

DB Technology  
and Management

# 시스템 통합환경을 고려한 데이터 모델링 도구의 설계

정인기  
고려대학교

백두권  
고려대학교



# 시스템 통합 환경을 고려한 데이터 모델링 도구의 설계

정인기, 백두권  
고려대학교 전산학과

## 요 약

고도 정보화 사회로 발전해 감에 따라 사회 전반에서 발생하는 정보들을 컴퓨터에 저장하여 관리하는 정보 관리 시스템들이 많이 개발되고 있다. 특히 컴퓨터 기술의 발달과 통신의 발달은 서로 떨어져 있는 정보 시스템끼리의 정보 교환을 보다 효율적으로 할 수 있게 되어 시스템 통합 환경으로 선택되고 있다.

이에 본 논문에서는 시스템 통합 환경에서 공유 데이터 저장소를 기반으로 하여 지역 데이터베이스를 구축할 수 있는 데이터 모델인 ESR 데이터 모델을 제안하고, 그에 따른 데이터 모델링 방법론을 제안하였다. 또한 공유 데이터 저장소를 기반으로 하여 ESR 데이터 모델링을 통한 데이터베이스를 설계할 수 있는 지원도구를 설계하였다.

## 1. 서론

제 2 의 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 발달 가속화시켜 왔으나 지금까지는 서로 독립적으로 발전하여 왔기 때문에 상대방의 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 이제는 많은 지역에 있는 컴퓨터 시스템들의 존재를 인식하고 이를 효과적으로 운용할 수 있는 많은 방법들이 연구되고 있다. 이제는 많은 정보 관리 시스템들을 하나로 통합한 환경에서 정보 시스템들을 개발하고 운용하는 시스템 통합 환경(system integration environment)으로 발전하게 되었다. [13]

시스템 통합 환경의 구축은 여러 지역에 존재하는 정보 관리 시스템들이 서로 유기적인 관계를 가지면서 데이터 및 처리의 중복 요소들을 최대한 제거하고, 데이터 무결성 및 일관성들을 유지시킬 수 있다. [13] 이러한 시스템 통합 환경의 구축은 지금까지는 기존의 여러 시스템들을 통합하는 것에 불과하였으며 시스템 통합 환경에서의 정보 관리 시스템의 개발은 기존의 방법론들

을 그대로 고수하면서 사람의 경험에 의존하여 타 시스템과의 중복성들을 제거하고 효율적으로 정보를 분산 배치하는 등의 방법을 사용하여 왔다. 그러나 이러한 방법들은 기존의 단독 시스템에서의 정보 관리 시스템 개발 방법들의 단순한 확장일 뿐 시스템 통합 환경이라는 새로운 환경에 적합한 것은 아니다.

시스템 통합 환경은 첫째, 다른 정보 관리 시스템과의 유기적인 관계를 필요로 한다. 따라서 단독 시스템에서의 정보 관리 시스템의 개발과는 다른 면을 가지고 있어야 한다. 둘째, 시스템 통합 환경에서는 공유 정보들을 저장하고 있는 데이터 저장소(data repository)의 존재가 필요하며[1], 시스템 통합 환경에서의 새로운 정보 관리 시스템의 개발은 이러한 데이터 저장소의 존재를 인식하고, 참조하는 형태가 되어야 한다. 뿐만 아니라 데이터 저장소의 내의 정보들은 체계화되어 있어야 하고, 메타 정보를 가지고 있어야 한다. 셋째, 시스템 통합 환경이 새로운 실행 환경이지만 기존의 여러 정보 관리 시스템의 개발 방법과 너무 상이한 방법이어서는 안된다.

이상과 같이 언급된 여러 사항들을 분석하여 보면 기존의 단독 시스템에서의 정보 관리 시스템의 개발 방법과 유사하면서도 시스템 통합 환경에서의 특성을 고려한 새로운 개발 방법론이 필요하다는 것을 알 수 있다.

따라서 본 논문에서는 시스템 통합 환경에서의 데이터베이스 구축 방법론을 소개하고, 이러한 방법론을 따라 데이터베이스를 구축할 수 있는 데이터 모델링 설계 도구를 설계하였다.

## 2. 시스템 통합 환경에서의 데이터 모델링

시스템 통합 환경에서는 많은 정보 관리 시스템들이 서로 유기적인 관계를 유지하도록 연결되어 있다. 이러한 시스템 통합 환경에서는 관계되는 정보 관리 시스템 사이의 여러 데이터들이 유기적인 관계들을 유지하여야 하므로 공유하는 데이터들의 중복성을 제거하고, 일관성 및 무결성을 유지시켜 주어야 한다. 이러한 문제점들을 해결하기 위해서는 관련되는 정보 관리 시스템들이 공유하는 데이터들을 한 곳에서 관리할 수 있는 전용의 시스템을 마련하여 공유 데이터들을 관리하는 것이 효과적인 방법이다.

이러한 공유 데이터 관리 시스템은 그 개념이 조금씩은 다르지만 데이터 사전 (data dictionary)[4], 데이터 저장소 (data repository)[10], 백과사전 (encyclopedia)[11], 및 정보 자원사전 시스템 (IRDS : Information Resource Dictionary System)[12] 등으로 정립되고 있다. 이러한 개념의 시스템들은 공유 데이터보다는 공유하는 데이터의 메타 정보들을 관리하는 것이 대부분이다.

