韓國IT서비스學會誌
第7卷 第4號
2008年 12月

통합 정보시스템에서의 데이터 이질성 해결 방안에 관한 연구*

박성진** · 박성공*** · 박화규****

A Study on the Method for Solving Data Heterogeneity in the Integrated Information System*

Seong Jin Park** · Sung Kong Park*** · Hwa Gyoo Park****

■ Abstract ■

As the technologies for telecommunication have been evolving, more enhanced information services and integrated information systems have been introduced, which can manage a variety of information from the heterogeneous systems. The major obstacle for the integrated information systems is the integrating heterogeneous databases in the systems and the heterogeneity problems can be classified into the structural and data heterogeneities. However, the previous researches have mainly highlighted into the solving structural heterogeneity problems.

This paper identifies the data heterogeneity problems for multi-database schema integrations and proposes a new solving method. We analyze the semantics equivalence in data values based on the functional dependency, primary and candidate keys, and present a procedural solution of data heterogeneity in the perspective of the concept of attribute equivalence, integration key and conceptual integration table.

Keyword : Integrated Information System, Multidatabase, Schema Integration, Data Heterogeneity

논문투고일 : 2008년 07월 22일 논문수정완료일 : 2008년 09월 08일 논문게재확정일 : 2008년 09월 10일
* 이 논문은 한신대학교 학술연구비 지원에 의하여 연구되었음.
** 한신대학교 컴퓨터공학부 교수
*** 고려대학교 대학원 컴퓨터학과 박사과정
**** 순천향대학교 보건행정학과 교수, 교신저자

1. 서 론

최근 글로벌화와 조직체간의 치열한 경쟁에 의한 기업, 공공기관, 연구소 등의 합병이 빈번해짐에 따라 이들 조직체들의 기존 정보시스템들의 효율적인 통합 문제가 주요 이슈로 부각되고 있다. 이러한 정보시스템들의 통합 과정의 핵심은 바로 데이터베이스 통합(database integration)으로 이질적이고 분산된 데이터베이스들에 대한 통일된 사용자 뷰의 제공과 통합 관리의 효율성이 해결해야할 쟁점으로 인식되고 있다. 특히, 개별적으로 운영되던 기존 데이터베이스를 통합하고자 할 때 해결해야하는 가장 주요한 문제 중의 하나는 바로 데이터베이스의 스키마 이질성(schema heterogeneity)이다. 이러한 스키마 이질성은 데이터가 저장된 데이터베이스의 구조(즉, 관계형, 객체관계형, 객체형 데이터베이스, 문서 저장소 등)와 데이터베이스 접근 방법(즉, SQL, Web 인터페이스, 메시지)에 따라 다양하게 나타난다. 멀티데이터베이스 통합 과정에서의 이질성은 크게 두 가지 문제 즉, 구조 이질성과 데이터 이질성 문제로 분리하여 생각할 수 있다[1].

스키마의 구조 이질성(structure heterogeneity)은 데이터베이스 스키마사이의 구조적인 상이함으로 인해 발생한다. 즉, 통합하고자 하는 멀티데이터베이스들의 물리적 스키마가 서로 다름으로 인해 발생하는 문제이다. 이 스키마 이질성에는 단순하게는 테이블 이름, 속성 이름이 상이한 경우부터 속성의 물리적 타입, 개수 등이 다르거나 스키마 구조가 한쪽은 속성으로 다른 한쪽은 테이블로 설계된 경우까지를 포함한다.

스키마의 데이터 이질성(data heterogeneity)이란 의미 이질성(semantic heterogeneity)이라고도 하며 통합 시에 스키마 구조가 같더라도 같은 의미(semantics)를 갖는 데이터가 서로 다른 표현(representation)을 가짐으로 인하여 발생하는 문제이다. 데이터 이질성은 같은 의미라도 서로 다른 표현 방식(예를 들어, 서로 다른 언어를 사용하거나 약어를 사용하는 등)을 사용하거나 단위(scaling)를 달리 쓰는 경우 그리고, 서로 다른 코드 체계 등을 사용했을 때 발생할 수 있다.

