공통 유사 서브스키마 추출을 통한 개념적 스키마 통합 : 다중 데이터베이스 시스템 적용사례

고 재 진[†] · 이 원 조[††]

요 약

최근 글로벌 기업들은 조직들이 지역적으로 분산되어 있고, 분산된 조직들로 인하여 정보시스템들도 분산되어 있다. 이러한 정보시스템의 중심이 되는 데이터베이스도 분산되어 있어, 다양한 사용자 응용 프로그램을 위한 공통된 뷰(view)의 제공 및 효과적인 관리가 어렵다. 이것을 해결하기 위한 개념이 MDBS이고, 이것을 효과적으로 구축하기 위한 방안이 스키마 통합이다. 본 논문에서는 공통 유사 서브스키마 추출을 통한 스키마 통합 방법론을 제시한다. 본 방법론은 분석 대상 스키마에 대하여 친밀도 분석, 유사 서브스키마 추출, 통합순서 결정, 의미충돌 해결, 그리고 스키마 통합 순서로 구성되어 있다. 방법론의 유용성을 검증하기 위하여, MDBS를 대상으로 사례분석을 수행하였다. 분석 결과, 본 논문의 방법론이 공통 유사 서브스키마의 추출 및 스키마 통합에 유용하게 적용될 수 있다는 것을 확인할 수 있었다.

A Conceptual Schema Integration through Extraction of Common Similar Subschemas : An Case Study of Multidatabase System

Jae Jin Koh[†] · Won Jo Lee[††]

ABSTRACT

Recently, most of global enterprises have geographically distributed organizations, thus have distributed information systems which have distributed database systems. So, it is difficult for these systems to provide common views for the application programs of end users. One of solutions to solve these difficulties is an MDBS(Multidatabase System). A method to effectively implement MDBS is a schema integration. This paper proposes a methodology for a schema integration through extraction of common similar subschemas. Our methodology is consisted of 5 phases : affinity analysis, extraction of similar subschemas, decision of integration order, resolution of semantic conflict, and schema integration. To verify the usability of our methodology, a case study is implemented with an object of MDBS. As a result, our approach can effectively be applied to the extraction of common similar subschemas and schema integration.

키워드 : 유사 서브스키마(Similar Subschemas), 다중 데이터베이스 시스템(Multidatabase System), 뷰(View), 친밀도 분석(Affinity Analysis), 스키마 통합(Schema Integration)

1. 서 론

최근 기업들은 글로벌 시대에 직면하여 대규모 기업간의 합병이 빈발하고 있어 이들 기업의 원활한 관리를 위해 분산된 기존 정보시스템들의 효율적인 통합화 문제가 중요한 화두로 부각되고 있다. 분산된 정보시스템들의 통합에 필연적으로 수반되는 중요한 요소가 데이터베이스 통합(database integration)이다. 그 이유는 여러 지역에 분산된 데이터베이스(distributed database)와 다양한 응용 프로그램들(applications)을 통한 사용자의 요구를 충족시킬 수 있는 공통된 뷰(view)의 제공과 관리의 효율성 문제이다. 따라서 데이터베이스 통합은 다양한 이용자들을 지원하는 응용 프로그램의 액세스 요청에 대한 접근성(accessibility) 및 데이터 무결성(data integrity), 그리고 관리성(manageability)이 향상될 수 있어야 하며, 단 기간 내에 최소의 비용으로 추진되어야 한다. 이러한 명제를 해결하기 위해서는 MDBS (multidatabase system)를 구축해야 하며, MDBS를 효과적으로 구축하기 위한 방안이 친밀도 분석(affinity analysis)을 통한 공통 유사 서브스키마 추출(common similar subschema extraction)이다[1, 16]. 이를 통하여 MDB 서버간 스키마 통합(schema integration)을 용이하게 지원하는 것이

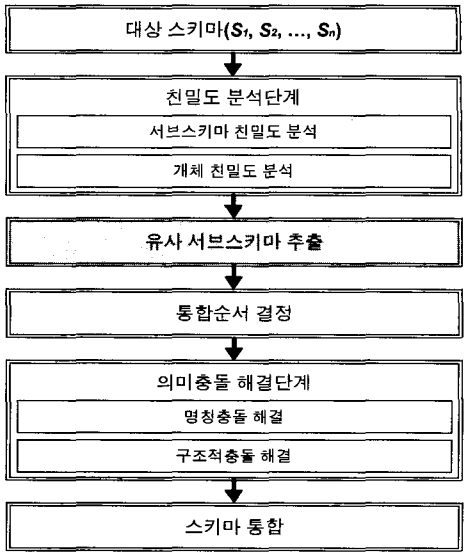
※ 본 연구는 2002학년도 울산대학교 학술연구 지원으로 수행되었음.
† 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수
†† 정 회 원 : 울산과학대학 컴퓨터정보학부 교수
논문접수 : 2004년 2월 23일, 심사완료 : 2004년 5월 3일

다[7, 8, 14]. 따라서 본 논문에서는 공통 유사 서브스키마 추출을 통한 스키마 통합 5단계 방법론을 제시한다.

