

# ERD시소러스를 이용한 뷰 통합 방법론

이 원 조<sup>†</sup> · 고 재 진<sup>\*\*</sup> · 장 길 상<sup>\*\*\*</sup>

## 요 약

본 논문에서는 데이터베이스 설계시 중요한 과정인 개념설계 단계에서 개체관계도(Entity Relationship Diagram)의 정보를 저장하는 ERD시소러스(Thesaurus)를 구축하고, 이러한 ERD시소러스를 기반으로 하는 뷰 통합 방법론을 제시하고자 한다. 제시된 방법론의 유용성을 입증하기 위하여, 적용사례에 대한 뷰 통합지원시스템의 프로토타입을 구축하였다. 적용결과, ERD시소러스 기반의 방법론이 기존의 뷰 통합 방법론보다 친밀도 분석, 의미충돌 해결, 뷰 통합과정에서 더 효과적임을 확인할 수 있었다. 따라서 이 방법론이 기존의 단편화된 스키마의 통합이나 대규모 데이터베이스 통합 설계시 유용하게 활용될 수 있을 것으로 기대된다.

## A Methodology for View Integration Using ERD Thesaurus

Won Jo Lee<sup>†</sup> · Jae Jin Koh<sup>\*\*</sup> · Gil Sang Jang<sup>\*\*\*</sup>

### ABSTRACT

This paper constructs ERD thesaurus that is storing information about Entity Relationship Diagram(ERD), and proposes an ERD thesaurus-based methodology for view integration in an important conceptual design step in designing databases. To show the usefulness of proposed methodology, the prototype for view integration support system is implemented for the applied case. As a result, ERD thesaurus-based methodology is more effective than the existing methodologies for view integration in the aspects of affinity analysis, semantic conflicts resolution, and view integration processes. Therefore, our methodology is expected to be utilized in integrating the existing fragmented schema or designing a large database integration.

**키워드**: 개체관계도(Entity Relationship Diagram), 시소러스(Thesaurus), 뷰 통합(View Integration), 친밀도 분석(Affinity Analysis), 의미충돌 분석(Semantic Conflicts Resolution)

### 1. 서 론

지금까지 기업이나 많은 조직에서 구축된 기존의 정보시스템들은 대부분 여러 조직 및 지역에 분산되어 개별적으로 구축되어 있거나, 규모가 방대하고, 진부화되어 이들의 통합을 위한 재구축에 직면해 많은 시간과 비용을 지불하고 있다. 즉, 기업에서 각 업무별로 구축된 정보시스템들은 대부분 별도의 데이터베이스를 운영하고 있는데, 이러한 데이터베이스에는 여러 가지 문제점을 내포하고 있다. 첫 번째는 업무 변화에 따른 데이터베이스의 확장성 및 연계성의 부족으로 인한 타 시스템과의 연계가 어렵고, 두 번째로 사업 부문별, 적용 업무별로 산재한 데이터베이스들의 정보에 대한 일치성과 관리가 어려우며, 세 번째는 데이터베이스 관리시스템(DBMS : Database management system)간

의 호환성 부족으로 명확한 시스템 통합이 곤란하고, 네 번째는 구축시점이 오래된 진부한 시스템이라는 것이다[2, 10, 13]. 이러한 문제점을 해소하기 위해서 기업들은 정보 전략의 일환으로 기업 통합 데이터 모델의 구축을 추진하고 있다.

기업의 통합 데이터 모델을 구축하기 위해서 우선적으로 기업의 단위 업무 영역에서 요구되는 개별 데이터 모델을 구축한 다음 각각의 개별 데이터 모델을 합병하여 하나의 통합 데이터 모델을 구축해야 한다. 이러한 데이터 모델의 구축은 단위 업무별로 구축하는 경우와 전체 통합모델을 구축하는 경우로 나눌 수 있는데, 이 두 가지 경우 모두 설계자의 역량부족이나 정보공유의 부재, 설계시점의 차이로 다음과 같은 여러 가지 문제를 발생하게 된다. 첫 번째는 다른 데이터 모델간에 동일한 데이터를 중복하여 표현할 수 있고, 두 번째는 동일한 데이터에 서로 다른 명칭을 명명할 수 있으며, 세 번째는 상이한 데이터에 대해 동일 명칭을 명명할 수 있다는 것이다. 이와 같은 문제점들은 데이

<sup>†</sup> 정 회 원 : 울산과학기술대학교 컴퓨터정보학부 교수

<sup>\*\*</sup> 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수

<sup>\*\*\*</sup> 정 회 원 : 울산대학교 경영학부 경영정보학 교수

논문접수 : 2003년 9월 1일, 심사완료 : 2004년 3월 10일

터의 정합성과 신뢰성을 저하시키는 중요한 원인이 되고 있다. 이러한 현실의 문제점들을 해결하여 하나의 통합 데이터 모델을 구축하는 과정이 뷰 통합인데, 통합 데이터 모델의 구축에서 매우 중요한 과정의 하나로 필수적인 과정이다[10, 14]. 개념적 데이터 모델링 방법은 피터 첸(Peter Chen)에 의해 1976년에 제안된 개체관계 모델(ERM: Entity Relationship Model)을 주로 사용하고 있는데, 이 모델이 지니고 있는 단순성 때문에 현재 광범위한 응용 분야에서 사용되고 있다.

본 연구의 주제인 뷰 통합은 논리적인 데이터베이스 설계시 설계 단계에서 하나의 데이터베이스에 대한 여러 개의 뷰를 완전한 데이터베이스에 대한 개념적인 스키마를 형성하기 위해 통합하는 것인데, 뷰는 데이터베이스의 응용 뷰 또는 외부 뷰를 나타낸다. 뷰 통합은 여러 단계에서 수행될 수 있지만 주로 개념설계 단계에서 수행된다. 뷰 통합의 목적은 독립적으로 생성된 여러 응용 뷰들에 대한 통합 스키마를 생성하는 것이다. 데이터베이스 통합은 분산 데이터베이스 환경에서 여러 데이터베이스들의 전역 스키마를 생성하는데, 전역 스키마는 통합된 모든 데이터베이스의 가상 뷰가 된다[15]. 통합 데이터베이스에서 스키마 통합은 존재하는 데이터베이스 스키마들이 확장된 또는 통합된 개념적인 뷰를 제공하는 하나의 전역 스키마를 형성하기 위해 나타났다.