데이터베이스 통합과 관련한 스키마 이질성을 해결하기 위해 많은 연구들이 수행되었다. 기존 연구들은 구조 이질성에 대해서는 다양한 해결 방법들을 제시하였지만, 데이터 이질성에 대한 다양한 해결 방법이 부족하고 대부분의 접근 방법들이 메타정보에 의존적이라는 단점이 있다. 즉, 제공되는 스키마에 관한 메타정보가 불충분하거나 심지어 존재하지 않는 경우에 대한 적절한 해결방안이 필요하다.

본 연구에서는 멀티데이터베이스 스키마 통합 과정에서의 데이터 이질성 문제에 대하여 정의하고, 속성 의미 동치성에 기반 한 문제 해결 방안을 제안하였다. 제안한 방법은 스키마에 대한 메타 정보(도메인, 타입, 제약조건 등)에 의존적인 기존 방법들과는 달리 데이터베이스의 실 속성 값들의 연관관계 분석을 통해 데이터 이질성을 해결함으로써 서로 다른 데이터 코드체계를 갖는 데이터와 같이 직접적인 데이터 값의 기계적 분석을 통해서만 해결될 수 있는 경우에 적합한 방법이다. 즉, 기존 구조적, 의미적 통합 방법들을 대체하기보다는 보완할 수 있는 데이터의 이질성 해결 방법이다.

2. 관련 연구

데이터의 의미를 고려한 정보 통합에 관한 기존 연구들은 세 가지 접근 방법으로 분류할 수 있다. 첫 번째 접근방법은 질의 언어(query language)를 통한 방법이다. 다수의 요소 데이터베이스를 공통된 언어로 접근할 수 있도록 단일 질의 언어로 제한하여 전역적인 뷰를 통한 통합 방법이다[2, 3, 4]. [4]에서는 다수의 요소 데이터베이스를 접근할 수 있는 nD-SQL이란 질의어를 제안하였는데, "ibm"이라는 단어가 어떤 요소 데이터베이스에서는 데이터값, 속성 이름 혹은 테이블 이름으로 존재할 수 있는데, 이러한 스키마 구조 이질성을 처리하

는 방법을 제안하였다. 그러나, 이 접근 방법의 경우, 통합 관리자가 적합한 뷰를 생성하기 위하여 복잡한 질의어를 작성하여야 하는 문제점을 갖는다. 두 번째 접근방법은 CDM(Common Data Model)을 통한 데이터 의미 통합이다[5, 6]. CDM이란 정보 소스의 의미를 통합하기 위해 단일화한 공통된 데이터 모델을 말한다. CDM은 소스 데이터의 모델을 통합한 공통된 모델로 변형하여 데이터에 대한 통일된 접근방법을 제공하는 것이다. 세 번째 접근방법으로는 전역적인 개념(concept)에 기반한 데이터 통합방법이다. 이 방법은 주로 온톨로지(ontology)를 이용하여 데이터의 의미를 일치시키는 방법이다[7]. 그러나 유용한 온톨로지를 구축하는 데는 많은 시간과 노력이 필요하고 온톨로지에 대한 질의 처리의 비효율성이 문제점으로 인식되고 있다.

이러한 기존 연구들은 스키마 통합을 위한 구조 이질성에 대한 해결방법은 제시하지만, 데이터 이질성에 대한 해결방안 제시에는 미흡한 실정이다. 기존 연구들이 고려하는 데이터 이질성의 성격은 주로 단위의 변환이나 동의어에 대한 고려, 개념에 기반한 데이터의 고려에 한정되어있다[8]. 그러나, 기존의 방법은 각종 장비들의 고유한 모델명이나 특정 도메인에 국한되지 않는 고유명사(예, 사람의 이름, 도시이름)들에 대하여서는 해결 방법을 제시하지 못하는 단점이 있다.