본 방법론은 분석 대상 스키마에 대하여 친밀도 분석(affinity analysis), 유사 서브스키마 추출(extraction of similar subschemas), 통합순서 결정(decision of integration order), 의미충돌 해결(resolution of semantic conflict), 그리고 스키마 통합(schema integration) 순으로 구성되어 있는데, 본 방법론의 유용성을 검증하기 위하여, MDBS 구축을 위한 군인사관리시스템 ERD(Entity Relational Diagram)로 사례 연구를 수행하였다. 여기서 서브스키마는 스키마의 부분 집합(sub set)으로 하나의 개체(entity) 또는 그 이상의 개체 그룹(entity group)으로 이루어진다. 그러므로 서브스키마 친밀도 분석(subschema affinity analysis)과 개체 친밀도 분석(entity affinity analysis)을 통해서 유사 서브스키마들을 추출하고, 이들 개념적 스키마를 통합한 MDBS를 구축한다[14, 15].

2. 스키마 분석 방법론

데이터베이스 설계단계에서 개념 스키마 모델링(modeling)에는 스키마 통합 방법과 속성(attribute) 합성 방법이 있는데, 스키마 통합은 응용 문제나 사용자 그룹을 기초로 각 부문별 서브스키마를 식별하여 모델링하고, 완성된 서브스키마들을 통합하여 하나의 전체적인 스키마로 만드는 것이며, 속성 합성은 속성들을 식별해서 분류하고, 속성들간의 관계를 파악해서 개체 생성, 관계 생성을 통해 전체적인 스키마를 생성하는 방법인데, 설계단계에서 스키마 통합은 여러 단계에서 수행될 수 있지만 주로 개념적 설계 단계에서 수행된다[2, 3, 5, 15]. 이러한 스키마 통합의 목적은 실세계의 동일 부분을 나타내는 개념적 입력 스키마를 찾아서 표현을 통일하고, MDBS 구축을 위한 스키마 통합에 목적이 있는데, 스키마 통합에는 설계자간 서로 다른 관점, 모순된 설계 명세, 모델에서 구성요소 사이의 동치성, 동일 개념의 다른 표현 충돌 등의 문제점을 효율적으로 해소하는데 있다[6-9]. 여기서 스키마 통합의 의미는 물리적 관점에서의 통합을 의미하는 것이 아니라, 분산되어 있는 DBMS에서 이질 또는 동질의 스키마들에 대해 전역적으로 접근할 수 있는 논리적 구조의 서브스키마를 생성함으로써, 통합된 서브스키마에 대해 사용자들이 일관성 있게 접근할 수 있는 인터페이스를 제공해 주는 논리적 통합을 의미한다[3, 12, 14]. 대상 서브스키마 친밀도 분석과 개체 친밀도 분석을 통한 유사 서브스키마의 추출, 스키마 합병 등을 통해서 MDBS를 생성하게 된다. MDBS를 효과적으로 구축하기 위한 방안이 공통 유사 서브스키마의 추출을 통한 스키마 통합이고, 이를 기반으로 한 데이터베이스 통합인데, 데이터베이스 통합시 가장 기본적인 단계인 유사 서브스키마 추출을 통한 스키마 통합 5단계 과정에 대한 방법론을 제시한다. 다음 (그림 1)은 제안된 스키마 통합 과정이다.



(그림 1) 스키마 통합과정

따라서, 스키마 통합 과정을 설명하면 다음과 같다. 대상 스키마(S_1, S_2, …, S_n)는 지역적으로 분산된 서버에 존재하는 스키마들 중에 통합 고려대상 서브스키마이다. 첫 번째 단계, 친밀도 분석단계는 유사 서브스키마 추출을 위해서 서브스키마간의 친밀도를 계산하는 단계인데, 서브스키마 친밀도 분석과 개체 친밀도 분석으로 이루어진다. 두 번째 단계, 유사 서브스키마 추출은 친밀도 분석을 통해서 계산된 친밀도 값이 가장 높은 서브스키마 쌍들을 찾아내는 단계이다. 세 번째 단계, 통합순서 결정단계는 서브스키마간 또는 개체간의 비교 순서 및 통합 순서를 결정한다. 네 번째, 의미충돌 해결단계는 통합대상 서브스키마에 존재하는 개체와 속성들의 명칭충돌과 구조적 충돌을 해결하는 단계이다. 다섯 번째, 서브스키마 통합은 서브스키마간 개별 개체의 동일 개념을 표현하고 있는 모든 개체와 속성들을 찾아 통합하는 단계이다.

2.1 서브스키마 친밀도 분석(subschema affinity analysis)

서브스키마 친밀도 분석의 목적은 모든 대상 서브스키마들간의 친밀도를 계산하여, 서브스키마간의 의미충돌 해결과 통합 순서를 결정하는 기본 자료로 활용한다. 따라서 서브스키마 친밀도 분석은 두 개의 서브스키마간에 공통 유사 정보를 얼마나 많이 포함하고 있는가를 나타내는 수치인데, 서브스키마간의 친밀도를 계산하기 위해서는 서브스키마를 개체들의 집합으로 보고, 선형대수의 벡터이론을 적용하여 계산할 수 있다. 스키마의 통합 순서 결정은 서브스키마간의 친밀도가 가장 높은 값을 갖는 것으로부터 스키마 비교 및 통합을 수행하는 전략을 수립한다. 친밀도 분석 절차는 개체관계 집합의 결정과 개체관계 집합의 벡터변환, 친밀도 계산, 통합 스키마의 결정 순으로 진행된다. 개체관계 집합 결정에서 서브스키마는 단위업무 영역을 개체 관

계도에 의해서 표현한 데이터 모델인데, 이 데이터 모델은 의미 개체들로 구성되어 있다. 따라서 서브스키마는 의미 개체들이 모인 집합으로 볼 수 있다. 의미개체를 $E_i(i = 1, 2, 3, \cdots, n)$, 전체 개체관계 집합을 F_e 라고 할 때, 다음과 같이 집합으로 표현할 수 있다.