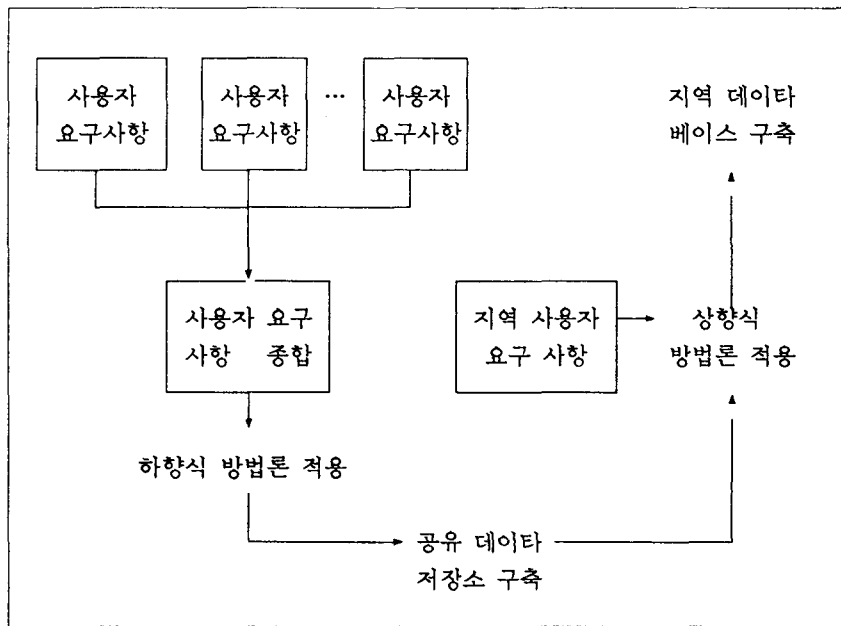
공유하는 데이터의 메타 정보를 별도로 관리하는 것은 시스템 통합 환경에서 데이터를 효과적으로 관리하는 방법 중의 하나이다. 그러나 시스템 통합 환경에 있어서 메타 정보를 별도로 관리하는 것보다는 메타 정보와 공유 데이터를 함께 관리하는 것이 효과적이다.[2] 또한 지역 데이터베이스 설계 시에는 중앙에서 관리하는 공유 데이터의 존재를 인식하고 이들과의 유기적인 관

계를 설정할 수 있는 데이터 모델링 방법이 필요하다.

따라서 본 논문에서는 시스템 통합 환경에서 정보 관리 시스템들이 공유하는 데이터들을 관리하는 시스템을 공유 데이터 저장소 (SDR : Shared Data Repository)라고 부르고자 한다. 공유 데이터 저장소는 시스템 통합 환경에서 정보 관리 시스템들의 공유 데이터 및 데이터의 메타 정보들을 저장·관리한다.

시스템 통합 환경에서의 데이터 모델링은 공유 데이터 저장소 (SDR)의 존재를 기반으로 할 때 기존의 시스템 분석 및 설계 방법론인 하향식 방법론 (Top-Down Methodology)을 그대로 수용하기 어렵다. 따라서 본 논문에서는 각 사용자 요구사항으로부터 공유 데이터 저장소를 구축한 후 지역 데이터베이스 구축 시에는 상향식 방법론 (Bottom-Up Methodology)으로 공유 데이터 저장소를 참조하며 구축하는 U 턴 방법론을 제안한다.

U 턴 방법론에서는 공유 데이터 저장소 구축 시에는 하향식 방법론을 수용하고, 지역 데이터베이스 구축 시에는 상향식 방법론을 수용한다. U 턴 방법론을 <그림 2-1>에 나타내었다.



<그림 2-1> U 턴 시스템 구축 방법론

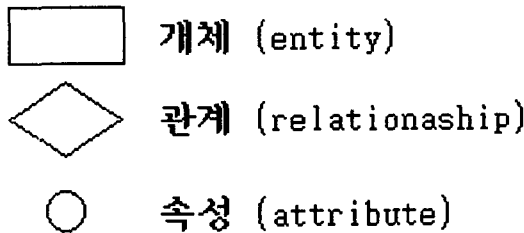
U 턴 방법론에는 데이터 모델은 ER 데이터 모델, DFD 등 대부분의 데이터 모델이 적용 가능하다. 그러나 공유 데이터 저장소의 데이터들은 주로 실세계에서 정적인 계층 구조를 가지는 개체 단위들의 집합인 경우가 대부분이다. 이들이 동적인 관계를 유지하는 것은 실제로 데이터를 조회하고, 처리하는 지역 데이터베이스 내에서의이다. 따라서 공유 데이터 저장소에서는 개체의 정적 계층 구조를 위주로 하고, 지역 데이터베이스에서는 개체들의 동적 관계를 위주로 하는 데이터 모델을 사용하는 것이 바람직하다.

그러나 U 턴 방법론에 적합한 데이터 모델이 없기 때문에 본 논문에서는 Chen[6]의 ER 데이터 모델과 김창화[14], 조동영[15]의 EA 데이터 모델을 결합하고 보완한 Entity-Structure-Relationship (ESR) 데이터 모델을 제안하고, 그에 따른 데이터 모델링 방법론을 제시하였다. 또한, 시스템 통합 환경에서 ESR 데이터 모델링을 통하여 데이터 모델을 구축할 수 있는 CASE 도구를 설계하였다.