멀티데이터베이스 객체의 구조적 유사성과 의미적 유사성을 비교문맥 측면에서 함께 고려한 통합적 연구[9]의 경우도 스키마 제약조건과 같은 메타정보에 기반하고 있어 메타정보가 없거나 부실한 경우에 적용에 어려움이 있다.

본 논문에서 제시하는 의미 동치성에 기반한 데이터 이질성 해결 방법은 기존 연구들의 약점인 속성에 대한 메타정보가 없거나 부족한 경우에 대한 대안으로 실 데이터 값의 연관관계 분석에 따른 동치 속성을 찾고 동치 속성간의 데이터 매핑을 지원한다. 제안하는 방법은 속성(attribute) 수준에서의 의미 이질성을 해결함으로써 필요한 지식의 양을 줄일 수 있을 뿐만 아니라 온톨로지 기반의 스키마 통합 방법을 보완할 수 있다. 즉, 온톨로지를 구축하는데 어려움을 겪는 인명, 도시명, 상품명과 같은 고유명사의 데이터 이질성을 해결하는데 효과적이다.

또한, 스키마 통합 자동화와 관련 연구 중[10]에서는 속성의 이름 유사도와 구조적 유사도를 이용하여 의미가 동일한 속성을 일정부분 자동화하는 방법을 제안하고 있다. [10]에서 제시하는 방법은 속성 동치일 가능성이 높은 후보 속성을 탐색하는데 적용할 수 있으므로, 본 논문에서 제시하는 방법과 결합될 수 있다.

3. 데이터 이질성 해결을 위한 의미 동치

3.1 데이터 이질성의 분류

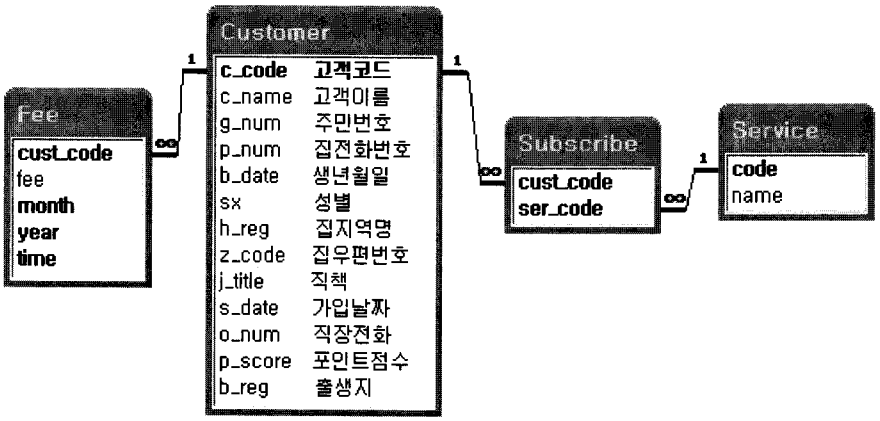
논문에서 제안하는 데이터 이질성 해결 방법을 적용하기 위한 예제 데이터베이스는 [그림 1], [그림 2]와 같다. 통합 대상인 CDB_1과 CDB_2는 임의의 서비스 가입자 정보들에 관한 데이터베이스로 가정한다.

본 논문에서는 예제 데이터베이스의 각 요소를 기술하기 위하여 도트(dot) 표기법을 사용한다. 즉, 데이터베이스명은 CDB_1, CDB_2로, 테이블명은 CDB_1.Customer, CDB_2.Fee로 속성명은 CDB_1.Customer.job, CDB_2.Fee.month으로 표기한다. 예에서 CDB_2.Fee.month는 CDB_2데이터베이스의 Fee 테이블안의 month속성을 의미한다.