$$F_e = \{ E_1, E_2, E_3, \cdots, E_n \}$$

또 서브스키마가 m 개이고, 각 서브스키마는 p개 이내의 의미개체를 포함하고 있는 경우, 각 서브스키마를 의미개체 집합으로 표현하면 다음과 같다.

$$F_{ei} = \{E_{i1}, E_{i2}, E_{i3}, \cdots, E_{ip}\},\ i = 1, 2, 3, \cdots, m$$

서브스키마는 개체관계의 집합으로 표현되는 스칼라 값을 가지는 벡터로 표현할 수 있는데, 통합 하고자 하는 서브스키마가 m 개이고, 그 서브스키마에 속해 있는 의미개체가 n 개일 때, 전체 의미개체 집합을 각 서브스키마에 대해 벡터로 표현할 경우,

서브스키마 벡터를 $S_i\ (i = 1, 2, 3, \cdots, m)$, 개체관계의 스칼라 값을 $a_{ik}\ (k = 1, 2, 3, \cdots, n)$ 라 할 때, 의미개체 집합의 벡터변환은 다음과 같이 표현될 수 있다.

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

$$a_{ik} = \begin{cases} 1, Ek \in Fei \\ 0, Ek \notin Fei \end{cases},\ i = 1, 2, 3, \cdots, n$$

즉, 스칼라 값 a_{ik} 는 서브스키마 S_i 에 있는 의미개체 E_k 의 값이다. a_{ik} 의 값은 의미개체 E_k 가 서브스키마 Fei에 존재할 때는 1의 값을, 존재하지 않을 때는 0의 값을 갖는다.

서브스키마 친밀도 값 계산은 서브스키마간에 동일 의미개체가 어느 정도 포함되어 있는가를 나타내는 것으로 스칼라곱(scalar multiplication)의 내적(inner Product)을 이용하여 두 서브스키마간의 친밀도 값을 구할 수 있다. 친밀도 값이 크면 클수록 서브스키마간 동일 의미개체의 수가 많음을 의미한다. 두 서브스키마 벡터를 각각 S_i, S_j, 두 서브스키마간의 친밀도를 A_{ij} 라고 할 때, 친밀도 값을 계산하는 식은 다음과 같다.

$$Aij = S_i \cdot S_j = \sum_{k=1}^{n} aikajk$$

위의 수식에 의해 계산된 친밀도 값에 따라 높은 값을 갖는 서브스키마 쌍에 대해서 먼저 의미충돌 해결 및 통합을 통해 제1부분통합 스키마를 생성한다. 이 통합 스키마와 나머지 서브스키마들과의 친밀도 값을 다시 계산하여 그 값이 가장 높은 값을 갖는 서브스키마 쌍에 대해서 의미충돌 해결과 통합을 하게된다. 이와 같은 과정을 모든 서브스키마가 통합될 때까지 반복 수행하여 최종 통합 스키마를 얻을 수 있다[12, 13].

2.2 개체 친밀도 분석(entity affinity analysis)

서브스키마 친밀도 분석에서는 대상 서브스키마간에 존재하는 개체간의 비교 순서 및 통합 순서를 결정한다. 개체 친밀도 분석은 서브스키마 친밀도 분석과 유사한 방법에 의해서 계산할 수 있다. 개체 친밀도 분석은 서브스키마간의 친밀도 결정을 위한 심층 단계인데, 서브스키마의 유사함 결정을 지원하기 위한 방법으로 수행한다. 개체 친밀도 분석 순서는 서브스키마간 속성 집합 결정, 속성 집합의 벡터 변환, 친밀도 계산, 통합 개체 결정의 순서를 통해서 수행된다. 서브스키마가 개체들의 집합으로 표현이 가능하듯이 개체도 속성들의 집합으로 표현이 가능하다. 따라서 서브스키마 친밀도 분석에서 분산된 MDBS 서버에 존재하는 두 서브스키마를 S_i, S_j 라고 하면 전체 속성들의 수가 m_a 일 때, 전체 속성 집합 F_a는 다음과 같이 표현할 수 있다.

$$F_a = \{ att_1, att_2, att_3, \cdots, att_{ma} \}$$

각각의 서브스키마 벡터 $V_i(i = 1, 2, 3, \cdots, m)$의 개체 $E_k(k = 1, 2, 3, \cdots, p)$에 속해 있는 속성 집합 F_{aik} 는 다음과 같이 표현될 수 있는데, 이때 각 개체가 갖는 속성의 수는 최대 n_a 개이다.

$$F_{aik} = \{ att_{ik1}, att_{ik2}, att_{ik3}, \cdots, att_{ikna} \}$$

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

$$B_{ik} = \{ a_{ik1}, a_{ik2}, a_{ik3}, \cdots, a_{ikna} \}$$

$$a_{ikl} = \begin{cases} 1, att_l \in F_{aik} \\ 0, att_l \notin F_{aik} \end{cases},\ l = 1, 2, 3, \cdots, m_a$$