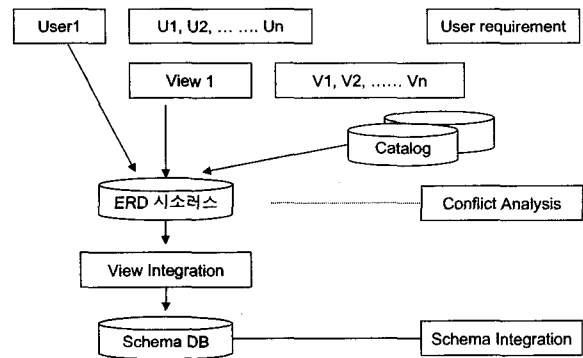
〈표 1〉 뷰 통합을 위한 기존연구

저자	통합 방법
S. Castano 외 3명(1998)	-개념 스키마 분석 기법(디스크립터들과 콘텐츠들을 추출, 유사도(Similarity)를 평가), 레퍼런스 컴포넌트들의 생성을 위한 분석 방법론(도메인의 주요 개념들에 대한 표준화된 설명을 제공, 이전의 개발 단계들을 재사용)
S. B. Navathe 외 2명(1996)	-ERM의 확장인 개체 범주 관계 모델을 기반으로 통합 방법(사전 통합, 유사 개체 통합, 서로 다른 개체 통합, 관계 통합)
C. Batini 외 2명(1986)	-ERM을 이용한 뷰 통합과 스키마 통합 방법론(사전 통합(Preintegration), 뷰 비교(View Comparison), 뷰 적합(View Conforming) 그리고 합병(Merging) 및 재구조화(Restructuring))
C. Batini 외 2명(1992)	-확장 개체관계 모델을 이용한 통합 방법론(명칭 충돌과 구조적 충돌 분석, 공통 개념과 통합화)
신기태 외 3명(1994)	-개체 관계 모델로 작성된 데이터 모델을 대상으로 작성된 개별 뷰를 통합 개념 뷰로 통합하는 뷰 통합 절차(뷰 통합 전처리 과정, 개체들의 통합, 관계들의 통합, 중복관계)
김기중(1998)	-통합 데이터베이스 시스템 고찰(시스템, 스키마 구조), 스키마 통합(선 통합, 스키마 비교, 정합, 병합 및 재구성)
임병학(1996)	-의미 객체 모델(Semantic Object Model: SOM)을 이용한 뷰 통합 방법(의미 객체 모델을 이용한 뷰 통합 방법론을 정립, 뷰간 친밀도 분석을 통한 뷰간 통합순서 결정)

데이터베이스의 통합 기법에 대한 많은 이전 연구들이 있는데, 주요 관련 연구 논문들을 요약하면 다음 <표 1>과 같다. 이러한 연구들에서 사용한 뷰 통합 기법은 방법론 제시 위주이거나, 뷰 통합에 이용된 정보가 파일 형태로 설계되어 있어 대규모 데이터베이스 통합에는 적용하기 어려운 측면이 있었다. 본 논문에서는 이러한 문제점을 개선하기 위하여 ERD시소러스 개념[25]을 도입하여 뷰 통합에 필요한 주요 정보들을 구조화하고, 이를 이용한 뷰 통합의 유용성과 가능성을 검증하고자 하였다. 따라서 본 논문에서는 개체관계도 정보를 저장하는 ERD시소러스를 구축하고, 이를 기반으로 한 뷰 통합 방법론을 제시하고자 한다.

## 2. ERD시소러스 구축

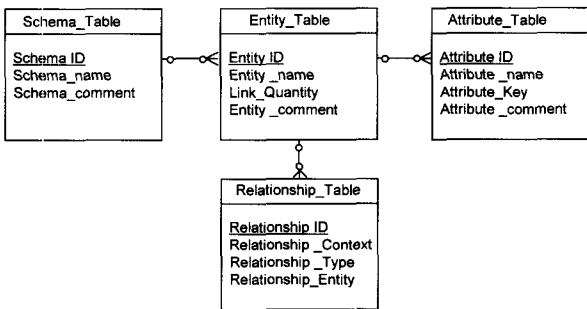
ERD시소러스 구축은 개념적 설계단계에서 작성된 개체관계도(ERD)와 기존 정보시스템 데이터베이스 카탈로그(Database Catalog)로부터 뷰 통합에 필요한 정보를 추출하여 ERD시소러스를 구축하였다. ERD시소러스 구축의 목적은 시소러스를 이용하여 뷰 통합을 위한 정보를 효율적으로 추출하기 위한 것이다[1-3]. 따라서 뷰 친밀도 분석과 의미 충돌 해결은 뷰 통합을 위한 사전단계이다. 그러므로 ERD시소러스의 구축은 뷰 통합에 필요한 정보의 저장과 효율적인 검색이 가능하도록 구조화해서 설계해야 한다[7, 19, 21]. 다음 (그림 1)은 ERD시소러스의 활용 개념도를 나타낸다.



(그림 1) ERD시소러스의 활용 개념도

뷰들 간의 통합 순서를 결정하기 위해서 뷰 간의 친밀도 분석을 수행하는데, 친밀도 값이 가장 큰 뷰들을 우선 충돌 해결을 통하여 통합한다. 따라서 부분 통합이 완료되면 다른 뷰와 친밀도 분석을 수행하여 충돌 해결과 통합을 반복하여 통합이 완료될 때까지 진행한다. 따라서 ERD시소러스 설계에도 친밀도 분석과 충돌 해결에 필요한 뷰명, 개체명, 속성명, 관계, 카디널리티(Cardinality) 등의 정보가 포함되어야 한다. 따라서 ERD시소러스의 최상위 단계인 1단계는

ERD와 카탈로그로부터 추출된 뷰 정보의 분류를 위해서 뷰(Schema) 테이블과 2단계는 각 뷰의 하위 구성 요소인 개체명(Entity) 테이블과 관계(Relationship) 테이블, 그리고 최종 단계인 3단계에서는 속성명(Attribute) 테이블로 설계하였는데, ERD시소러스의 데이터베이스 모델링은 CASE 툴(PowerDesigner)을 사용하였다. (그림 2)는 각각 CASE 툴로 모델링한 뷰 친밀도 분석을 위한 ERD시소러스의 개념적 모델링을 나타낸다.

