시하는 기호를 나타내고 있다.

집단화는 하위 계층의 개체들이 상위 계층의 개체의 요소로 구성되는 추상화이다. 따라서 IS-PART-OF의 관계로 나타난다. 집단화에서 disjointness 제한은 disjoint 및 overlap 관계로 나누어 볼 수 있고, connectivity 제한은 single 및 multiple로 나누어 볼 수 있다.

또한, participation 제한은 mandatory 및 optional의 관계로 나누어 볼 수 있다. disjoint와 overlap 관계는 일반화 계층에서의 관계와 동일하다. connectivity 제한에 있어서 single은 일대일의 관계를 표현하고, multiple은 일대다의 관계를 표현한다. 개체들의 인스턴스들이 모두 관계에 참가하느냐는 participation 제한은 man-datory와 optional 관계로 표현하는데 mandatory는 관계에 참여하는 개체들의 모든 인스턴스들이 관계에 참여하는 것이고, optional 관계는 관계하는 개체들의 모든 인스턴스들이 관계에 참여하지 않을 수도 있는 형태이다. 집단화는 이러한 여러 제한들의 조합 형태로 나타나므로 모두 8 가지의 제한 관계를 가진다. 〈그림 2-2 (c)〉는 집단화 계층의 기호를 나타내고 있다.

다중측면화는 하나의 개체를 서로 다른 측면에서 바라볼 때 다른 개체처럼 보이는 것을 추상화한 것으로 disjointness 제한에 따라 disjoint와 overlap 관계로 나누어 볼 수 있다. 이 관계들은 일반화 및 집단화에서의 disjoint와 overlap 관계와 개념이 동일하다. 그림 2-2 (d)는 다중측면화 계층에서 사용하는 기호를 나타내고 있다.

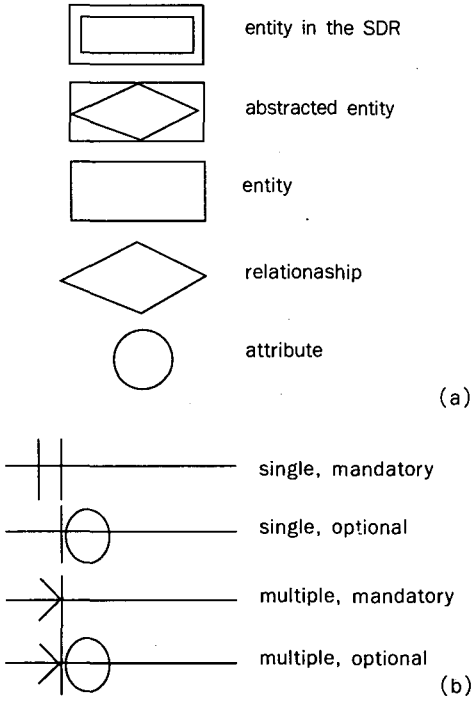
〈그림 2-1〉의 예 중에서 개체들을 ES 도표로 표현하면 〈그림 2-3〉과 같다.



<그림 2-3> 그림 2-1의 Entity-Structure 도표

2-2 Entity-Relationship 데이터 모델

ER 데이터 모델에서는 개체의 위치에 따라 local 및 SDR (Shared Data Repository)에 존재하는 개체로 나누고, 추상화 여부에 따라 추상화된 개체와 독립 개체로 나눈다. 개체의 표현 기호는 〈그림 2-4(a)〉에 나타내었다.



<그림 2-4> Entity-Relationship 모델에서 사용하는 기호

여기에서 SDR이란 공유 데이터를 통합 관리하는 데이터 저장소를 가리킨다. 이와 같이 개체의 추상화 여부와 위치를 표현함으로써 시스템 통합 환경에 적용이 용이해진다.

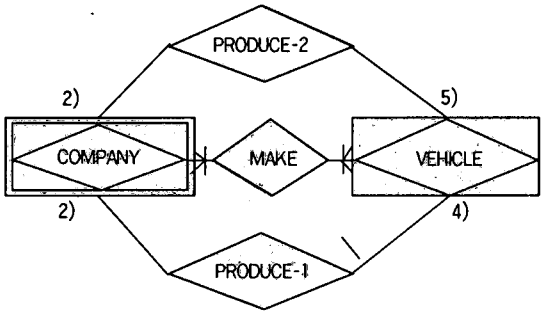
ER 데이터 모델에서는 관계의 종류를 degree에 따라 unary, binary 및 ternary로 나누었고, connectivity에 따라 single과 multiple로 나누었다. 또한 관계의 participation에 따라 mandatory와 optional로 분류하였다. unary 관계는 동일한 개체 내부에서의 관계를 표현하고, binary 관계는 두 개체 사이의 관계를 표현한다.