즉 a_{ikl} 의 값은 전체 속성 집합 F_a 에 있는 속성이 각 서브스키마의 개체속성 집합 F_{aik} 에 속해 있으면 1의 값을, 속해 있지 않으면 0의 값을 갖는다. 개체 친밀도는 두 개체간에 동일 속성의 보유 정도를 나타내는데, 개체 친밀도의 계산은 서브스키마 친밀도 계산과 같이 두 속성 집합 벡터의 내적에 의해 구할 수 있다. 서브스키마의 친밀도가 가장 높은 두 서브스키마가 S_i, S_j 이고, S_i 에 있는 개체 명칭은 k, t 서브스키마 S_j 에 있는 개체 명칭 k', 두 서브스키마간에 공통으로 갖는 개체들의 친밀도를 A_{ij}^{a} 라 할 때 일반적인 계산식은 다음과 같다.

$$A_{ij}^{a} = B_{ik} \cdot B_{jk'} = \sum_{l=1}^{na} a_{ikj}\, a_{jk'i}$$

통합 개체의 결정은 개체 친밀도 값이 가장 큰 개체가 우선 의미충돌 해결 및 통합을 수행하게 되고, 다음 순위의 친밀도 값을 갖는 개체가 차례로 의미충돌 해결 및 통합을

수행한다[12, 13].

2.3 충돌해결(Conflict resolution)

충돌 해결에서 발견된 명칭 충돌(naming conflict)과 구조적 충돌(structural conflict)을 해결하는 방식으로 서브스키마간 친밀도 분석에 의한 통합 순서에 따라 서브스키마간 통합을 수행하여 전체 통합 스키마를 만든다. 충돌해결 단계는 친밀도 분석에서 선택된 두개의 대상 서브스키마를 의미개체 표현시 발생할 수 있는 의미 충돌을 찾아 해결하는 것이다. 충돌 분석의 목적은 개체들간의 친밀도 값을 계산하여 서브스키마의 유사성을 추출하기 위한 것인데, 명칭 충돌 분석과 구조적 충돌 분석으로 나눌 수 있다. 명칭 충돌은 실세계의 동일한 개념이 두개의 스키마에서 서로 동일한 명칭으로 표현된 동일어와 다른 명칭으로 표현된 이음동의어(synonym) 그리고 실세계의 다른 개념을 두개의 서브스키마에서 동일한 명칭으로 표현한 동음이의어(homonym)로 분류한다. 그리고 구조적 충돌은 모델링 구조나 무결성 제약을 서로 다르게 사용했을 경우에 발생하게 되는데, 구조적 충돌에는 타입 충돌(type conflict), 카디날리티 충돌(cardinality conflict)이 있다. 최종적으로 판정이 모호한 경우에는 실무자와의 협의를 거쳐서 조정하고, 설계자의 주관적인 판정에 의해 통합을 결정한다[3, 10, 12, 20].

2.4 스키마 통합 방법론 비교

본 논문에서 제안된 방법론은 분산된 서버에서 유사 서브스키마를 추출하고, 이를 통합하여 MDBS를 구축하는 것인데, 기존 논문에서는 ERM의 단편(개체)이나 단일 데이터베이스의 SOM(Semantic Object Model)을 사례로 사용하여 3단계 통합 방법론(친밀도 분석, 의미충돌 해결 단계, 뷰 통합)을 적용하였는데, 본 연구에서는 다중 데이터베이스의 ERM을 사례로 5단계 통합 방법론을 적용하였다[12]. 본 방법론의 특징은 공통 유사 서브스키마 추출을 통한 스키마 통합의 단계적인 접근법을 제시하고, 분산된 다중서버의 ERD를 통합하여 MDBS를 구축하는 사례를 보여주고 있다. 다음 <표 1>은 기존의 서브스키마 방법론과 본 방법론의 비교표이다.

〈표 1〉 스키마 통합 방법론 비교표

구 분	통합방식	데이터 모델	명 칭 충 돌	구조적충돌	통 합 절 차
S. Castano 외 3명(19980)[1]	스키마 통합	개체관계모델 (ERM)	- 시소러스사용 - 이음동의어 - 동음이의어	없 음	- 스키마 디스크립터 추출 - 스키마 추상화 - 스키마 유사성 평가 - 참조 컴포넌트 추출
B. Navathe 외 2명(1986)[2]	스키마 통합	개체범주관계모델 (ECR)	- 기 해결 가정	- 기 해결 가정	- 사전 통합 - 유사 객체 통합 - 다른 객체 통합 - 관계 통합
C. Batini 외 2명(1986)[3]	스키마 통합	개체관계모델 (ERM)	- 개체명칭충돌 ·이음동의어 ·동음이의어	- 타입 충돌 - 카디날리티 ·종속 충돌 ·키 충돌 ·행위 충돌	- 사전 통합 - 뷰 비교 - 뷰 적합 - 합병 및 재구조화
C. Batini 외 2명(1992)[4]	스키마 통합	확장개체관계모델 (EERM)	- 개체명칭충돌 ·이음동의어 ·동음이의어	- 타입 충돌 - 식별자 - 카디날리티	- 충돌분석 해결 - 합병(통합화)
이희석 외 3명(1996)[12]	뷰 통합	의미객체모델 (SOM)	- 이음동의어 - 동음이의어	- 타입충돌 - 식별자 - 카디날리티	- 친밀도 분석(개체/뷰) - 의미충돌해결 - 뷰 통합
김기중(1998)[15]	스키마 통합	확장개체관계모델 (EERM) MDBS	- 이음동의어 - 동음이의어	- 타입충돌 - 카디날리티	- 사전 통합 - 뷰 비교 - 뷰 적합 - 합병 및 재구조화
신기태 외 3명(1994)[17]	스키마 통합	개체관계모델 (ERM)	- 기 해결 가정	- 기 해결 가정	- 개체 통합 - 관계 통합
제안모델	스키마 통합	개체관계모델 (ERM) MDBS	- 동일어 - 이음동의어 - 동음이의어	- 타입충돌 - 카디날리티	- 서브스키마 친밀도 분석 - 유사 서브스키마 추출 - 통합순서 결정 - 의미충돌 해결