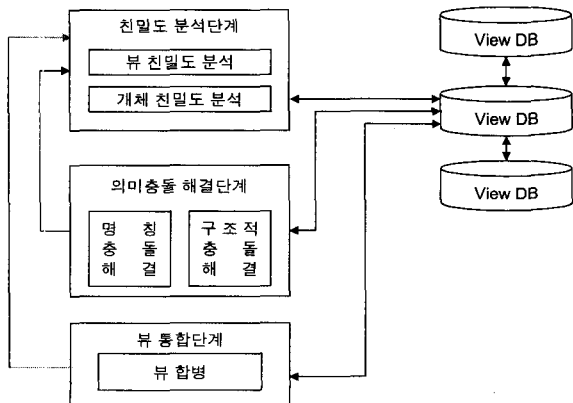


(그림 2) ERD시소러스 개념적 모델링

### 3. 뷰 통합 방법론

#### 3.1 뷰 통합 절차

뷰 친밀도 분석은 뷰의 효율적인 통합을 위한 방법론인데, 뷰 친밀도 분석을 위해서는 뷰에 포함된 개체와 속성들의 명칭과 특성(Property), 관계 등을 통하여 친밀도 분석을 한다. 일반적인 뷰 통합의 접근 방법을 보면 이진 통합을 사용하는데, 예를 들면, 뷰1과 뷰2에 대한 충돌을 분석하여 충돌 리스트를 작성하고, 여기서 발견된 충돌을 해결한 후 뷰 간의 성질을 참조하여 뷰 합병을 수행한다. 이러한 과정의 반복을 통해서 통합 뷰를 완성하게 된다. 다음(그림 3)은 ERD시소러스를 이용하여 친밀도 분석을 통한 뷰 통합절차를 나타낸다.



(그림 3) ERD시소러스를 이용한 뷰 통합 절차

#### 3.2 뷰 친밀도 분석

뷰 친밀도 분석의 목적은 모든 대상 뷰들간의 친밀도를 계산하여, 뷰간의 의미충돌 해결과 통합순서를 결정하는 기본자료로 활용한다. 따라서 뷰 친밀도 분석은 두개의 뷰 간에 공통 유사정보를 얼마나 많이 포함하고 있는가를 나타내는 수치인데, 뷰 친밀도를 계산하기 위해서는 뷰를 개체들의 집합으로 보고, 선형대수의 벡터이론을 적용하여 계산할 수 있다. 뷰의 통합순서 결정은 뷰 간의 친밀도가 가장 높은 값을 갖는 것으로부터 뷰 비교 및 통합을 수행하는 전략을 수립한다. 뷰 친밀도 분석절차는 개체관계 집합의 결정과 개체관계 집합의 벡터변환, 친밀도 계산, 통합 뷰의 결정 순으로 진행된다. 개체관계 집합 결정절차에서 뷰는 단위업무 영역을 개체관계도에 의해서 표현한 데이터 모델인데, 이 데이터 모델은 의미개체들로 구성되어 있다. 따라서 뷰는 의미개체들이 모인 집합으로 볼 수 있다. 의미개체를  $E_i (i = 1, 2, 3, \dots, n)$ , 전체 개체관계 집합을  $S_0$ 라고 할 때, 다음과 같이 집합으로 표현할 수 있다.

$$S_0 = \{E_1, E_2, E_3, \dots, E_n\}$$

또 뷰가  $m$ 개이고, 각 뷰는  $p$ 개 이내의 의미개체를 포함하고 있는 경우, 각 뷰를 의미개체 집합으로 표현하면 다음과 같다.

$$S_{0i} = \{E_{i1}, E_{i2}, E_{i3}, \dots, E_{ip}\}, i = 1, 2, 3, \dots, m$$

뷰는 개체관계의 집합으로 표현되는 스칼라 값을 가지는 벡터로 표현할 수 있는데, 통합 하고자하는 뷰가  $m$ 개이고, 그 뷰에 속해 있는 의미개체가  $n$ 개일 때, 전체 의미개체 집합을 각 뷰에 대해 벡터로 표현할 경우,

뷰 벡터를  $V_i (i = 1, 2, 3, \dots, m)$ , 개체관계의 스칼라 값을  $a_{ik} (k = 1, 2, 3, \dots, n)$ 라 할때, 의미개체 집합의 벡터변환은 다음과 같이 표현될 수 있다.

$$V_i = (a_{i1}, a_{i2}, \dots, a_{in}), i = 1, 2, 3, \dots, m$$

$$a_{ik} = \begin{cases} 0, & E_k \notin S_{0i} \\ 1, & E_k \in S_{0i} \end{cases}, i = 1, 2, 3, \dots, n,$$

즉, 스칼라값  $a_{ik}$  는 뷰  $V_i$ 에 있는 의미개체  $E_k$ 의 값이다.  $a_{ik}$ 의 값은 의미개체  $E_k$ 가 뷰  $S_{0i}$ 에 존재 할 때는 1의 값을, 존재하지 않을 때는 0의 값을 갖는다.

뷰 친밀도 값 계산은 뷰간에 동일 의미개체가 어느 정도 포함되어 있는가를 나타내는 것으로 스칼라곱(Scalar Multiplication)의 내적(Inner Product)을 이용하여 두 뷰간의 친

밀도 값을 구할 수 있다. 친밀도 값이 크면 클수록 뷰간 동일 의미개체의 수가 많음을 의미한다. 두 뷰 벡터를 각각  $V_i, V_j$ , 두 뷰간의 친밀도를  $A_{ij}$  라고 할때, 친밀도 값을 계산하는 식은 다음과 같다.

$$A_{ij} = V_i \cdot V_j = \sum_{k=1}^n aikajk$$

위의 수식에 의해 계산된 친밀도 값에 따라 높은 값을 갖는 뷰 쌍에 대해서 먼저 의미충돌 해결 및 통합을 통해 제1 부분통합 뷰를 생성한다. 이 통합 뷰와 나머지 뷰들과의 친밀도 값을 다시 계산하여 그 값이 가장 높은 값을 갖는 뷰쌍에 대해서 의미충돌 해결과 통합을 하게된다. 이와 같은 과정을 모든 뷰가 통합될 때까지 반복 수행하여 최종 통합 스키마를 얻을 수 있다[5, 14].