### 3. Entity-Structure-Relationship 데이터 모델

#### 3.1 Entity-Relationship 데이터 모델

ER 데이터 모델은 Chen[6]이 처음 제안한 데이터 모델로서 실세계의 객체들을 개체로서 표현하고 개체들 간의 관계를 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. ER 데이터 모델은 개체, 관계 및 속성으로 표현되며 이를 기호로 나타내면 <그림 3-1>과 같다.

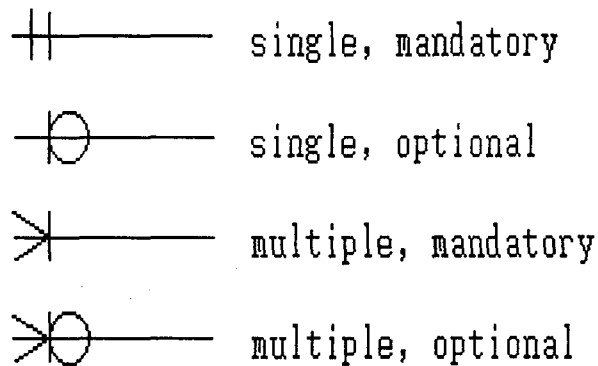


<그림 3-1> ER 데이터 모델에서 사용하는 기호

ER 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 데이터베이스 관리 시스템의 데이터 모델로 변환시켜 주어야 한다. 이미 Chen[6]은 ER 데이터 모델을 계층형 데이터 모델 (Hierarchical Data Model), 네트워크형 데이터 모델 (Network Data model), 관계형 데이터 모델 (Relational Data Model)로 변환 가능성을 보여주었고, 현재는 많은 사람들이 객체지향 데이터 모델 (Object-Oriented Data Model)로의 변환을 연구하고 있는 상태이다.

그러나 초기의 ER 데이터 모델은 개체와 개체들의 관계만을 주로 표현함으로써 개체 자체의 추상화 개념과 관계의 여러 종류를 나타내는 표현력이 부족하였다. 이러한 문제점들에 대한 해결책에 대하여 많은 연구가 진행되었는데 Batini[8], Elmasri[9], Teorey[7]등은 ER 데이터 모델에 추상화의 개념을 추가하였고, Martin은 관계의 종류를 세분화하였는데 이러한 개념의 추가는 ER

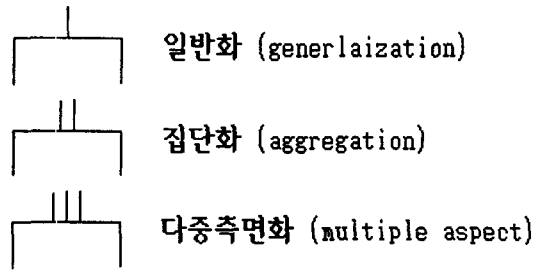
데이터 모델의 표현력을 좀 더 풍부하게 해주었다. 본 논문에서는 개체에 대한 추상화 개념은 ES 데이터 모델에서 다루어지므로 ER 데이터 모델에서는 Martin이 제안한 관계의 종류를 주로 참조하였다. 관계를 세분화하는 것은 관계형 데이터 모델로의 변환시 효율적으로 변환하는 데 많은 도움을 주게 된다. 본 ER 데이터 모델에서는 관계의 종류를 degree에 따라 unary, binary 및 ternary로 나누었고, 연결 상태에 따라 single과 multiple로 나누었다. 또한 관계의 참여도에 따라 mandatory와 optional로 분류하였다. unary 관계는 동일한 개체 내부에서의 관계를 표현하고, binary 관계는 두 개체 사이의 관계를 표현한다. 또한 ternary 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, 연결 상태 및 관계 참여도는 조합으로 표현되어 4 가지의 표현 방법이 된다. 이러한 관계의 표현은 Martin의 표현과 유사하다. <그림 3-2>는 ER 모델에서의 관계를 표현하는 기호를 나타낸다.



<그림 3-2> 관계의 표현

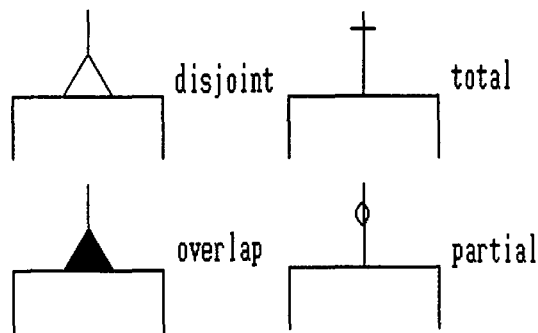
### 3.2 Entity-Structure 데이터 모델

본 논문에서 제안하는 ES 데이터 모델은 데이터의 수직적 계층 구조를 표현하기 위한 모델이다. 데이터의 수직적 계층 구조는 여러 추상화 개념으로 표현할 수 있는데 여러가지 추상화 개념 중에서 가장 대표적인 것은 일반화 (generalization), 집단화 (aggregation) 등의 개념[9]으로 파악할 수 있으며 그 외에 범주화 (categorization)[8], 다중측면화 (multifaceted aspect)[14,15] 등이 있다. 이러한 여러 추상화 개념 중에서 본 ES 데이터 모델에서는 일반화, 집단화, 다중측면화의 세 가지 추상화 개념을 표현한다. 일반화, 집단화, 다중측면화 개념을 본 논문에서는 <그림 3-3>과 같이 표현한다.