또한 ter-nary 관계는 세 개체 사이의 관계를 표현한다. single은 일대일의 관계를 의미하며, multiple은 일대다의 관계를 의미한다. 또한 다대다의 관계는 일대다의 관계인 multiple의 조합으로 이루어진다. mandatory는 개체의 모든 인스턴스가 관계에 참여하는 것을 의미하며, 그렇지 않은 경우는 optional이 된다.

degree는 개체에 연결된 선의 수로 표현되고, connectivity 및 participation은 조합으로 표현되어 4 가지의 표현 방법이 된다.

이러한 관계의 표현은 Martin의 표현과 유사하다. 〈그림 2-4 (b)〉는 ER 모델의 관계를 표시하는 기호를 나타낸다.

〈그림 2-1〉의 예 중에서 개체들간의 관계를 ER 도표로 표현하면 〈그림 2-5〉와 같다.

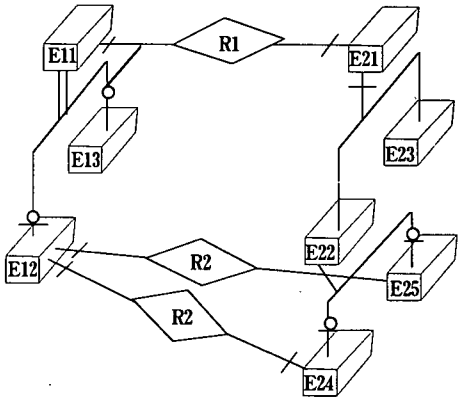


<그림 2-5> 그림 2-1의 Entity-Relationship 도표

2-3 ESR 데이터 모델 : 3 차원 데이터 모델

Entity-Structure-Relationship(ESR) 데이터 모델은 데이터의 수직적 계층 구조와 수평적 관계를 보다 효과적으로 표현하기 위하여 제안한 데이터 모델이며, 개체의 추상화 계층 구조와 개체들의 관계를 3 차원 평면상에 표현한다.

즉, 데이터의 수직적 계층 구조는 ES(Entity-Structure) 데이터 모델로 표현하고, 데이터의 수평적 관계는 ER(Entity-Relationship) 데이터 모델로 표현하여 데이터 모델을 3 차원으로 나타낸다. ESR 데이터 모델의 개념을 그림으로 표현하면 〈그림 2-6〉과 같다.



<그림 2-6> 그림 2-1의 ESR 데이터 모델 개념도

E11:COMPANY
E12:PRODUCT-DEPT
E13:SALES-DEPT
E21:VEHICLE
E22:CAR
E23:TRUCK
E24:ENGINE
E25:CHASSIS
R1:MAKE
R2:PRODUCE-1
1R3:PRODUCE-2

3. 관계형 데이터 모델로의 변환

ER 및 EER 데이터 모델은 실세계를 표현하는 능력은 탁월하지만 ER 데이터 모델을 그대로 구축할 수 있는 데이터베이스 관리 시스템은 상용화된 것이 거의 전무한 상태이다. 따라서 ER 데이터 모델로 설계된 정보 시스템은 상용화된 데

이터베이스 관리 시스템의 데이터 모델로 변환시켜 주어야 한다. 이미 Chen은 ER 데이터 모델을 계층형 데이터 모델 (Hierarchical Data Model), 네트워크형 데이터 모델 (Network Data model), 관계형 데이터 모델 (Relational Data Model)로 변환 가능함을 보여주었고, 현재는 많은 사람들이 객체지향 데이터 모델 (Object-Oriented Data Model)로의 변환을 연구하고 있는 상태이다.

Chen의 ER 데이터 모델은 계층형 데이터 모델, 네트워크형 데이터 모델, 관계형 데이터 모델로 모두 변환할 수 있는 특성을 가지고 있으며, 이는 ER 데이터 모델이 가지는 장점 중의 하나이다. 따라서 ER 데이터 모델을 확장시키고자 하는 경우에는 이러한 특성을 그대로 간직하고 있어야 한다.