### 3.3 개체 친밀도 분석

개체 친밀도 분석에서는 뷰와 뷰 간에 존재하는 개체 간의 비교순서 및 통합순서를 결정한다. 개체 간 친밀도 계산은 뷰 친밀도 계산과 유사한 방법에 의해서 구할 수 있다. 개체 친밀도 분석은 뷰 친밀도 결정을 위한 심층단계인데, 뷰의 유사함 결정을 지원하기 위한 방법으로 수행한다. 개체 친밀도 분석순서는 뷰간 속성집합 결정, 속성집합의 벡터변환, 친밀도 계산, 통합개체 결정과 같은 순서를 통해서 수행된다. 뷰가 개체들의 집합으로 표현이 가능하듯이 개체도 집합으로 표현이 가능하다. 뷰 친밀도 분석에서 두 뷰를  $V_i, V_j$  라고 하면 전체 속성들의 수가  $m_a$  일 때, 전체 속성 집합  $S_a$  는 다음과 같이 표현할 수 있다.

$$S_a = \{att_1, att_2, att_3, \dots, att_{m_a}\}$$

각각의 뷰벡트  $V_i(i=1,2,3,\dots,m)$ 의 개체  $E_k(k=1,2,3,\dots,p)$ 에 속해 있는 속성집합  $S_{ik}$  는 다음과 같이 표현될 수 있는데, 이때 각 개체가 갖는 속성의 수는 최대  $n_k$  개이다.

$$S_{ik} = \{att_{ik1}, att_{ik2}, att_{ik3}, \dots, att_{ikn_k}\}$$

각각의 개체에 해당하는 속성들의 집합은 스칼라 값을 갖는 벡터로 변환이 가능하고, 뷰  $V_i$ 의 개체  $E_k$ 에서 속성이 갖는 값들을  $a_{ik}$  이라고 할 때, 속성벡터  $B_{ik}$  는 다음과 같이 표현할 수 있다.

$$V_i = (a_{i1}, a_{i2}, \dots, a_{im}), i = 1, 2, 3, \dots, m$$

$$a_{ik} = \begin{cases} 0, & att_i \notin S_{ik} \\ 1, & att_i \in S_{ik} \end{cases}, i = 1, 2, 3, \dots, m_a,$$

즉  $a_{ik}$ 의 값은 전체 속성집합  $S_a$ 에 있는 속성이 각 뷰의

개체 속성집합  $S_{aik}$ 에 속해 있으면 1의 값을, 속해있지 않으면 0의 값을 갖는다. 개체 친밀도는 두 개체 간에 동일 속성의 보유 정도를 나타내는데, 개체 친밀도의 계산은 뷰 친밀도 계산과 같이 두 속성집합 벡터의 내적에 의해 구할 수 있다. 뷰의 친밀도가 가장 높은 두 뷰가  $V_i, V_j$ 이고,  $V_i$ 에 있는 개체명칭은  $k$ , 뷰  $V_j$ 에 있는 개체명칭  $k'$ , 두 뷰간에 공통으로 갖는 개체들의 친밀도를  $A_{ij}^a$ 라 할 때 일반적인 계산식은 다음과 같다.

$$A_{ija} = B_{ik} \cdot B_{jk'} = \sum_{i=1}^{n_a} aikajk' i$$

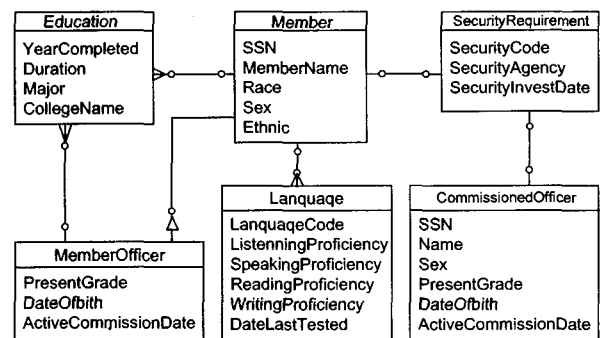
통합 개체의 결정은 개체 친밀도 값이 가장 큰 개체가 우선 의미충돌 해결 및 통합을 하게되고, 다음 순위의 친밀도 값을 갖는 개체가 차례로 의미충돌 해결 및 통합을 계속 수행한다[5, 13, 14].

### 4. 적용 사례

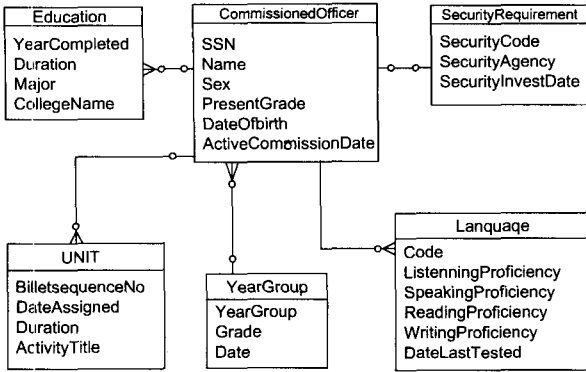
ERD시소러스를 이용한 뷰 통합의 유용성을 입증하기 위해서 군사정보시스템의 3개 뷰 모델[2, 14]을 사용하여 사례 연구를 수행한다.

- 뷰1 : IMAPMIS(Inactive Manpower and Personal Management Information System)
- 뷰2 : OPIS(Officer Personal Information System)
- 뷰3 : ADMID(Active Duty Military Inventory Database)

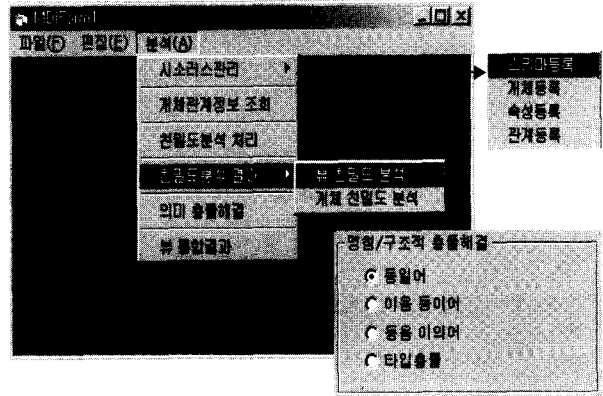
사례에서 3개의 개체관계도로부터 뷰 정보를 추출하여 시소러스에 저장하는 방법으로 ERD시소러스를 구축한다. 다음 (그림 4), (그림 5), (그림 6)은 각각 뷰1, 뷰2, 뷰3의 개념적 모델을 나타낸다.