〈그림 3-3〉 ES 데이터 모델에서 사용하는 기호

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 하위계층간의 관계는 disjoint와 overlap 관계로 나눌 수 있고, 상위 계층의 개체와 하위 계층의 개체간의 관계는 total과 partial 관계로 나누어 볼 수 있다. 즉, 하위계층 간의 관계에 있어서 disjoint 관계는 하위 계층의 개체들 사이에 인스턴스들이 상호 배타적 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 하위계층의 인스턴스의 합집합이 같을 때 total 관계라고 하며, 그렇지 않은 경우를 partial 관계라고 한다. 이러한 disjoint/overlap 관계 및 total/partial 관계는 서로 조합으로 나타날 수 있다. 따라서 일반화 관계는 4 가지의 관계로 나타날 수 있다. 〈그림 3-4〉는 일반화 계층을 표시하는 기호를 나타내고 있다.

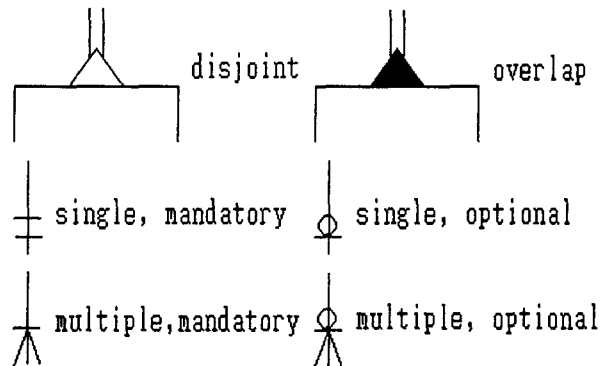


〈그림 3-4〉 일반화 계층의 표시

집단화는 하위 계층의 개체들이 상위 계층의 개체의 요소로 구성되는 추상화이다. 따라서 IS-PART-OF의 관계로 나타난다. 집단화에서도 상위 및 하위 계층 사이의 관계를 여러 가지로 나누어 볼 수 있는데, 먼저 하위 계층의 개체들 사이의 관계는 disjoint 및 overlap 관계로 나누어

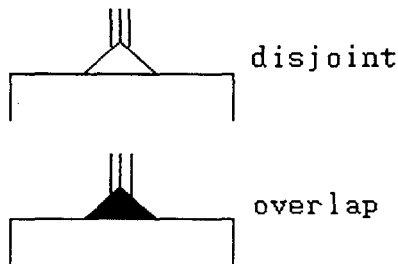


볼 수 있고, 상위 계층의 개체와 하위 계층의 개체들 간의 관계는 single 및 multiple 그리고 mandatory 및 optional의 관계로 나누어 볼 수 있다. 하위 계층의 개체들 간의 관계에 있어서 disjoint와 overlap 관계는 일반화 계층에서의 관계와 동일하다. 상위 계층의 개체와 하위 계층의 개체들 간의 관계에 있어서 single은 일대일의 관계를 표현하고, multiple은 일대다의 관계를 표현한다. 개체들의 인스턴스들이 모두 관계에 참가하느냐는 mandatory와 optional 관계로 표현하는데 mandatory는 관계에 참여하는 개체들의 모든 인스턴스들이 관계에 참여하는 것이고, optional 관계는 관계하는 개체들의 모든 인스턴스들이 관계에 참여하지 않을 수도 있는 형태이다. 집단화는 이러한 여러 관계들의 조합 형태로 나타나므로 모두 8 가지의 관계를 가진다. <그림 3-5>은 집단화 계층의 기호를 나타내고 있다.



<그림 3-5> 집단화 계층의 표시

다중측면화는 하나의 개체를 서로 다른 측면에서 바라볼 때 다른 개체처럼 보이는 것을 추상화한 것으로 하위 계층의 개체들 간의 관계에 따라 disjoint와 overlap 관계로 나누어 볼 수 있다. 이 관계들은 일반화 및 집단화에서의 disjoint와 overlap 관계와 개념이 동일하다. <그림 3-6>은 다중측면화 계층에서 사용하는 기호를 나타내고 있다.



<그림 3-6> 다중측면화 계층의 표시

### 3.3 ESR 데이터 모델 : 3 차원 데이터 모델

ER 모델은 처음에 개체와 관계를 주로 표현하는 데 불과하여 개체들의 추상화 개념을 표현하는 데는 부족하였으므로 많은 사람들이 ER 모델에 추상화 개념을 추가하는 데 많은 연구를 기울였다.

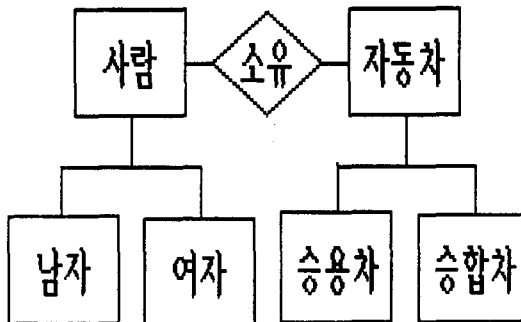
Chen[6]의 ER 모델은 데이터의 특성 중에서 수평적 관계를 잘 표현해 주는 데이터 모델이다. 그러나 Chen의 ER 데이터 모델은 데이터가 내부적으로 가지는 계층 구조 및 추상화를 표현하는데 미흡하였기 때문에 많은 사람들이 추상화 및 계층 구조를 ER 데이터 모델 내에서 표현하기 위하여 연구하여 왔다. 이러한 연구 결과들은 여러 추상화 개념들을 만들어 왔는데 일반화 (generalization), 집단화 (aggregation) 개념 등이 바로 그것이다. 또한 데이터 계층 사이의 상속 관계를 유지하고자 ER 데이터 모델에 객체 지향 개념을 도입한 경우도 나타났다.