그런데, 현대에 주로 구축되는 데이터베이스는 관계형 데이터 모델을 기반으로 하는 것이 많으므로 관계형 데이터 모델로의 변환을 주로 시도하고 있다. 따라서 본 논문에서도 ESR 데이터 모델을 관계형 데이터 모델의 테이블로 변환하는 규칙을 제시한다.

3-1 ES 데이터 모델의 변환

ES 데이터 모델은 개체의 추상화 개념을 표현하는 데이터 모델로서 실세계의 객체를 자연스럽게 표현할 수 있는 능력을 가지고 있다. 그러나 관계형 데이터 모델은 이러한 능력을 가지고 있지 못하다. 따라서 ES 데이터 모델로부터 관계형 데이터 모델로 변환을 하는 경우에는 개체들 사이의 연관 관계의 존재만을 표현하게 되며 의미는 잃어버리게 된다.

또한 추상화 개념은 서로 상이한 개념을 가지고 있으므로 동일한 변환 과정을 통하여 변환할 수 없다.

따라서 각 추상화 개념마다 다른 변환 규칙을 적용하게 된다. 각 추상화 개념에 따른 변환 규칙은 다음과 같다.

(1) 일반화 개체

일반화 개념은 2.1절에서 언급한 바와 같이 4가지로 분류할 수 있다. 이와 같은 4 가지의 일반화 개체에 대하여 관계형 데이터 모델로 변환하는 규칙이 <그림 3-1>에 나타나 있다.

알고리즘 GEN-TO-RDM

```
Begin
 if (disjointness = disjoint and coverage =
  total) then
  하위클래스 개체들만 테이블로 변환;
  상위클래스 개체의 속성들은 하위클래스 개
  체의 테이블에 포함;
 else
  상위클래스 개체를 테이블로 변환;
  하위클래스 개체를 테이블로 변환;
 endif;
End.
```

<그림 3-1> 일반화 개체의 변환

(2) 집단화 개체

집단화 개념은 2-1절에서 언급된 바와 같이 8가지로 분류할 수 있다. 이와 같은 8 가지의 집단화 개체에 대하여 관계형 데이터 모델로 변환하는 규칙이 <그림 3-2>에 나타나 있다.

(3) 다중측면화 개체

다중측면화 개념은 2-1절에서 언급된 바와 같이 2 가지로 분류할 수 있다. 이와 같은 2 가지의 다중측면화 개체에 대하여 관계형 데이터 모델로 변환하는 규칙이 <그림 3-3>에 나타나 있다.

```
알고리즘 AGR-TO-RDM

Begin
 if (disjointness = disjoint and participation =
  mandatory) then
  해당 경우 없음;
 else
 if (connectivity = single ) then
  상위클래스 개체를 테이블로 변환;
  하위클래스 개체를 테이블로 변환;
  하위클래스의 키 속성을 상위클래스 개체의 테
  이블에 포함;
 else
  상위클래스 개체를 테이블로 변환;
  하위클래스 개체를 테이블로 변환;
  각 하위클래스에 대하여 상위클래스와 하위클
  래스 개체의 키 속성을 복합키로 하는 테이블
  생성;
 endif;
 endif;
End.
```

<그림 3-2> 집단화 개체의 변환

```
알고리즘 ASP-TO-RDM

Begin
 superclass만 table로 생성;
 subclass entity의 attribute는 생성된 table에 추가;
 subclass는 view로 생성;
End.
```