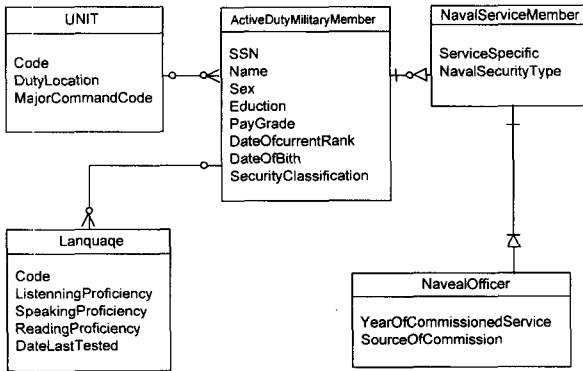
(그림 4) 뷰1 : IMAPMIS 개념적 모델



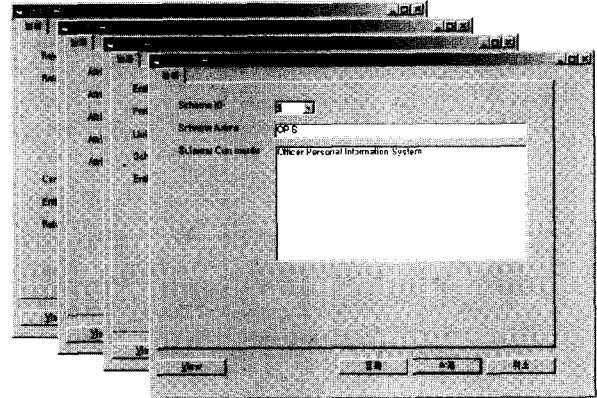
(그림 5) 뷰2 : OPIS 개념적 모델



(그림 8) ERD시소러스 시스템의 초기화면



(그림 6) 뷰3 : ADMID 개념적 모델



(그림 9) ERD시소러스 시스템의 자료등록 화면

다음 (그림 7)은 뷰에 대한 ERD시소러스의 정보를 질의어로 검색한 결과를 나타낸다.

SCHEMA_ID	ENTITY_NAME	ATTRIBUTE_NAME	TO_SCHEMA
1	Language	ReadingProficiency	2
1	Language	WritingProficiency	2
1	Language	DateLastTested	2
1	CommissionedOfficer	SSN	2
1	CommissionedOfficer	Name	2
1	CommissionedOfficer	Sex	2
1	CommissionedOfficer	PresentGrade	2
1	CommissionedOfficer	DateOfBirth	2
1	CommissionedOfficer	ActiveCommissionDate	2
2	Education	YearCompleted	1
2	Education	Duration	1
2	Education	Major	1
2	Education	CollegeName	1
2	CommissionedOfficer	SSN	1
2	CommissionedOfficer	Name	1
2	CommissionedOfficer	Sex	1
2	CommissionedOfficer	PresentGrade	1
2	CommissionedOfficer	DateOfBirth	1
2	CommissionedOfficer	DateOfBirth	1
2	CommissionedOfficer	ActiveCommissionDate	1

(그림 8) ERD시소러스의 뷰 정보

4.1 ERD시소러스 시스템 구현

사례연구의 뷰 모델과 제안된 방법론을 적용한 ERD시소러스의 효율적인 자료 관리와 방법론의 응용을 위해 ERD시소러스 시스템을 구현하였다. 이 시스템은 오라클(ORACLE 9i)과 비주얼 베이직(MS Visual Basic 6.0)으로 개발하였는데, (그림 8)은 ERD시소러스 시스템의 초기 화면이고, (그림 9)는 자료 등록 화면이다.

4.2 뷰 친밀도 분석사례

뷰 친밀도 분석은 ERD시소러스 시스템으로 구현된 사례의 3개 뷰 개체관계도 간의 친밀도 값을 계산하여 뷰 간의 합병 순서를 결정하는 과정인데, 각 뷰 간 유사 개체의 포함 갯수로 표현된다. 예를 들면, 각 뷰에 대한 벡터 변환은 <표 2>와 같이 나타낼 수 있다. 그리고 (그림 10)은 ERD시소러스로부터 3개의 뷰 간에 친밀도 값을 계산한 결과이며, 다음은 개체 집합을 나타낸다.

$$S_0 = \{ActiveDutyMilitaryMember, CommissionedOfficer, Education, Language, Member, MemberOfficer, NavalOfficer, NavalServiceMember, SecurityRequirement, UNIT, YearGroup\}$$

$$S_{01} = \{Education, Member, SecurityRequirement, MemberOfficer, Language, CommissionedOfficer\}$$

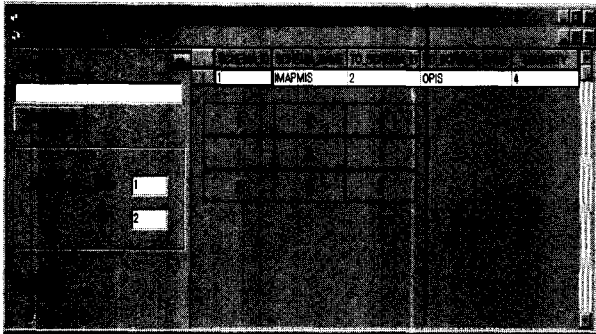
$$S_{02} = \{Education, CommissionedOfficer, SecurityRequirement, YearGroup, UNIT, Language\}$$

$$S_{03} = \{UNIT, ActiveDutyMilitaryMember, NavalServiceMember, Language, NavalOfficer\}$$

〈표 2〉 뷰간 벡터 변환 표

구분	1	2	3	4	5	6	7	8	9	0	1
V1	0	1	1	1	1	1	0	0	1	0	0
V2	0	1	1	1	0	0	0	0	1	1	1
V3	1	0	0	1	0	0	1	1	0	1	0

$A_{12} = 4, A_{13} = 1, A_{23} = 2$



(그림 10) 뷰 친밀도 분석

사례의 결과와 같이 뷰1과 뷰2는 4, 뷰1과 뷰3은 1, 뷰2와 뷰3은 2로 3개의 뷰 중 뷰1과 뷰2 간의 친밀도 값이 4로 가장 높게 나타났다. 그러므로 뷰1과 뷰2를 우선통합 대상으로 결정한다. 따라서 뷰 친밀도 값을 ERD시소러스 시스템을 이용하여 간단하게 추출할 수 있다.

4.3 개체 친밀도 분석사례

개체 친밀도 분석의 목적은 ERD시소러스에서 추출된 뷰 친밀도 분석을 근거로 뷰에 포함된 개체들의 친밀도를 검증하기 위한 방안으로 수행한다. 예를 들면, 뷰1과 뷰2에 대한 벡터 변환은 <표 3>과 같이 나타낼 수 있다. 따라서 ERD시소러스로부터 뷰 간에 친밀도 값이 가장 높은 뷰1과 뷰2에 포함된 개체 친밀도 값을 계산한 결과 (그림 11)과 같이 개체 친밀도 값을 ERD시소러스 시스템을 이용하여 간단하게 추출할 수 있었으며, 다음은 속성 집합을 나타낸다.