그러나 확장된 ER 데이터 모델들도 문제점을 가지고 있다. 즉, 데이터의 수직적 계층 구조와 수평적 관계를 하나의 도면에 표현함으로써 수직적 계층 구조에서 나타나는 추상화의 개념을 자연스럽게 표현하고 있지 못하다는 점이다.

즉, <그림 3-7>과 같은 ER 도표가 있다고 할 때 사람 개체를 여자 개체와 남자 개체로 상세화 (specialization)할 수 있다. 이와 같은 경우에 확장된 ER 데이터 모델의 도표로 표시하면 <그림 3-8>와 같이 표현된다. 그런데 이러한 확장 ER 도표는 기본 ER 도표에서 사람 개체를 단순히 대체하는 것이 아니라 도표 상에서 그림이 커져가며, 변화하는 현상을 보이고 있다.



<그림 3-7> ER 도표의 예



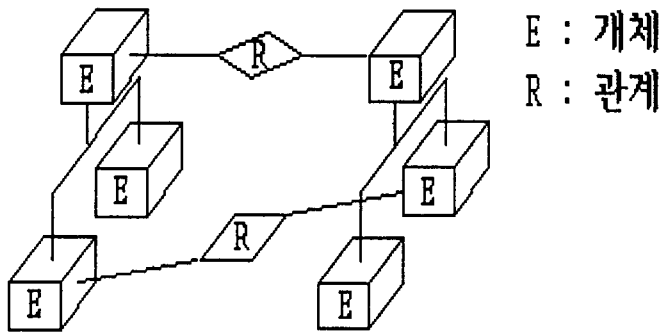
<그림 3-8> 확장된 ER 도표의 예

따라서 확장된 ER 데이터 모델은 개체를 추상화된 개체로 대체할 수 있는 개념을 올바르게 표현하기에는 미흡한 점이 있다. 즉, 추상화된 개체라면 추상화된 개체를 존재하는 개체의 위치에 그대로 대체시킬 수 있을 때 그 의미를 좀 더 확실하게 전달할 수 있을 것이다. 만약 두 가지의 데이터 특성을 다른 도면에서 표현한다면 문제점들을 해결할 수 있다.

따라서 본 논문에서는 개체의 계층과 개체들의 관계를 3 차원 상에서 나타내는 새로운 데이터 모델을 제안하며 이 모델을 Entity-Structure-Relationship 데이터 모델이라 부르하고자 한다. ESR (Entity-Structure-Relationship) 데이터 모델은 데이터의 수직적 계층 구조와 수평적 관계를 보다 효과적으로 표현하기 위하여 제안한 데이터 모델이다.

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 수직적 계층 구조와 외부적으로 나타나는 수평적 관계를 모두 표현하는 것이 좋은 데이터 모델이 될 것이다.

따라서 본 논문에서는 데이터의 수직적 계층 구조와 수평적 관계를 별도로 표시하여 나타내는 방법을 제안한다. 즉, 데이터의 수직적 계층 구조는 ES (Entity-Structure) 데이터 모델로 표현하고, 데이터의 수평적 관계는 ER (Entity-Relationship) 데이터 모델로 표현하여 데이터 모델을 3 차원으로 나타내고자 한다. ESR 데이터 모델의 개념을 그림으로 표현하면 <그림 3-9>과 같다.



<그림 3-9> ESR 데이터 모델의 개념

#### 4. 관계형 데이터 모델로의 변환

Chen[6]의 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. 또한 학문적인 관계형 데

이타 모델로의 변환보다는 실제 관계형 데이터베이스의 테이블로 선언할 수 있는 형태로 변환하는 것이 많다. 따라서 본 논문에서도 실제 관계형 데이터베이스에서의 테이블로 변환하고, SQL 문장의 형태로 직접 코딩할 수 있는 단계로 변환하는 알고리즘을 제시하였다.

#### 4.1 ER 데이터 모델의 변환

Chen[6]이 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 3.1절에서 언급한 바와 같이 4 가지로 분류하였으며 이진 관계 (Binary Relationship)인 경우의 변환 과정을 보이면 <그림 4-1>과 같다.

```

알고리즘 ER-TO-RDM

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

<그림 4-1> ER 데이터 모델의 변환

## 4.2 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 데이터로 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 과정을 적용하게 된다. 각 추상화 개념에 따른 변환 과정은 다음과 같다.

### (1) 일반화 개체

일반화 개념은 3.2절에서 언급한 바와 같이 4 가지로 분류할 수 있다. 이와 같은 4 가지의 일반화 개체에 대하여 관계형 데이터 모델로 변환하는 과정이 <그림 4-2>에 나타나 있다.

### 알고리즘 GEN-TO-RDM

```
Begin
  if disjoint and total
    subclass entity만 table로 생성;
  endif;
  if overlap and total
    superclass entity만 table로 생성;
    subclass를 표시하는 flag attribute를 생성된 table에 추가;
    subclass의 attribute들은 null값 허용;
  endif;
  if disjoint and partial
    superclass와 subclass들 모두를 table로 생성;
    subclass의 key attribute는 superclass의 key
      attribute를 상속;
  endif;
  if overlap and partial
    superclass와 subclass들 모두를 table로 생성;
    subclass의 key attribute는 superclass의 key
      attribute를 상속;
  endif;
End.
```

<그림 4-2> 일반화 개체의 변환

## (2) 집단화 개체

집단화 개념은 3.2절에서 언급된 바와 같이 8 가지로 분류할 수 있다. 이와 같은 8 가지의 집단화 개체에 대하여 관계형 데이터 모델로 변환하는 과정이 <그림 4-3>에 나타나 있다.

### 알고리즘 AGR-TO-RDM

Begin