<그림 3-3> 다중측면화 개체의 변환

3-2 ER 데이터 모델의 변환

Chen이 처음 ER 데이터 모델을 제안할 때부터 관계형 데이터 모델로의 변환은 매우 자연스러웠다. 그러나 관계의 종류를 세분하여 나타내지 않음으로써 변환시 고려 사항에 대한 정보를 충분히 제공하지 못하는 단점이 있었다. 이러한 문제점을 해결하고자 Martin은 관계의 종류를 연결수와 참여도에 따라 5 가지로 나누어 표현함으로써 관계형 데이터 모델로 변환시에 편리하도록 하였다. 본 논문에서도 관계형 데이터 모델로의 변환을 고려하여 관계의 종류를 2-2절에서 언급한 바와 같이 4 가지로 분류하였으며 이진 관계(Binary Relationship)인 경우의 변환 규칙을 보이면 <그림 3-4>과 같다.

```
알고리즘 ER-TO-RDM

Begin
 entity를 table로 생성;
 table의 attribute들은 entity attribute로 구성;
 table의 key는 entity의 key로 구성;
 if (양쪽의 entity가 모두 multiple) then
  relationship에 해당하는 table 생성;
 elseif (한 쪽이 multiple) then
  multiple인 entity가 변환되는 table에
  single인 entity인 entity의 key attribute를 삽입;
 elseif (한 쪽이 optional) then
 optional인 entity가 변환되는 table에
  mandatory인 entity의
 key attribute를 삽입;
 else
  어느 한 쪽이 entity에 다른 entity의 key
  attribute를 추가;
 endif;
 if (상대편 entity가 single이고, optional) then
  null 값 허용;
 else
  null 값 허용 안함;
 endif;
End.
```

<그림 3-4> Entity-Relationship 데이터 모델의 변환

3-3 ESR 데이터 모델의 변환

ESR 데이터 모델은 2.3절에서 언급한 바와 같이 ES 데이터 모델과 ER 데이터 모델을 교차시켜 표현하는 3 차원적 모델이다.

따라서 ESR 모델에서 나타나는 관계의 수는 ES 데이터 모델에서의 개체의 종류와 ER 데이터 모델에서의 관계의 수를 곱한 수이다.

ES 데이터 모델에서의 개체의 종류는 여러 추상화 개념이 복합적으로 나타나지 않는 경우만 생각하더라도 일반화 개체 4 가지, 집단화 개체 8 가지, 다중측면화 개체 2 가지 등으로 모두 14 가지를 가지고 있다.

알고리즘 ESR-TO-RDM

```
Begin
 (SDR의 ES 데이타 모델 변환);
 지역 DB의 ES 데이타 모델 변환;
 if (두 entity가 모두 SDR에 존재) then
  관계에 해당하는 table 생성;
 elseif (한 entity는 SDR에 존재하고, 다른 entity
  는 지역 DB에 존재) then
 if (SDR의 entity가 multiple) then
 관계에 해당하는 table 생성;
 elseif (SDR의 entity가 mandatory이고 single)
  then
  SDR의 entity의 key를 지역 DB의 table에 추가;
 else
  SDR entity의 key를 지역 DB의 table에 추가;
  추가된 attribute는 null 값 허용;
 endif;
 else
  알고리즘 ER-TO-RDM 적용;
 endif;
End.
```

<그림 3-5> 공유 데이터 저장소를 고려한 ESR 데이터 모델의 변환

또한 ER 데이터 모델에서의 관계의 수는 degree 면에서 3 가지, 그 밖의 개념의 조합으로 8 가지를 가지고 있으며, binary 및 ternary의 경우에 각 개체의 종류가 다르게 조합되는 경우를 상정해 볼 수 있다.

따라서 이러한 경우의 수를 계산해 보면 최소한 (14+1) * 2 * 3 * 8 = 720 가지에 달하며, 좀 더 복잡한 경우를 상정하면 방대한 양의 경우의 수가 발생하므로 일반적인 방법으로는 모든 경우의 수를 고려하기 힘들다.

그러나 공유 데이터 저장소가 존재하는 시스템 통합 환경에서는 공유 데이터 저장소 내의 개체들의 구조는 수정하지 않는다는 제한 사항이 발생하므로 변환 과정을 좀 더 단순화시킬 수 있다.

따라서 본 논문에서는 시스템 통합 환경에서 공유 데이터 저장소 내의 개체를 참조하는 경우의 변환 과정을 제시한다.