$S_{a12} = \{ActiveCommissionDate, ActivityTitle, BillSequenceNo, Code, CollegeName, Date, DateAssigned, DateLastTested, DateOfBirth, Duration, Ethnic, Grade, LanguageCode, ListenningProficiency, Major, MemberName, Name, PresentGrade, Race, ReadingProficiency, SSN, SecurityAgency, SecurityCode, SecurityInvestDate, Sex, SpeakingProficiency, WritingProficiency, YearCompleted, YearGroupNo\}$

$S_{a1, 101} = \{CollegeName, Duration, Major, YearCompleted\}$

$S_{a1, 102} = \{Ethnic, MemberName, Race, SSN, Sex\}$

$S_{a1, 103} = \{SecurityAgency, SecurityCode, SecurityInv-$

$estDate\}$

$S_{a1, 104} = \{ActiveCommissionDate, DateOfBirth, PresentGrade\}$

$S_{a1, 105} = \{DateLastTested, LanguageCode, ListenningProficiency, ReadingProficiency, SpeakingProficiency, WritingProficiency\}$

$S_{a1, 106} = \{ActiveCommissionDate, DateOfBirth, Name, PresentGrade, SSN, Sex\}$

$S_{a2, 201} = \{CollegeName, Duration, Major, YearCompleted\}$

$S_{a2, 202} = \{ActiveCommissionDate, DateOfBirth, Name, PresentGrade, SSN, Sex\}$

$S_{a2, 203} = \{SecurityAgency, SecurityCode, SecurityInvestDate\}$

$S_{a2, 204} = \{Date, Grade, YearGroupNo\}$

$S_{a2, 205} = \{ActivityTitle, BillSequenceNo, DateAssigned, Duration\}$

$S_{a2, 206} = \{Code, DateLastTested, ListenningProficiency, ReadingProficiency, SpeakingProficiency, WritingProficiency\}$

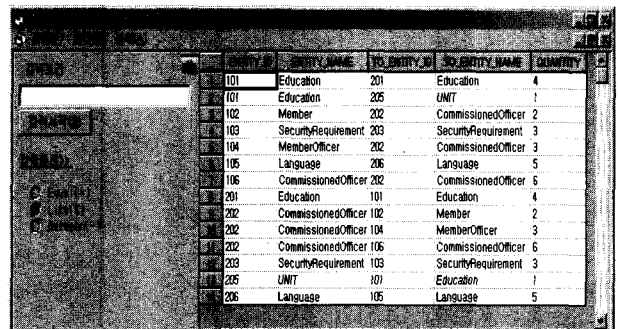
〈표 3〉 뷰1과 뷰2간 벡터 변환 표

구분	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
B <sub>1, 101</sub>	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
B <sub>1, 102</sub>	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0
B <sub>1, 103</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
B <sub>1, 104</sub>	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
B <sub>1, 105</sub>	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1
B <sub>1, 106</sub>	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0
B <sub>2, 201</sub>	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
B <sub>2, 202</sub>	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0
B <sub>2, 203</sub>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
B <sub>2, 204</sub>	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
B <sub>2, 205</sub>	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
B <sub>2, 206</sub>	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1

$A_{12, 101, 202} = 4, A_{12, 101, 205} = 1, A_{12, 102, 202} = 2,$

$A_{12, 103, 203} = 3, A_{12, 104, 202} = 3, A_{12, 105, 206} = 5,$

$A_{12, 106, 202} = 6$



(그림 11) 개체 친밀도 분석

사례의 결과와 같이 개체 친밀도 분석은 개체간에 동일한 명칭의 속성을 얼마나 많이 포함하고 있는지를 대상의 모든 개체에 대하여 한 쌍씩 계산한 것인데, 개체 “*CommissionedOfficer*”, “*Language*”, “*Education*”, “*SecurityRequirement*”를 먼저 의미충돌 해결을 한 다음 차례로 “*MemberOfficer*”와 “*CommissionedOfficer*” 그리고 “*Member*”와 “*CommissionedOfficer*” 개체에 대해서 의미충돌을 해결하고, ERD 시소러스에 저장한다.

### 4.3.1 명칭충돌(Name Conflict)

① 동일 개체명칭과 개념들이 유사한 특성과 관계(동일어)

SCHEMA_NAME	ENTITY_ID	ENTITY_NAME	ATTRIBUTE_NAME
IMAPMIS	106	CommissionedOfficer	
OPIS	202	CommissionedOfficer	
IMAPMIS	101	Education	
OPIS	201	Education	
IMAPMIS	105	Language	
OPIS	206	Language	
IMAPMIS	102	Member	
IMAPMIS	104	MemberOfficer	
IMAPMIS	103	SecurityRequirement	
OPIS	203	SecurityRequirement	
OPIS	205	UNIT	
OPIS	204	YearGroup	

(그림 12) ERD시소러스를 이용한 동일어 추출

(그림 12)의 동일어 사례에서 개체명칭이 동일한 개체는 “*CommissionedOfficer*”, “*Education*”, “*Language*”, “*Language*”, “*SecurityRequirement*”로 추출되었다.

② 다른 개체명칭과 개념들이 유사한 특성과 관계(이음동이어)

ENTITY_ID	ENTITY_NAME	ATTRIBUTE_NAME
102	Member	SSN
102	Member	MemberName
102	Member	Sex
102	Member	Race
102	Member	Ethnic
104	MemberOfficer	PresentGrade
104	MemberOfficer	DateOfBirth
104	MemberOfficer	ActiveCommissionDate
106	CommissionedOfficer	SSN
106	CommissionedOfficer	Name
106	CommissionedOfficer	Sex
106	CommissionedOfficer	PresentGrade
106	CommissionedOfficer	DateOfBirth

(그림 13) ERD시소러스를 이용한 이음동이어 추출

(그림 13)의 이음동이어 사례에서 “*MemberOfficer*”의 속성은 “*CommissionedOfficer*”의 부분 집합이다. 따라서 이 두 개의 개체명을 “*NavalOfficer*”로 재명명하고, “*Mem-*

*ber*”와는 서로 개체 관계도 상에서 포함관계이므로 “*NavalOfficer*”를 “*CommissionedOfficer*”의 하위 개체가 되도록 하여 통합 결정한다.

③ 동일 개체명칭과 개념들이 다른 특성과 관계(동음이의어)

ENTITY_ID	ENTITY_NAME	ATTRIBUTE_NAME
205	UNIT	BillSequenceNo
205	UNIT	DateAssigned
205	UNIT	Duration
205	UNIT	ActivityTitle
301	UNIT	Code
301	UNIT	DutyLocation

(그림 14) ERD시소러스를 이용한 동음어 추출

[그림 14]의 동음이의어 사례에서 뷰2와 뷰3의 개체명 “*UNIT*”은 명칭은 동일하나, 속성이 서로 달라 동음이의어로 판정하고, 뷰2의 개체명을 “*DutyStationBilletAssignment*”로 재명명하여 충돌을 해결한다.