3. 스키마 통합 사례 연구

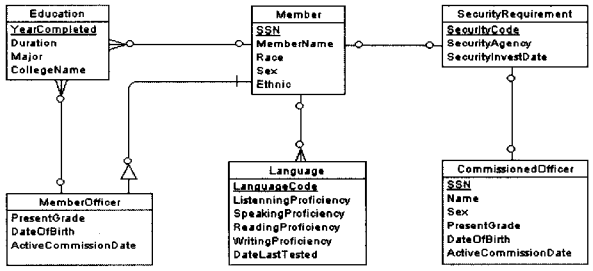
본 논문의 공통 유사 서브스키마 추출을 통한 스키마 통합 방법론의 유용성을 입증하기 위하여 물리적으로 분산된 서버에 존재하는 3개의 군인사관리시스템 ERD를 이용하여 스키마 통합 사례를 수행한다.

① 서브스키마(S_1) : A-서버 "예비군 인사관리시스템" (그림 2)

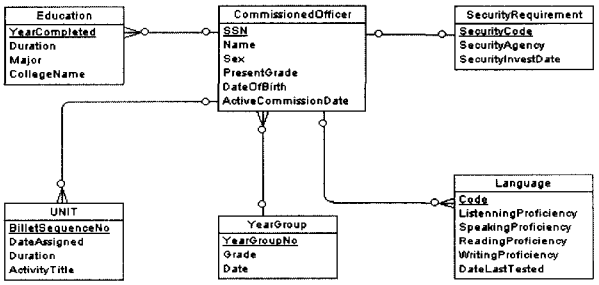
② 서브스키마(S_2) : B-서버 "장교 인사관리시스템" (그림 3)

③ 서브스키마(S_3) : C-서버 "해군 사병인사관리시스템" (그림 4)

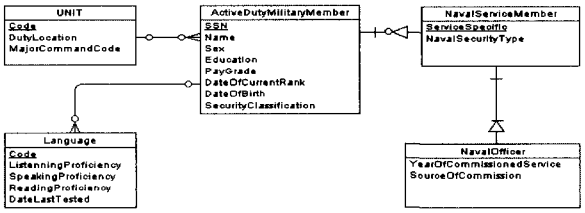
다음 (그림 2)과 (그림 3)와 (그림 4)는 CASE 툴(Power-Designer)로 설계된 분산데이터베이스 서버(A, B, C)에 존재하는 통합 대상 서브스키마 ERD를 나타낸다.



(그림 2) 서브스키마(S_1) : A-서버 "예비군 인사관리시스템"



(그림 3) 서브스키마(S_2) : B-서버 "장교 인사관리시스템"



(그림 4) 서브스키마(S_3) : C-서버 "해군사병인사관리시스템"

3.1 유사 서브스키마 추출

3.1.1 서브스키마 친밀도 분석

사례 서브스키마(S_1, S_2, S_3)에서 의미 개체 전체집합(F_e)과 각 서브스키마에 대한 의미 개체 집합(F_{ei})을 다음과 같이 각각 나타낼 수 있다.

① 임의의 서브스키마(S_1, S_2, S_3)의 전체 의미 개체집합(F_e)으로 표현된다.

S_o = {*ActiveDutyMilitaryMember, CommissionedOfficer, Education, Language, Member, MemberOfficer, NavalOfficer, NavalServiceMember, SecurityRequirement, UNIT, YearGroup*}

② 각각의 서브스키마에 속해 있는 개체의 의미 개체집합(F_{ei})으로 표현된다.

S_{o1} = {*Education, Member, SecurityRequirement, MemberOfficer, Language, CommissionedOfficer*}

S_{o2} = {*Education, CommissionedOfficer, SecurityRequirement, YearGroup, UNIT, Language*}

S_{o3} = {*UNIT, ActiveDutyMilitaryMember, NavalService-Member, Language, NavalOfficer*}

③ 개체가 갖는 값들을 a_{ik} 일 때, 서브스키마 벡터(S_i)로 나타내고, 두 서브스키마간에 동일 개체의 보유 정도를 0과 1로 나타낼 수 있다. 즉, 전체 개체집합(F_e)과 각각의 개체집합(F_{ei})의 요소들을 순차적으로 비교해서 전체 개체집합(F_e)에 존재하면 1, 존재하지 않으면 0을 표내의 해당 셀(cell)에 기록한다. 각 서브스키마간의 벡터 변환표는 <표 2>와 같다.

<표 2> 서브스키마간 벡터 변환표

구 분	1	2	3	4	5	6	7	8	9	0	1
V_1	0	1	1	1	1	1	0	0	1	0	0
V_2	0	1	1	1	0	0	0	0	1	1	1
V_3	1	0	0	1	0	0	1	1	0	1	0

④ 두 서브스키마간 친밀도 값이 가장 큰 서브스키마간에 우선 의미충돌을 해결한다. <표 2>에서 서브스키마간 친밀도 값은 <표 2>의 두 서브스키마 벡터(S_i)간에 동일 열(column)에 1이 나타나는 개수로 표현된다. 즉, 서브스키마 친밀도 값이 4이면 두 서브스키마 집합(F_{ei})간에는 동일한 의미개체가 4개 존재한다는 의미이다. 따라서 동일한 의미개체를 많이 포함하면 유사한 서브스키마로 보아 우선 통합의 고려 대상으로 결정한다. 다음은 계산된 서브스키마 친밀도 값이다.