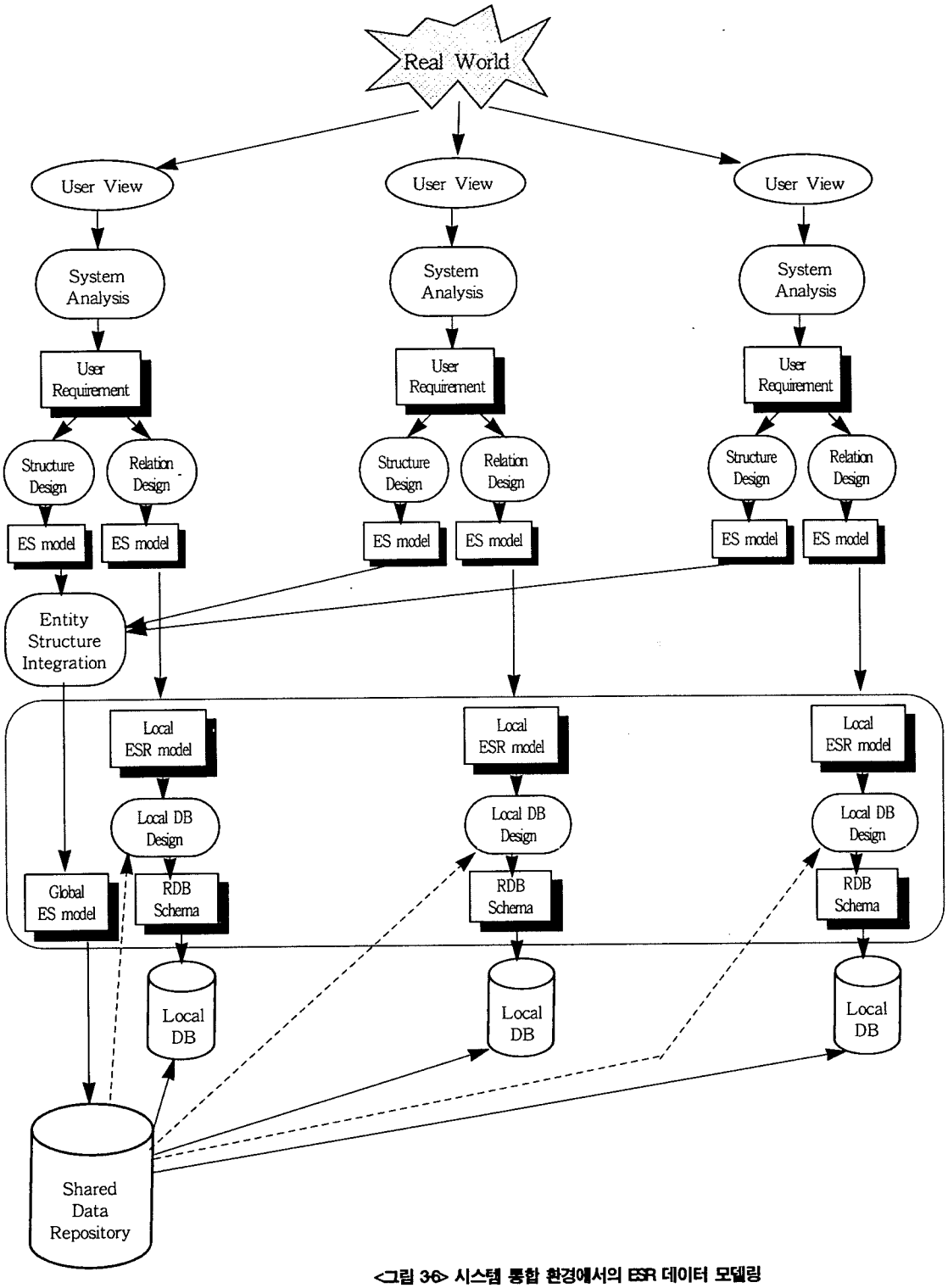
ESR 데이터 모델을 관계형 데이터 모델로 변환하는 과정에 있어서 여러가지 문제점이 발생하는 데 그 중 하나가 ESR 모델에서 관계를 나타내는 부분이 변환과정에서 독립적인 테이블로 생성되지 않고 연관된 개체의 테이블로 추가되는 경우가 발생한다.

이러한 경우에 어느 개체에 포함시킬 것인가를 결정하는 것은 뚜렷한 기준이 없다.

또한 단독 시스템에서의 데이터베이스를 구축하는 경우에는 생성되는 테이블 구조에 약간의 변경이 가해져도 상관없지만 시스템 통합 환경하에서 공유 데이터 저장소의 존재를 인식하고 데이터베이스를 구축하고자 하는 경우에는 공유 데이터 저장소에 저장되어 있는 테이블은 수정되지 않는 것이 좋다.

따라서 시스템 통합 환경에서의 공유 데이터 저장소(SDR)를 고려하는 경우의 변환 규칙은 약간의 수정이 가해져야 한다.