### 4.3.2 구조적 충돌(Structural Conflict)

구조적 충돌은 타입충돌(Type Conflict)과 카디날리티 충돌(Cardinality Conflict)이 있는데, 사례의 (그림 15)와 같이 “*Education*” 개체는 뷰1과 뷰2에 동일어로 존재한다. 따라서 동일 개체명칭과 개념들이 유사한 특성과 관계를 갖는 예로 명칭충돌에서 언급되었다. 그러나 뷰3의 개체 “*ActiveDutyMilitaryMember*”의 속성이 “*Education*”으로 타입충돌이 발견되었다.

ENTITY_ID	ENTITY_NAME	ATTRIBUTE_NAME
101	Education	302 Education
201	Education	302 Education

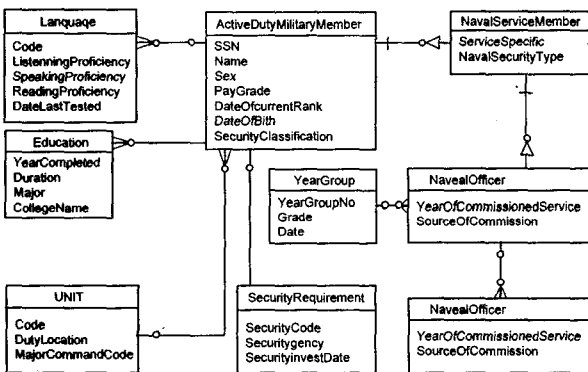
(그림 15) 뷰 간 타입충돌 발견과 해결

(그림 15)의 사례에서 발견된 타입충돌의 해결방법은

“ActiveDutyMilitaryMember” 개체에서 “Education” 속성을 제거하고, “ActiveDutyMilitaryMember” 개체와 “Education” 개체간 관계를 설정하여 충돌을 해결한다.

4.4 뷰 통합단계

의미충돌 해결단계에서 발견된 명칭충돌과 구조적 충돌을 해결하는 방식으로 뷰 간의 친밀도 분석에 의한 통합순서에 따라 뷰 간의 통합을 수행하여 전체 통합 뷰를 만든다. 이러한 과정은 개체와 개체, 개체와 속성간의 명칭충돌 해결방법을 통하여 동일어와 이음동어어 그리고 동음이의어를 찾아 개체관계의 명칭충돌을 해결하고, 구조적 충돌분석을 통하여 충돌을 해결한다. 따라서 ERD시소러스 시스템을 이용하면 뷰 통합을 위한 다양한 정보의 추출이 가능해 뷰 통합에 적용하면 유용성이 매우 높다. 따라서 ERD시소러스 시스템을 이용한 사례와 같이 다음 (그림 16)은 뷰 친밀도 분석과 의미충돌 해결을 통해 통합된 전체 뷰의 개체관계도이고, (그림 17)은 ERD시소러스에 저장된 통합 뷰 정보를 보여준다[2, 7, 14].



(그림 16) 전체 통합 뷰의 개념적 모델

# MPMS	402 ActiveDutyMilitaryMember	4207 SecurityClassification	
# MPMS	403 NavalServiceMember	4301 ServiceSpecific	PK
# MPMS	403 NavalServiceMember	4302 NavalSecurityType	
# MPMS	404 Education	4401 YearCompleted	PK
# MPMS	404 Education	4402 Duration	
# MPMS	404 Education	4403 Major	
# MPMS	404 Education	4404 CollegeName	
# MPMS	405 YearGroup	4501 YearGroupNo	
# MPMS	405 YearGroup	4502 Grade	
# MPMS	405 YearGroup	4503 Date	
# MPMS	406 NavalOfficer	4601 YearOfCommissionedService	
# MPMS	406 NavalOfficer	4602 SourceOfCommission	
# MPMS	407 UNIT	4701 Code	PK
# MPMS	407 UNIT	4702 DutyLocation	
# MPMS	407 UNIT	4703 MajorCommandCode	
# MPMS	408 SecurityRequirement	4801 SecurityCode	PK
# MPMS	408 SecurityRequirement	4802 SecurityAgency	
# MPMS	408 SecurityRequirement	4803 SecurityInvestDate	
# MPMS	409 DutyStationBilletAssignment	4901 BilletSequenceNo	
# MPMS	409 DutyStationBilletAssignment	4902 DateAssigned	

(그림 17) ERD시소러스의 통합 스키마 정보

4.5 적용 방법론 평가

대규모 기업 집단간의 합병이나, 진부화된 데이터베이스 재구축을 위한 효율적인 통합지원 자동화시스템 개발에 설계자들의 관심이 매우 높다. 그러나 이전의 많은 연구들에서 대부분 사용한 뷰 통합 기법은 방법론 제시 위주이거나, 뷰 통합에 이용된 정보가 과일형태나 단순 데이터베이스 형태로 작성되어 있어 대규모 데이터베이스 통합에는 적용하기 어려웠다. 따라서 본 연구는 이러한 사용자들의 요구를 해소하기 위해 ERD 시소러스를 구축하여 뷰 통합에 응용하는 방안을 제시하기 위해 시도되었다. 본 방법론의 특징은 뷰 통합에 필요한 스키마 정보들을 구조화하여 ERD 시소러스를 구축하고, 이를 적용하여 뷰 통합지원을 위한 프로토타입 시스템을 구축하였다.

본 방법론에서는 뷰 통합에 필요한 정보가 ERD 시소러스에 구조적 관련성을 갖고 저장되어 있어 다양한 정보의 추출이 가능해 뷰 친밀도 분석과 의미충돌 해결을 위한 정보의 추출이 용이하다. 따라서 효율적인 본 시스템의 적용을 위해서는 설계자가 임의로 부여하게되는 객체들의 명칭 표준화가 전제되어야 한다. 그러므로 뷰 통합자동화 연구 이전에 스키마 설계시 발생하는 명명법 표준화에 대한 연구가 선행되어야 하며, 이러한 명명법 표준화시 기존의 상용 시소러스(Roget)를 이용하면 개체명칭과 속성명칭 표준화가 용이하리라 사료된다. 그러나 이러한 전제 없는 ERD 시소러스 구축은 무의미하며, 완전한 뷰 통합에 대한 많은 노력에도 불구하고 자동화는 매우 어려운 과제이다.