```
if overlap and single and mandatory
  superclass entity만 table로 생성;
  subclass entity의 attribute들은 생성된 table에 추가;
endif;
if overlap and single and optional
  superclass entity만 table로 생성;
  subclass entity의 attribute들은 생성된 table에 추가;
  추가된 attribute들은 null값 허용;
endif;
if overlap and multiple and mandatory
  superclass와 subclass 모두를 table로 생성;
  subclass entity의 Key는 subclass와 superclass의 Key
  attribute들의 복합 Key로 구성;
endif;
if overlap and multiple and optional
  superclass와 subclass 모두를 table로 생성;
  subclass entity의 Key는 subclass와 superclass의 Key
  attribute들의 복합 Key로 구성;
endif;
if disjoint and single and optional
  superclass만 table로 생성;
  subclass entity의 attribute는 생성된 table에 추가;
  추가된 attribute는 null값 허용;
  subclass를 식별하는 flag attribute를 table에 추가;
endif;
if disjoint and multiple and optional
  superclass와 subclass 모두를 table로 생성;
  subclass entity의 Key는 subclass와 superclass의 Key
  attribute들의 복합 Key로 구성;
endif;
End.
```

<그림 4-3> 집단화 개체의 변환

### (3) 다중측면화 개체

다중측면화 개념은 3.2절에서 언급된 바와 같이 2 가지로 분류할 수 있다. 이와 같은 2 가지의 다중측면화 개체에 대하여 관계형 데이터 모델로 변환하는 과정이 <그림 4-4>에 나타나 있다.

```
알고리즘 ASP-TO-RDM

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

<그림 4-4> 다중측면화 개체의 변환

## 4.3 ESR 데이터 모델의 변환

ESR 데이터 모델은 3.3절에서 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. 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.

#### 4.4 시스템 통합 환경을 고려한 변환

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 환경 하에서 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 저장소를 고려하는 경우의 변환 과정은 약간의 수정이 가해져야 한다. 공유 데이터 저장소를 고려한 ESR 데이터 모델의 변환 과정 중 이진 관계인 경우는 <그림 4-5>와 같다.

#### 알고리즘 ESR-TO-RDM

Begin

(SDR의 ES 데이터 모델 변환):

지역 DB의 ES 데이터 모델 변환;

if 두 entity가 모두 SDR에 존재하는 경우

    관계에 해당하는 table 생성;

elseif 한 entity는 SDR에 존재하고, 다른 entity는 지역 DB에 존재하는 경우

    if SDR의 entity가 multiple인 경우

        관계에 해당하는 table 생성;

    elseif SDR의 entity가 mandatory이고 single인 경우

        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 4-5> 공유 데이터 저장소를 고려한 ESR 데이터 모델의 변환



## 5. 데이터 모델링 도구의 설계 및 구현

### 5.1 시스템 통합 환경에서의 ESR 데이터 모델링

본 논문에서 제안하는 시스템 통합 환경에서의 ESR 데이터 모델링 방법론을 설명하면 다음과 같다. 대상이 되는 실세계의 모델은 사용자의 관점에서는 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.