SDR을 고려한 ESR 데이터 모델의 변환 규칙 중 이진 관계인 경우는 <그림 3-5>와 같다.



<그림 36> 시스템 통합 환경에서의 ESR 데이터 모델링

3-4 시스템 통합 환경에서의 ESR 데이터 모델링

본 논문에서 제안하는 시스템 통합 환경에서의 ESR 데이터 모델링 방법론을 설명하면 다음과 같다.

대상이 되는 실세계의 모델은 사용자의 관점에서 각 응용 영역마다 다른 관점으로 보여진다. 따라서 초기 데이터 모델링은 각 사용자 관점에서 이루어지며, 이 과정에서는 각 사용자 관점의 응용 영역마다 사용자 요구사항들이 만들어진다.

각 사용자 관점에서의 응용은 데이터의 구조를 나타내는 ES 데이터 모델과 데이터 개체들 간의 관계를 표현하는 ER 데이터 모델로 표현된다. 각 사용자 요구사항으로부터 작성된 ES 데이터 모델로부터 공유 데이터를 도출하여 공유 데이터 저장소 (SDR)에 반영한다.

SDR에 반영되지 않은 ES 데이터 모델상의 개체들과 ER 데이터 모델은 지역 데이터베이스를 구축하는 데 이용한다. 따라서 각 사용자 요구사항의 모든 사항은 데이터베이스 구축에 반영된다.

본 논문에서 제안하는 데이터베이스 설계 방법론에서 공유 데이터 저장소 (SDR)내의 데이터는 ES 데이터 모델로만 표현된다. 공유 데이터 저장소내의 개체들은 주로 데이터 개체들만을 표현하고, 개체들 간의 관계는 지역 데이터베이스에서 표현한다.

따라서 지역 데이터베이스 구축은 ESR 데이터 모델을 사용한다. 시스템 통합 환경에서의 ESR 데이터 모델링 방법론의 개념도는 〈그림 3-6〉과 같다.

시스템 통합 환경에서의 데이터 모델링은 공유 데이터를 저장하고 관리하는 공유 데이터 저장소 (SDR)의 존재를 인식하고, 이를 바탕으로 데이터 모델링하는 것이 기존의 단독 시스템에서의 데이터베이스 설계시 수행되는 데이터 모델링과 다른 점이다. 이와 같은 고려사항은 설계하고자 하는 데이터 개체를 모두 지역 데이터베이스에서 생성하는 것이 것이 아니라 SDR 내에 존재하는 개체를 참조하여 모델링함으로써 데이터의 중복성을 제거하고, 일관성 및 무결성을 유지할 수 있도록 한다.

또한 SDR내의 개체들은 이미 구축되어 있는 상태의 개체들이므로 구성 속성들이 변화할 수 없다는 제한점이 있다. 따라서 ER 데이터 모델에서의 관계를 변화하고자 하는 경우에 독립적인 테이블로 변환되거나 지역 데이터베이스의 개체가 변환되는 테이블에 추가하여야 한다.

4. 결론

정보화 사회로 발전해 감에 따라 사회 전반에서 발생하는 많은 정보들이 정보 관리 시스템을 거쳐 사람들이 사용할 수 있게 되었다. 정보 통신 기술의 비약적인 발전의 도움을 받아 데이터베이스 또한 단독 시스템에서 독립적으로 데이터를 관리하는 것이 아니라 시스템 통합 환경 하에서 다른 데이터베이스와 서로 유기적인 관계를 가지면서 정보를 교환하게 되었다. 이러한 환경에서의 데이터 모델링은 과거와는 다르게 공유 데이터 저장소의 존재를 인식하고, 포함하는 형태의 데이터 모델링 방법론이 필요하게 되었다.

따라서 본 논문에서는 실세계를 모델링하는 과정에서 나타나는 개체의 수직적 계층 구조와 수평적 관계를 분리하여 모델링하며 3차원적인 특성을 가지는 ESR 데이터 모델을 제안하고, 시스템 통합 환경 하에서 ESR 데이터 모델로 설계된 모델을 관계형 데이터 모델로 변환하는 규칙을 제시하였다.

본 논문에서 제안하는 데이터베이스 설계 방법론은 현재 데이터 모델링 CASE 도구의 프로토타입으로 개발 중이며, 앞으로 시스템 통합 환경 하에서 데이터 모델링을 하는 경우에 많은 도움을 주게 될 것이다. DB