5. 결론 및 향후 연구

본 연구에서는 데이터베이스 설계시 뷰 통합을 위하여 ERD시소러스 기반의 뷰 통합 방법론을 제안하였다. 본 방법론의 세부 적용단계로는 친밀도 분석단계, 의미충돌 해결 단계 그리고 뷰 통합단계로 구성되어 있다. 본 방법론의 유용성을 입증하기 위하여, 적용사례 및 뷰 통합지원을 위한 프로토타입 시스템을 구축하였다. 그 결과, ERD시소러스 기반의 본 방법론이 실제 통합 데이터베이스 설계시 유용하게 적용될 수 있을 것으로 판단된다. 그러나 이전에도 뷰 통합에 대한 자동화 연구가 다양하게 시도되었으나, 설계자가 개체와 속성명칭 명명시 표준화된 규칙을 따르지 않을 경우 뷰 통합의 자동화 시도는 무의미할 것으로 추정된다. 따라서, 향후 연구로는 대규모 데이터베이스 설계와 통합을 위하여 ERD시소러스를 이용한 스키마 설계의 표준안을 정



의하고, 이 규칙에 따라 스키마를 설계할 수 있는 방안을 제시하는 것이 필요할 것으로 판단된다. 또한 기존의 상용 시소러스(Roget)를 데이터베이스 설계시 개체와 속성명칭의 명명법 표준화에 이용할 수 있는 연구도 필요할 것으로 판단되며, 현재 이 분야에 대한 연구가 진행 중이기 때문에 이에 대한 연구결과를 가까운 시일 내에 발표할 수 있을 것으로 기대한다.

## 참 고 문 헌

- [1] S. Castano, V. D. Antonellis, M. G. Fugini and B. Pernici, "Conceptual Schema Analysis : Techniques and Applications," ACM Transactions on Database System, Vol.23, No.3, pp.286-333, September, 1998.
- [2] M. Kamel, Identifying, "Classifying and Resolving Semantic Conflicts in Distributed Heterogeneous Databases : Case Study," Journal of Database Management, Vol.6, No.1, pp.20-32, May, 1994.
- [3] C. Batini, M. Lenzerini, S. B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration," ACM Computing Surveys, Vol.18, No.4, December, 1986.
- [4] C. Batini, M. Lenzerini, "A Methodology for Data Schema Integration in the Entity Relationship Model," IEEE Transaction on software engineering, Vol.SE-10, No.6, Nov., 1984.
- [5] M. S. Bazara, J. J. Jarvis, "Linear Programming and Network Flows, John Wiley & Sons," 1977.
- [6] S. B. Navathe, R. Elmasri, J. Larson, "Integration User Views in Database Design," IEEE Computer, Jan, 1986.
- [7] C. Batini, S. Ceri, S. B. Navathe, "Conceptual Database Design : An Entity-Relationship Approach", Benjamin Cummings, pp.119-133, 1992.
- [8] M. W. Bright, A. R. Hurson and S. Pakzad, "Automated Resolution of Semantic Heterogeneity in Multidatabases," ACM Transaction on Database System, Vol.19, No.2, pp.212-253, June, 1994.
- [9] S. B. Navathe, & G. Gadqil, "A Methodology for View Integration in Logical Database Design," Proceedings of the International Conference on Very Large Data Bases, Mexico City, pp.142-164, September, 1982.
- [10] Y. Dupont, "Resolving Fragmentation Conflicts in Schema Integration," Switzerland.
- [11] R. Jakobovits, "Integrating Autonomous Heterogeneous Information Sources," University of Washington, July, 1997.
- [12] G. Preuner, S. Conrad, "View Integration of Life-Cycles in Object-Oriented Design," 1999.
- [13] 이희석, 임병학, 김영삼, 홍의기, "의미객체 모델을 이용한 뷰 통합 지원시스템 개발", 한국경영정보학회 학술대회 논문집, 1996.
- [14] 임병학, "의미객체 모델을 이용한 뷰통합 지원시스템 개발", 한국과학기술원 석사학위논문, 1996.
- [15] 김기중, "통합 데이터베이스를 위한 스키마 통합 방법", 정보통신학회 논문지, Vol.6, No.2, August, 1998.
- [16] 김홍수, "참조 스키마 생성을 위한 개념적 스키마 분석", 한국 OA학회 논문지, 2002.
- [17] 신기태 외3명, "제조 데이터베이스 설계에서의 뷰 통합 방안", 한국경영정보학회 논문지, 제11권 제3호, pp.668-670, 1994.
- [18] 이재윤, "동적 시소러스의 구축에 관한 실험적 연구", 연세대학교 석사학위 논문, 1994.
- [19] 이재윤, 김태수, "WordNet과 시소러스", 언어정보연찬회 발표논문집, 1998.
- [20] 김영택 외 공저, "자연언어처리", 생능출판사, pp.132-177, 2001.
- [21] 전학병, "시소러스 작성법", 산업기술정보원, 1991.
- [22] 이원조 외3명, "스키마 통합을 위한 시소러스의 활용 방안", 한국정보과학회 춘계학술대회, 제30권 제1호, pp.668-670, 2003.
- [23] 이원조 외1명, "서브스키마 통합을 위한 의미객체 시소러스의 활용", 울산과학대학 연구 논문집, 제30권 제1호, pp.183-189, 2003.
- [24] 이원조 외1명, "사전 코퍼스에서 추출한 통계적 의미정보에 기반한 개선된 의미 중의성 분별모델", 울산과학대학 연구 논문집, 제30권 제1호, pp.175-181, 2003.
- [25] <http://www.helloec.net/network/thesaurus.htm>.



## 이 원 조

e-mail : wjlee@mail.ulsan-c.ac.kr  
 1988년 울산대학교 산업공학과 공학사  
 1998년 울산대학교 정보통신대학원 정보통신공학과 공학석사  
 2002년 울산대학교 대학원 컴퓨터정보통신공학부 박사과정 수료

1981년~1999년 한일이화(주) 정보화팀 차장  
 1999년~현재 울산과학대학 컴퓨터정보학부 전임강사  
 관심분야 : SI, POP, DB분석 및 설계, ERP, e-Business



고 재 진

e-mail : jjkoh@mail.ulsan.ac.kr

1972년 서울대학교 응용수학과 공학사

1981년 서울대학교 대학원 계산통계학과  
이학석사

1990년 서울대학교 대학원 컴퓨터공학과  
공학박사

1975년~1979년 한국후지쯔(주) 기술개발부 사원

1979년~현재 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : DB시스템, 전문가 시스템, DB설계, ERP



장 길 상

e-mail : gsjang@mail.ulsan.ac.kr

1986년 울산대학교 산업공학과 공학사

1988년 한국과학기술원 산업공학과 공학  
석사

1997년 한국과학기술원 경영정보공학과  
공학박사

1995년~2000년 한국오라클(주) Senior Technical Specialist

2000년~2002년 동국대학교 산업정보학과 조교수

2002년~현재 울산대학교 경영학부 경영정보학전공 조교수

관심분야 : DB응용, ERP, e-Business, DW, 생산정보시스템, 객  
체지향 분석설계