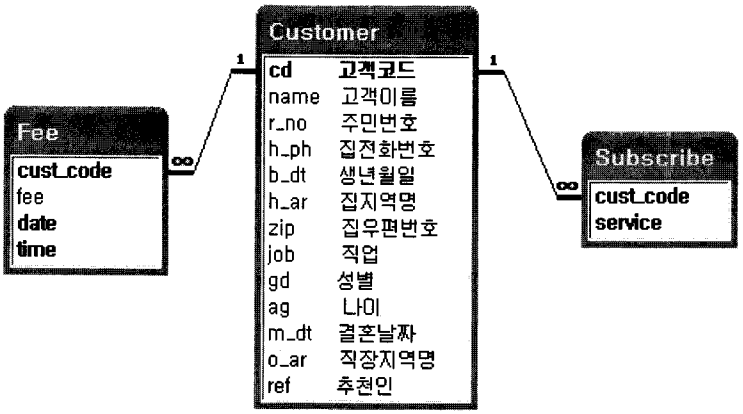
스키마 통합 시에 데이터베이스 구조에 대한 이질성을 해결하더라도 추가적으로 통합을 위해 해결해야할 다양한 데이터 이질성의 종류를 분류하면 다음과 같다.

3.1.1 다중언어(multi-lingual) 이질성

다중언어 이질성은 같은 의미를 갖는 속성에 대해 서로 다른 언어를 사용했을 경우 발생한다. 예



[그림 1] CDB_1 데이터베이스 스키마



[그림 2] CDB_2 데이터베이스 스키마

를 들어, [그림 3]과 [그림 4] 예제 데이터베이스의 CDB_1.Customer.c_name 속성과 CDB_2.Customer.name 속성은 같은 이름(홍길동)에 대해서 각각 한글(홍길동)과 영문(Hong, K. D.)으로 표현하고 있다. 이러한 이질성은 "홍길동"의 통화료를 찾는 질의에 대해서 "Hong, K. D."은 가입자가 아니라고 간주하고 질의에 대한 만족스러운 결과를 반환하지 못한다.

<예 1> 질의 : "홍길동"의 2002년 1월 통화료는 얼마인가?

원하는 결과 : CDB_1에서 20000, CDB_2에서 30000

3.2.2 범주 데이터(categorical data) 이질성

범주 데이터 이질성은 데이터 값들 사이에 의미적 계층 구조가 존재하는 경우에 발생한다. 직업이 학생인 모든 사람의 이름을 반환하는 질의에 대해서 CDB_1.Customer.j_title 속성의 "대학생"과 CDB_2.Customer.job 속성의 "학생"은 서로 동일한 의미 관계에 놓여 있지만, 결과로 반환되지 않는다. 이러한 문제는 두 요소 데이터베이스의 통합할 경우뿐만 아니라 단일 테이블 내에서도 발생할 수 있다.

<예 2> 질의 : 직업이 "학생"인 고객의 이름과

Customer : 테이블

c_code	c_name	g_num	p_num	b_date	sx	h_reg	z_code	j_title	s_date	o_num	p_score	b_reg
AB001	홍길동	700323-1005522	331-8822	70,03,23	남	서울	135-617	회사원	93,04/30	980-1677	40	광주
AB002	김철수	771012-1003311	325-1902	77,10,12	남	용인	446-769	대학원생	97,08,18	467-8403	25	서울
AB003	박영희	801230-2006543	324-1905	80,12,30	여	화성	435-766	교사	99,10,15	379-6651	77	수원
AF010	김찬	880103-1328876	452-8490	88,01,03	남	수원	410-777	대학생	02,07,21	230-7792	15	수원
AZ111	박미선	780714-2550638	477-8901	68,07,14	여	성남	210-050	대학원생	92,09,28	650-2145	23	성남

Fee : 테이블

cust_code	fee	month	year	time
AB001	20000	1	2007	500
AB002	25000	1	2007	600
AB003	15000	1	2007	400