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

서브스키마(S_1, S_2, S_3) 간의 친밀도 값을 계산한 결과 서브스키마(S_1)과 서브스키마(S_2)는 4개의 의미개체를 포함하고 있어, 두 서브스키마간 우선 통합을 수행한다. 그리고 1차 통합 서브스키마와 서브스키마(S_3)를 통합한다. 만약 여러개의 서브스키마를 통합할 경우 이러한 기법으로 통합 순서를 결정하고, 통합 과정을 반복하여 전체 스키마를 통합한다.

3.1.2 개체 친밀도 분석

개체 친밀도 분석의 목적은 대상 서브스키마에 포함된

구성 요소인 의미 개체들의 친밀도 계산을 통해서 유사 의미 개체를 추출하여 통합하기 위한 것이다. 사례와 같이 서브스키마(S_1)과 서브스키마(S_2) 간에 우선 통합을 수행한다. 따라서 개체들의 친밀도를 계산하기 위해서 각 개체간에 동일 또는 유사 의미 속성을 찾아 숫자로 표현하는데, 스키마(S_1)과 스키마(S_2) 간의 벡터 변환표는 <표 6>과 같이 나타낼 수 있다. 다음은 서브스키마(S_1)과 서브스키마(S_2) 간의 속성 집합(attribute set)을 나타내는데, 이러한 개체와 속성 집합들은 데이터베이스의 스키마 사전(schema dictionary)과 시스템 카탈로그(system catalog)로부터 간단하게 추출할 수 있다.

① 서브스키마(S_1)과 서브스키마(S_2)간의 전체 속성집합(Fa)으로 표현된다.

Fa_{12} = {*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*}

② 각각의 개체(E_k)에 속해 있는 속성 집합(F_{aik})으로 표현된다.

〈표 4〉 서브스키마(S_1) 속성 집합(F_{aik})과 개체명 대비표

$F_{a1,\ 101}$	$F_{a1,\ 102}$	$F_{a1,\ 103}$	$F_{a1,\ 104}$	$F_{a1,\ 105}$	$F_{a1,\ 106}$
Education	*Member*	*Security-Requirement*	*Member-Officer*	*Language*	*Commissioned-Officer*

$F_{a1,\ 101}$ = {*CollegeName, Duration, Major, YearCompleted*}
$F_{a1,\ 102}$ = {*Ethnic, MemberName, Race, SSN, Sex*}
$F_{a1,\ 103}$ = {*SecurityAgency, SecurityCode, SecurityInvestDate*}
$F_{a1,\ 104}$ = {*ActiveCommissionDate, DateOfBirth, PresentGrade*}
$F_{a1,\ 105}$ = {*DateLastTested, LanguageCode, ListenningProficiency, ReadingProficiency, SpeakingProficiency, WritingProficiency*}
$F_{a1,\ 106}$ = {*ActiveCommissionDate, DateOfBirth, Name, PresentGrade, SSN, Sex*}

〈표 5〉 서브스키마(S_2) 속성 집합(F_{aik})과 개체명 대비표

$F_{a2,\ 201}$	$F_{a2,\ 202}$	$F_{a2,\ 203}$	$F_{a2,\ 204}$	$F_{a2,\ 205}$	$F_{a2,\ 206}$
Education	*Commissioned-Officer*	*Security-Requirement*	*UNIT*	*YearGroup*	*Language*

$F_{a2,\ 201}$ = {*CollegeName, Duration, Major, YearCompleted*}
$F_{a2,\ 202}$ = {*ActiveCommissionDate, DateOfBirth, Name, PresentGrade, SSN, Sex*}
$F_{a2,\ 203}$ = {*SecurityAgency, SecurityCode, SecurityInvestDate*}
$F_{a2,\ 204}$ = {*Date, Grade, YearGroupNo*}
$F_{a2,\ 205}$ = {*ActivityTitle, BillSequenceNo, DateAssigned, Duration*}
$F_{a2,\ 206}$ = {*Code, DateLastTested, ListenningProficiency, ReadingProficiency, SpeakingProficiency, WritingProficiency*}

③ 속성이 갖는 값들이 a_{ikl} 일 때, 속성 벡터(B_{ik})로 나타내고, 두 개체간에 동일 속성의 유무를 0과 1로 나타낼 수 있다. 즉, 전체 속성 집합(Fa)과 각각의 개체별 속성 집합(F_{aik})의 항목들을 순차적으로 비교해서 전체 속성집합(Fa)에 존재하면 1, 존재하지 않으면 0을 표내의 해당 셀에 기록한다. 두 스키마간의 벡터 변환표는 <표 6>과 같다.

〈표 6〉 서브스키마(S_1)과 서브스키마(S_2) 간 벡터 변환표

구 분	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
$B_{1,\ 101}$	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$B_{1,\ 102}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0
$B_{1,\ 103}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$B_{1,\ 104}$	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$B_{1,\ 105}$	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
$B_{1,\ 106}$	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
$B_{2,\ 201}$	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$B_{2,\ 202}$	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
$B_{2,\ 203}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
$B_{2,\ 204}$	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$B_{2,\ 205}$	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$B_{2,\ 206}$	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0

(4) 두 스키마간에 공통으로 갖는 개체들의 친밀도는 개체 친밀도 값이 가장 큰 개체간에 우선 의미 충돌을 해결한다. 추출된 2개 이상의 유사 속성을 포함한 개체는 다음과 같다. 여기서 친밀도 값은 <표 6>의 두 속성 벡터(B_{ik})간에 동일 열(column)에 1이 나타나는 개수로 표현된다. 즉, 친밀도 값 6이면 두 속성 집합(F_{aik})간에는 동일한 속성이 6개 존재한다는 의미이다. 따라서 동일한 속성을 많이 포함하면 유사한 의미 개체로 보아 우선 통합의 고려 대상이 된다.