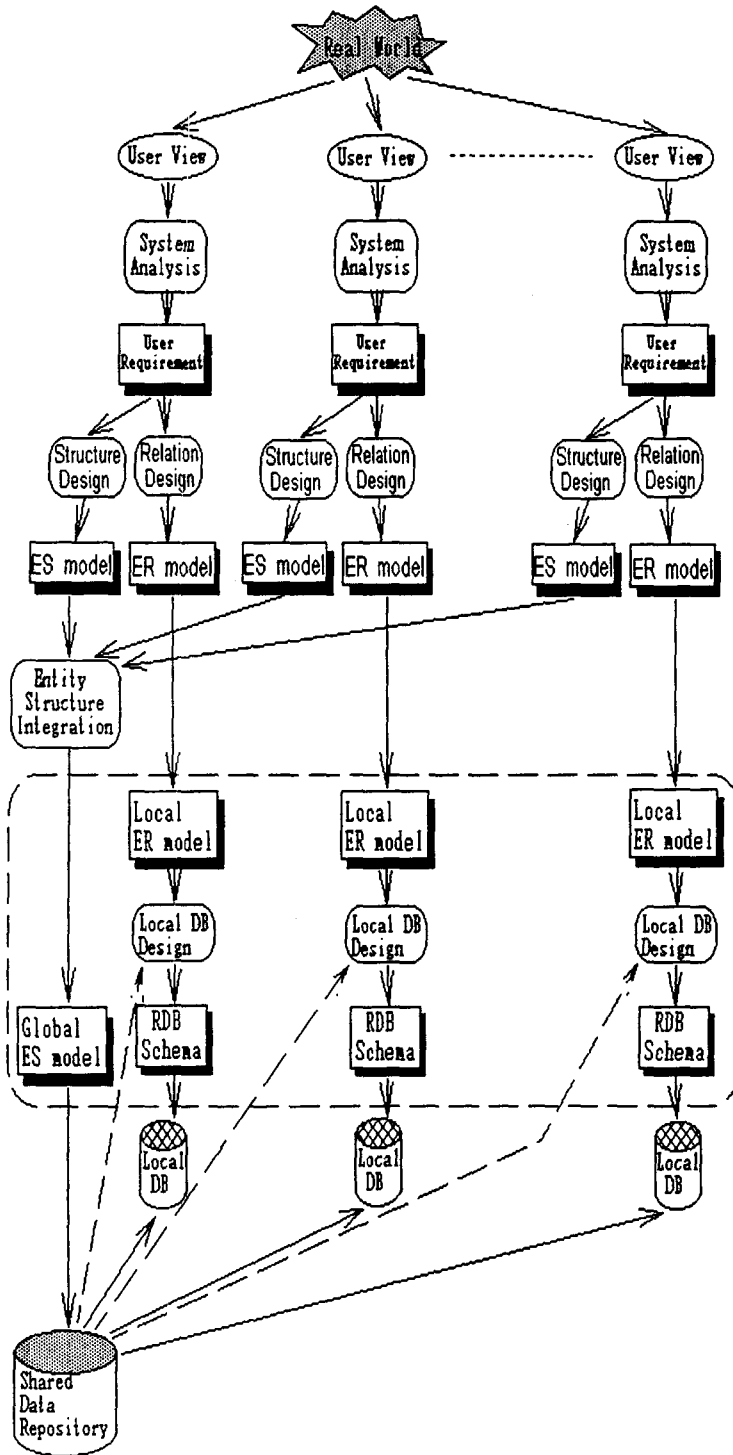
본 논문에서 제안하는 U 턴 데이터 모델링 방법론에서 공유 데이터 저장소 (SDR)내의 데이터는 ES 데이터 모델로만 표현된다. 공유 데이터 저장소내의 개체들은 주로 데이터 개체들만을 표현하고, 개체들 간의 관계는 지역 데이터베이스에서 표현한다. 따라서 지역 데이터베이스 구축은 ES 데이터 모델과 ER 데이터 모델을 함께 사용한다. 시스템 통합 환경에서의 ESR 데이터 모델링 방법론의 개념도를 <그림 5-1>에 나타내었다.

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 구축시 수행되는 데이터 모델링과 다른 점이다. 이와 같은 고려사항은 구축하고자 하는 데이터 개체를 모두 지역 데이터베이스에서 생성하는 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.

### 5.2 ESR 데이터 모델링 도구의 설계 및 구현

데이터 모델은 데이터베이스 구축시에 모델링 방향을 제시하는 역할을 한다. 즉, 데이터베이스 구축시에 발생할 수 있는 여러 선택의 길에서 하나를 선택할 수 있는 가이드 역할을 하는 것이다.



<그림 5-1> 시스템 통합 환경에서의 ESR 데이터 모델링

따라서 데이터 모델이 완전한 형태를 가지고 있으면 이러한 모델은 CASE 도구로서 구현 가능하게 되며, CASE 도구는 사람이 데이터베이스를 구축하고자 하는 경우에 많은 노력을 기울이지 않고 쉽게 데이터베이스를 구축할 수 있게 해 준다. 뿐만 아니라 CASE 도구는 사람이 데이터베이스를 구축하는 경우에 발생할 수 있는 여러 오류를 사전에 제거하여 주며, 구축된 데이터 모델 내에서의 여러 관계들의 일관성 및 무결성을 보장해 줄 수 있다.

ER 데이터 모델과 같이 실세계를 중심으로 설계되는 의미론적 데이터 모델을 구축되는 관계형 데이터 모델로 변환하는 경우를 고려하면 데이터 모델링을 위한 CASE 도구의 필요성은 더욱 절실해진다. 이는 ER 데이터 모델에서의 각 경우에 따라 관계형 데이터 모델에서의 서로 다른 테이블의 형태로 나타날 수 있기 때문이다.

특히 ESR 데이터 모델은 ER 데이터 모델에서의 개체 추상화 개념을 독립적인 데이터 모델로 전환시키면서 데이터 모델내의 경우의 수가 매우 많아지게 되며, SDR 내의 공유 데이터를 함께 고려하면서 모델링을 하기 때문에 사람이 수작업으로 하기에는 많은 어려움이 따르게 된다. 따라서 ESR 데이터 모델을 이용하여 데이터베이스를 구축하기 위해서 CASE 도구의 개발은 반드시 필요하다.

ESR 데이터 모델링 도구는 시스템 통합 환경을 고려하여 설계되었기 때문에 공유 데이터 저장소 (SDR)의 내용을 참조하면서 데이터베이스를 구축할 수 있다. ESR 데이터 모델링 도구의 메뉴는 <그림 5-2>와 같다.

ESR 데이터 모델링 도구의 주 메뉴는 File, Edit, Translation, Output 등 4 개로 이루어진다. File 메뉴는 존재하는 파일을 열고, 닫는 역할을 한다. 본 데이터 모델링 도구에서는 ES 데이터 모델을 저장하고 있는 파일의 확장자는 ESM, ER 데이터 모델을 저장하고 있는 데이터 파일의 확장자는 ERM으로 하고 있으며, 특히 공유 데이터 저장소의 ES 데이터 모델을 저장하고 있는 데이터 파일은 확장자를 SDR로 하고 있다. 또한 변환된 테이블의 구조를 저장하고 있는 데이터 파일은 TBL, 변환된 SQL 문장을 저장하고 있는 데이터 파일은 SQL이라는 확장자를 가지고 있다.