Service : 테이블

code	name
S01	발신전화표시
S02	착신통화전화
S03	3인통화

Subscribe : 테이블

cust_code	service
PC002	발신번호표시
PC002	착신통화전화
PC003	발신번호표시

[그림 3] CDB_1 데이터베이스 예제 데이터

Customer : 테이블

cd	name	r_no	h_ph	b_dt	h_ar	zip	job	gd	ag	m_dt	o_ar	ref
PC002	Hong, K. D.	700323-1005522	331-8822	70/03/23	서울	135617	회사원	M	39	00/01/06	서울	김나미
PC003	Kim, C. S.	771012-1003311	634-1902	77/10/12	경기	446769	학생	M	32	07/12/11	서울	이금철
PC006	Lee, D. S.	610303-1008799	622-3324	61/03/03	서울	142700	공무원	M	48	09/08/21	경기	박나희
PC007	Kim, C.	880103-1328876	452-8490	null	경기	410777	회사원	M	21	null	경기	김영목
PC132	Park, M. S.	780714-2550638	null	68/07/14	경기	210050	학생	F	31	04/05/26	경기	김길창

Fee : 테이블

cust_code	fee	date	time
PC002	30000	1/2002	10,5
PC003	25000	1/2002	8

Subscribe : 테이블

cust_code	service
PC002	발신번호표시
PC002	착신통화전화
PC003	발신번호표시

[그림 4] CDB_2 데이터베이스 예제 데이터

주민번호, 직업 목록을 검색하라.

원하는 결과 :

CDB_1에서 김철수, 771012-1003311, 대학원생

박영희, 801230-2006543, 대학생

박미선, 780714-2550638, 대학원생

CDB_2에서 Kim, C. S., 771012-1003311, 학생

3.1.3 코드화(encoding) 이질성

코드화 이질성은 값들의 유일성을 보장하면서 관리상의 목적으로 사용하는 숫자나 문자열 즉, 코드 값이 동일하거나 유사한 의미로 사용될 경우에 발생한다. 실세계의 데이터베이스에서 이러한 현상은 자주 발생한다. 예를 들어, CDB_1.Subscribe.ser_code와 CDB_2.Subscribe.service는 같은 의미를 서로 다른 코드로 표현한 것이다.

<예 3> 질의 : 발신번호표시 서비스에 가입한 가입자의 목록을 검색하라.

원하는 결과 : CDB_1에서 홍길동, 박영희

CDB_2에서 Hong, K. D., Kim, C. S.

3.1.4 단위(scaling) 이질성

단위 이질성은 서로 다른 도량형을 사용할 경우 흔히 발생하는 문제이다. CDB_1.Fee.time은 통화시간을 초단위로 표시한데 반하여, CDB_2.Fee.time은 통화시간을 분단위로 표시하였다.

<예 4> 질의 : 통화시간이 500초 이상인 가입자의 목록을 검색하라.
원하는 결과 : CDB_1에서 홍길동, 김철수
CDB_2에서 Hong, K. D.

3.2 의미 동치(semantics equivalence)

스키마 통합에 있어 단순히 구조 통합만 했을 경우에는 앞의 <예 1>~<예 4> 질의에 대한 원하는 결과를 얻을 수 없다. 즉, 스키마 통합에 있어 구조 통합만으로는 모든 문제점들이 해결될 수 없으며 이를 위해서는 데이터의 의미 통합도 함께 이루어져야한다. 이러한 의미 통합을 위해서는 통합 대상이 되는 데이터들을 결정하고 대응되는 데이터간의 연관 관계에 대한 분석이 필요하며 이를 위해 본 논문에서는 의미 동치 개념을 새로이 제안하였다. 이 과정에서 의미 동치의 개념은 기존 함수 종속성과 유사하게 각 속성의 데이터 값들 사이의 연관성을 평가하는데 함수 종속성과는 다르게 각기 다른 테이블에 위치한 속성들간의 의미적 연관성을 분석하는 것으로 방향성과 동치 기준이 되는 속성이 별개로 존재한다는 점에서 구별된다.