$A_{12,\ 101,\ 202} = 4$, $A_{12,\ 101,\ 205} = 1$, $A_{12,\ 102,\ 202} = 2$, $A_{12,\ 103,\ 203} = 3$
$A_{12,\ 104,\ 202} = 3$, $A_{12,\ 105,\ 206} = 5$, $A_{12,\ 106,\ 202} = 6$

따라서, 개체 통합의 우선 고려 순서는 다음과 같이 결정된다.

$A_{12,\ 106,\ 202} > A_{12,\ 105,\ 206} > A_{12,\ 101,\ 202} > A_{12,\ 103,\ 203} >= A_{12,\ 104,\ 202} > A_{12,\ 102,\ 202} > A_{12,\ 101,\ 205}$

서브스키마간의 친밀도 분석 결과 2개 이상의 유사 속성을 포함한 개체가 6개 추출되었다. 따라서 서브스키마 친밀도 분석 과정에서 특성(property)과 제약 조건(constraints)에 대한 고려는 설계자의 주관적인 결정이 중요해진다.

3.2 충돌해결(Conflict resolution)

3.2.1 명칭충돌(Name conflict)

① 동일 개체명칭과 개념들이 유사한 특성과 관계(동일어)

동일어 사례에서 개체명칭이 동일한 개체는 *"Commissioned-Officer"*, *"Education"*, *"Language"*, *"Language"*, *"Security-Requirement"*로 추출되었다.

② 다른 개체명칭과 개념들이 유사한 특성과 관계(synonym)

이음동의어 사례에서 *"MemberOfficer"*의 속성은 *"CommissionedOfficer"*의 부분집합이다. 따라서 이 두 개의 개체명을 *"NavalOfficer"*로 재명명하고, *"Member"*와는 서로 개체관계도 상에서 포함관계이므로 *"NavalOfficer"*를 *"CommissionedOfficer"*의 하위개체가 되도록 하여 통합을 결정한다.

③ 동일 개체명칭과 개념들이 다른 특성과 관계(homonym)

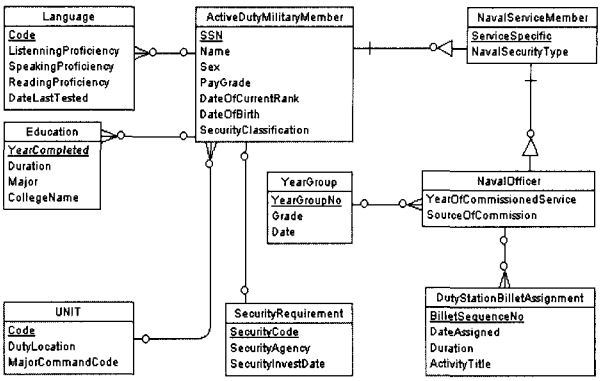
동음이의어 사례에서 서브스키마(S_2)와 서브스키마(S_3)의 개체명 *"UNIT"*은 명칭은 동일하나, 속성내용이 서로 달라 동음이의어로 판정하고, 서브스키마(S_2)의 개체명을 *"DutyStationBilletAssignment"*로 재 명명하여 충돌을 해결한다.

3.2.2 구조적 충돌(Structural conflict)

구조적 충돌은 타입충돌(Type conflict)과 카디날리티 충돌(Cardinality conflict)이 있는데, 사례와 같이 *"Education"* 개체는 서브스키마(S_1)과 서브스키마(S_2)에 동일어로 존재한다. 따라서 동일 개체명칭과 개념들이 유사한 특성과 관계를 갖는 예로 명칭충돌에서 언급되었다. 그러나 서브스키마(S_3)의 개체 *"ActiveDutyMilitaryMember"*의 속성이 *"Education"*으로 타입충돌이 발견되었다. 그리고 다른 사례에서 발견된 타입충돌의 해결방법은 *"ActiveDutyMilitaryMember"* 개체에서 *"Education"* 속성을 제거하고, *"ActiveDutyMilitary-Member"* 개체와 *"Education"* 개체간 관계를 설정하여 충돌을 해결한다.

3.3 스키마 통합 결과

서브스키마 친밀도 분석과 의미개체 친밀도 분석을 통하여 공통 유사 서브스키마의 추출이 가능하고, 이러한 과정을 통해 추출된 서브스키마 정보를 이용하여 서브스키마 그룹 결정과 개체간 충돌 해결이 가능하다. 따라서 이 방법론을 적용하면 MDBS 구축을 위한 개념적 스키마의 통합이 용이하고, 응용 프로그램의 접근성과 데이터 무결성 그리고 서브스키마 관리가 편리해진다. 다음 (그림 5)는 분산 서버에 존재하는 서브스키마(S_1, S_2, S_3)의 통합된 스키마 ERD를 나타낸다.