Edit 메뉴는 ES 데이터 모델 및 ER 데이터 모델을 작성하기 위한 메뉴로 항목을 선정하면 그 항목에 해당하는 그림이 화면에 나타나며, 위치는 마우스로 선정할 수 있다. Edit 메뉴에서 Entity는 개체를 나타내고, Relationship은 관계를 나타낸다. Generalization은 일반화를, Aggregation은 집단화를, Multiple Aspect는 다중측면화를 표시한다. Link 항목은 각 개체들과 관계를 연결하는 데 사용한다.

Translation은 ESR 데이터 모델을 관계형 데이터 모델로 변환하는 메뉴이다. 여러 상용 RDBMS에서 사용하는 문장들로 변환하는 것이 목표이지만 본 시스템에서는 시각적으로 쉽게 파악할 수 있는 형태인 테이블 형태와 표준 SQL의 형태로 변환할 수 있도록 하였다.

Output 메뉴는 ES 데이터 모델, ER 데이터 모델, 변환된 테이블과 SQL을 출력할 수 있도록 하는 메뉴이다.

## ESR 데이터 모델링 도구의 메뉴

1. File
  - New
  - Open
  - Save
  - Close
  - Quit
2. Edit
  - Entity
  - Relationship
  - Generalization
  - Aggregation
  - Multiple Aspect
  - Link
3. Translation
  - Table
  - SQL
4. Output
  - ES model
  - ER model
  - Table
  - SQL

<그림 5-2> ESR 데이터 모델링 도구의 메뉴

## 6. 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 정보 관리 시스템 또한 단독 시스템에서 독립적으로 정보를 관리하는 것이 아니라 시스템 통합 환경 하에서 다른 정보 관리 시스템과 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 모델링하는 3차원적인 특성을 가지는 ESR 데이터 모델을 제안하고, 시스템 통합 환경 하에서 ESR 데이터 모델로 설계된 모델을 관계형 데이터 모델로 변환하는 방법을 제시하였다. 또한 ESR 데이터 모델을 바탕으로 하는 데이터 모델링 도구를 설계하였으며, 구현시에 고려하여야 할 점에 대하여 논하였다.

본 논문에서 제안하는 데이터 모델링 도구는 현재 프로토타입을 개발하고 있는 중이며, 앞으로 시스템 통합 환경 하에서 데이터 모델링을 하는 경우에 많은 도움을 주게 될 것이다. 앞으로 ESR 데이터 모델의 완전성을 위하여 형식론을 개발하고, ESR 데이터 모델을 사용하여 실제의 데이터베이스를 구축할 예정이다.

## 7. 참고문헌

- [1] James P. Davis, Ronald D. Bonnell, "EDICT - An Enhanced Relational Data Dictionary : Architecture and Example", Proceedings Fourth International Conference on Data Engineering, 1988, pp. 184-191.
- [2] Cheng Hsu et al., "Information Resources Management in Heterogeneous Distributed Environments : A Metadatabase Approach", IEEE Trans. sw eng., June 1991.
- [3] Larry P. English, "DRM Today, DRM Tomorrow", Database Programming & Design, November 1993, pp. 54-57.
- [4] Ali Hazzah, "Data Dictionaries : Paths to a Standard", Database Programming & Design, August, 1989, pp. 26-32.
- [5] Victor M. Markowitz, Arie Shoshani, "Representing Extended Entity-Relationship Structures in Relational Databases : A Modular Approach", ACM TODS, Vol. 17, No. 3, September 1992, pp. 423-464.
- [6] Peter Pin-Shan Chen, "The Entity-Relationship Model - Toward a Unified View of Data", ACM TODS, Vol. 1, No. 1, March 1976, pp. 9-36.
- [7] Toby J. Teorey, Dongqing Yang, James P. Fry, "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", ACM Computing Surveys, Vol. 18, No. 2, June 1986, pp. 197-222.
- [8] Batini, Ceri, Navathe, "Conceptual Database Design - An Entity-Relationship Approach", Benjamin/Cummings Publishing Company, Inc., 1992.
- [9] Elmasri, Navathe, "Fundamentals of Database Systems", Benjamin/Cummings Publishing Company, Inc., 1989.
- [10] Terry Moriarty, "Are You Ready for a Repository?", Database Programming & Design, March 1990, pp. 64-71.
- [11] James Martin, "Information Engineering I", Prentice-Hall, 1994.
- [12] Mohan Prabandham, William J. Selfridge, Douglas D. Mann, "A View of the IRDS Reference Model", Database Programming & Design, March 1990, pp. 42-53.

- [13] InKee Jeong, YoonJin Chang, DooKwon Baik, "A Temporal Database for Data Code Based Information Management", Proceedings of 1st ICCTA, 1994, pp. 111-116.
- [14] 김창화, "EA 모델에 의한 지식 표현 모델링", 고려대학교 대학원 박사학위 논문, 1989. 12.
- [15] 조동영, "다중 측면을 고려한 객체 중심 데이터베이스 모델링", 고려대학교 대학원 박사학위 논문, 1991. 12.
- [16] 정인기, 백두권, "효율적인 정보관리를 위한 데이터 코드 관리 시스템의 설계", 한국경영과학회/대한산업공학회 춘계공동학술대회 발표논문집, 1993, pp. 464-473.
- [17] 정인기, 백두권, "효율적인 정보관리를 위한 데이터코드베이스의 설계 및 구현", 한국정보과학회 가을학술발표논문집, Vol. 20, No. 2, 1993, pp. 214-217.