논문에서 데이터간의 의미 동치는 다음과 같이 정의할 수 있다.

[정의 1] 데이터 의미 동치(data semantics equivalence)
특정 속성 a_1과 특정 속성 a_2의 각 도메인 내에 존재하는 임의의 데이터 값 v_1과 v_2에 대해서 v_1에서 v_2로 사상하는 의미변환 함수 $f(v_1) = v_2$가 존재하면 v_2가 속한 속성 a_2에 대하여 v_1과 v_2는 의미 동치 또는 데이터 동치라고 하며 $dataSem_{a2}(v_1) = dataSem_{a2}(v_2)$로 표현한다.

특정 속성에 존재하는 어떤 데이터 값이 다른 속성에 존재하는 다른 데이터 값과 동일한 의미를 갖는 경우를 데이터 의미 동치 즉, 데이터 동치 관계에 있다고 할 수 있다. 결국, 스키마 통합에서의 데이터 통합 과정은 궁극적으로 각 속성에 존재하는 데이터 값과 같은 의미를 갖는 데이터 값들의 연관 관계를 분석하는 따라서, 연관 관계에 있는 속성들 간의 매핑 규칙(mapping rule) 혹은 매핑 함수(mapping function)를 결정해가는 과정으로 정의할 수 있다. 데이터 동치의 경우, 속성의 데이터 값이 고유명사나 코드처럼 그 의미를 직관적으로 파악하기 어려운 경우에는 $dataSem_{a2}(v_1)$과 $dataSem_{a2}(v_2)$의 개별 데이터의 의미 동치나 연관 함수를 한번에 정의하기 어려울 수 있으며 이러한 경우, 동일 속성의 여러 데이터 값들의 데이터 동치 관계와 의미함수를 반복해서 비교분석함에 따른 속성 수준의 의미 동치를 파악함을 통해서 보다 명확해질 수 있다.

[정의 1]의 데이터 의미 동치에 대한 정의를 기반으로 하여 속성 a_1과 속성 a_2가 의미적으로 연관 관계에 있음을 속성간의 의미 동치로 다음과 같이 정의할 수 있다.

[정의 2] 속성 의미 동치(attribute semantics equivalence)
특정 도메인 D와 R에 속하는 임의의 속성 a_1과 속성 a_2에 대하여 데이터 의미 동치 $dataSem_{a2}(v_1) = dataSem_{a2}(v_2)$를 만족하는 함수 $f : D \rightarrow R$ 단, $D = Domain(a_1)$, $R = Domain(a_2)$인 f가 존재할 때 a_2의 도메인 R에 대하여 a_1과 a_2는 속성의미 동치 또는 속성 동치라고 하며 $dataSem_R(a_1) = dataSem_R(a_2)$로 표현한다.

속성의미 동치 즉, 속성 동치는 사상함수 f의 차수 즉, v_1과 데이터 값 속성 동치인 v_2의 수에 따라 분류하면 다음과 같다.

- 완전 속성 동치(complete attribute equivalence) : 속성 a_1의 모든 데이터 값 v_1에 대하여 데이터 의미 동치 관계에 있는 속성 a_2의 데이터 값 v_2가 오직 하나만 존재하고, v_2에 대해서도 데이터 의미동치인 v_1이 하나만 존재할 때, 즉, 사상함수 f가 일대일 관계이면서 f의 역함수가 존재할 때 속성 a_1과 속성 a_2는 완전 속성 동치라고 한다. 즉, $dataSem_R(a_1) = dataSem_R(a_2)$이면서 $dataSem_D(a_2) = dataSem_D(a_1)$인 경우를 의미한다.
- 불완전 속성 동치(incomplete attribute equivalence) : 속성 a_1의 모든 데이터 값 v_1에 대하여 데이터 의미 동치 관계에 있는 속성 a_2의 데이터 값 v_2가 존재하지만, 역은 성립하지 않는 경우, 즉 f의 역함수가 존재하지 않을 때 불완전 속성 동치라고 한다. 즉, $dataSem_R(a_1) = dataSem_R(a_2)$이면서 $dataSem_D(a_2) \neq dataSem_D(a_1)$인 경우나 $dataSem_R(a_1) \neq dataSem_R(a_2)$이면서 $dataSem_D(a_2) = dataSem_D(a_1)$인 경우를 의미한다.