(그림 5) 전체 통합 스키마의 개념적 모델

3.4 적용 방법론 평가

대규모 기업 집단간의 합병이나, 진부화된 데이터베이스 재구축을 통한 사용자 요구 사항을 수용하기 위해서 MDBS 구축에 기업들의 관심이 매우 높다. 따라서 본 연구는 대규모 MDBS의 재설계 및 통합을 지원하기 위한 스키마 시소러스(thesaurus) 개발을 통한 스키마 통합 자동화시스템 알고리즘 개발을 위한 기초 연구이다. 본 방법론의 주요 특징을 보면 분산된 서버에 존재하는 ERD를 이용한 서브스키마 통합 방법론이고, 설계자 주관에 의한 통합순서 결정을 친밀도 분석을 통한 공통 유사 서브스키마 추출과 통합 순위결정 그리고 의미 충돌해결을 통한 스키마 통합이다. 본 방법론은 기존의 방법론 보다 세분화되어 자동화시스템 개발의 단계와 문제점 해결을 더욱 명확하게 해준다.

4. 결론 및 향후연구

본 논문에서는 데이터베이스 통합시 가장 기본적인 단계인 공통 유사 서브스키마 추출을 통한 스키마 통합 방법론을 다루었다. 본 방법론은 친밀도 분석, 유사 서브스키마 추출, 통합순서 결정, 의미충돌 해결, 그리고 스키마 통합의 5단계로 구성되었으며, 본 방법론의 특징은 친밀도 분석을 통한 공통 유사 서버스키마 추출과 스키마 통합의 접근이라는 것이다. 이 방법이 단일 데이터베이스뿐만이 아니라, 다중 데이터베이스 구축에도 유용하게 적용될 수 있다는 것을 확인할 수 있었다. 또한, 본 방법론은 이용자들의 응용 프로그램 액세스 요청에 대한 접근성 및 데이터 무결성, 그리고 관리의 용이성을 높이고, 분산된 스키마들을 논리적으로 통합하는 공통된 뷰를 제공하는 것이다.

향후 연구는 대규모 MDBS의 재설계 및 통합을 위한 스키마 시소러스의 개발과 상용 시소러스를 활용한 데이터베이스 설계시 발생하는 명명법(스키마, 서브스키마, 개체, 속성)에 대한 표준화 연구가 스키마 통합의 자동화 구현시 중요한 요소이므로, 이 분야에 대한 연구가 진행중에 있다.

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] S. B. Navathe, R. Elmasri, J. Larson, "Integrating User Views in Database Design," IEEE Computer, Jan., 1986.

[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] S. Castano, V. D. Antonellis, "Semantic Dictionary Design for Database Interoperability," ACM SIGSOFT, pp.43-54, 1996.

[6] S. B. Nabathe, R. Elmasri, J. Larson, "Integration User Views in Database Design," IEEE Computer, Jan., 1986.

[7] P. Martin, W. Powley, "Database Integration using Multidatabase Views," CASCON'93, pp.1-14, 1993.

[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, S. 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] I. Mirbel, "Semantic integration of conceptual schemas," Proceedings of the First International Workshop on Applications of Natural Language to Databases(NLDB'95).

[11] R. Jakobovits, "Integrating Autonomous Heterogeneous Information Sources," University of Washington, July, 1997.

[12] 이희석 외 3명, "의미객체 모델을 이용한 뷰통합 지원시스템 개발", 한국경영정보학회 학술대회논문집, pp.52-70, 1996.

[13] 임병학, "의미객체 모델을 이용한 뷰통합 지원시스템 개발", 한국과학기술원 석사학위논문, 1996.

[14] 이상태 외 1명, "객체-관계 데이터 모델을 토대로 한 다중 데이터베이스 시스템의 스키마 통합에 관한 연구", 한국멀티미디어학회 춘계학술대회논문집, pp.256-261, 1998.

[15] 김기중, "통합 데이터베이스를 위한 스키마 통합 방법", 정보통신학회논문지, 제6권 제2호, pp.1-13, 1998.

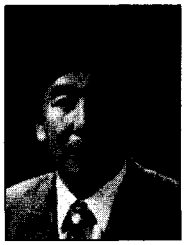
[16] 김홍수, "참조 스키마 생성을 위한 개념적 스키마 분석", 한국OA학회논문지, 제7권 제4호, pp.83-88, 2002.

[17] 신기태 외 3명, "제조 데이테베이스 설계에서의 뷰 통합 방안", 한국경영과학회논문지, 제11권 제3호, pp.668-670, 1994.

[18] 최태광 외 5명, "데이터 유사성 척도를 이용한 생산정보 데이터베이스의 분산 구조 설계", 대한산업공학회논문지, 제8권 제3호, pp.269-278, 1995.

[19] 이원조 외 3명, "스키마 통합을 위한 시소러스의 활용 방안", 한국정보과학회 춘계학술대회, 제30권 제1호, pp.668-670, 2003.

[20] 이원조, "유사 서브스키마 추출을 통한 스키마 통합 방법론", 정보과학회전문대학논문지, 제11권 제2호, pp.151-160, 2003.



고 재 진

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

1972년 서울대학교 응용수학과 공학사

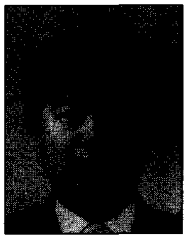
1981년 서울대학교 대학원 계산통계학과 이학석사

1990년 서울대학교 대학원 컴퓨터공학과 공학박사

1975년~1979년 한국후지쯔(주) 기술개발부 사원

1979년~현재 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : DB시스템, 전문가 시스템, DB설계, ERP



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