또, 속성 동치는 도메인과 속성의 실 데이터 집합과의 관계에 따라 분류할 수 있다. $D_1 = \{v_1 \mid dataSem_{a2}(v_1) = dataSem_{a2}(v_2), Domain(a_2) \ni v_2\}$이고, $D_2 = \{v_2 \mid dataSem_{a1}(v2) = dataSem_{a1}(v1), Domain(a_1) \ni v_1\}$이라고 가정할 때 다음 4가지 유형으로 분류할 수 있다.

- 전체 속성 동치(full attribute equivalence) : $D_1 = Domain(a_1)$ 이면서 $D_2 = Domain(a_2)$인 경우, 전체 속성 동치라고 한다.
- 내포 속성 동치(nested attribute equivalence) : $D_1 = Domain(a_1)$이면서 $D_2 \subset Domain(a_2)$인 경우, 내포 속성 동치라고 한다.
- 포함 속성 동치(inclusive attribute equivalence) : $D_1 \subset Domain(a_1)$ 이면서 $D_2 = Domain(a_2)$인 경우, 포함 속성 동치라고 한다.
- 부분 속성 동치(partial attribute equivalence) : $D_1 \subset Domain(a_1)$ 이면서 $D_2 \subset Domain(a_2)$인 경우, 부분 속성 동치라고 한다.

위와 같은 분류에 따르면 예제 데이터베이스안의 CDB_1.Customer.c_name속성은 CDB_2.Customer.name속성과 완전 속성동치이면서 또, 부분 속성 동치 관계에 있다고 정의할 수 있다.

위와 같은 다양한 분류의 속성 동치를 사용하여 모든 데이터 값들 사이의 동치 관계를 개별적으로 정의하지 않고 속성 수준에서 일반화시켜 정의할 수 있다. 즉, 속성 동치를 사용함으로써 데이터 동치 수준에 비해 보다 요약해서 함축적으로 표현할 수 있다.

4. 의미 동치 분석을 이용한 스키마 통합

스키마 통합을 위한 각 데이터들에 대한 이질성 탐색은 데이터베이스 관리자의 각종 전문지식, 데이터베이스 스키마에 대한 메타 정보와 각 속성에 존재하는 실 데이터 값들에 대한 분석을 통하여 이루어진다[11]. 그 결과 데이터 값들의 이질성을 탐색하는 과정에서 발견된 이질성들을 해결하기 위한 사상 규칙이 구성되고, 이 사상 규칙을 통해 서로 다른 스키마에 속한 데이터들의 통합이 가능해진다.

이때 스키마 통합 시에 각 데이터 값과 동일한 의미를 갖는 데이터 값을 결정하는 방법 중 대표적 방법은 온톨로지를 활용하는 방법이다. 이 방법은 데이터 값이 일반 명사나 전문분야에서 사용하는 전문 용어인 경우에 적합한 방법이다. 그러나, 일반적인 용어에 비해 데이터베이스에 저장되는 데이터는 관리상의 목적 때문에 코드, 약어와 같은 다양한 표현을 갖는 경우가 많다. 특히, 인명이나 특정 상품의 모델명과 같은 경우에는 온톨로지를 구축하는 비용과 효용 가치 측면을 상호 비

교하면 결코 효율적이지 않다. 다른 접근 방법으로는 동일한 의미를 갖는 데이터 값을 속성 수준에서 표현하는 방법이다. 즉, $dataSem_R(a_1)$ = $dataSem_R(a_2)$를 만족하는 매핑 규칙이나 매핑 함수를 사용하는 방법이다.

본 논문에서는 제시한 다양한 속성 동치 유형을 이용함으로써 직접적인 데이터들 사이의 의미 동치를 기술하는 것보다 필요한 정보의 양을 감소시켰다.

스키마 통합 과정에서의 데이터 이질성 해결 과정은 크게 다음 4개의 단계로 나누어 수행되어진다.

4.1 후보키 탐색 단계

각 테이블에서 함수 의존성(functional dependency) 관계를 찾아내는 단계로 기본 키(PK; Primary Key)를 포함한 속성 중에서 후보키(CK; Candidate Key)들을 <표 1>과 같이 탐색한다.

〈표 1〉 후보키 탐색

테이블명	후보키 대상
CDB1.Customer	g_num, c_code
CDB1.Service	name
CDB2.Customer	r_no, cd

기본키와 기본키 이외의 후보키는 일대일 관계를 가지며 일반적으로 같은 의미의 다른 표현으로 볼 수 있다. 예를 들면, CDB_1.Customer.g_num와 CDB_1.Customer.r_no의 경우 모두 고객을 의미하는 다른 표현으로 볼 수 있다. 하지만, 기본키(PK) 이외에도 의미 동치 관계에 있는 테이블간의 속성들을 찾기 위한 통합키 선정을 위해 후보키(CK)를 검색한다.

이때, 앞에서 정의한 속성 동치를 적용하면 기본키와 후보키는 해당 데이터 값들 사이에 속성 의미 동치 관계에 있다고 할 수 있다.

4.2 통합키 선정 단계

통합하기 위해서 먼저 각 데이터베이스에서 통합의 핵심 연결고리 역할을 할 수 있는 속성을 선택해야 한다. 통합의 키 역할을 할 수 있는 속성은 특정 테이블의 후보키이면서 두 데이터베이스 이상에 존재하는 공통 속성이다. 통합된 두 데이터베이스의 개념적인 기본키 역할을 할 수 있는 속성을 통합키(IK; Integration Key)로 선정한다.

예제 데이터베이스의 경우, 각 데이터베이스의 Customer 테이블안의 속성 g_num, r_no는 후보키이며, 두 데이터베이스에 존재하며 같은 타입과 크기를 가지며 공통의 데이터가 존재하므로 이를 통합키(IK)로 선정한다.

CDB_1.Customer.g_num(IK)
↔ CDB_2.Customer.r_no(IK)

4.3 개념적 통합 테이블 생성 단계

선정한 통합키를 기본키로 개념적 통합 테이블(conceptual integration table)을 생성한다. 이 테이블은 통합 과정을 위해 통합 데이터베이스 관리자에 의해 생성되는 테이블로 실제 물리적으로 생성되는 것은 아니며 개념적으로만 존재한다. 이 과정은 두 개의 테이블을 통합키를 이용하여 전체 외부조인(full outer join[12])함으로써 생성할 수 있다.

[그림 5] 테이블에서 r_no와 g_num은 통합키이고 조인된 테이블은 요소 데이터베이스의 테이블 속성들로 이루어져 있다. 단, 통합키에 의해 조인된 동일행의 데이터는 실세계의 동일 개체를 표현하므로 데이터 값의 연관성을 분석함에 의해 동일한 의미를 갖는 속성들 간의 연관성을 찾아낼 수 있다.

4.4 사상 규칙 생성 단계

어떤 속성들이 상호간에 속성 동치인지를 찾기

CUSTOMER_통합:테이블 : 통합 쿼리

cd	name	r_no	h_ph	b_dt	h_ar	zip	job	gd	ag	m_dt	o_ar	ref	c_cod	c_nam	g_num	p_num	b_date	sx	h_reg	z_code	j_title	s_date	o_num	p_score	b_reg
													AB003	박영희	801230-2006543	324-1905	80,12,30	여	화성	435-766	교사	99,10,15	379-6651	77		수원
PC002	Hong, K, D	700323-1005522	331-8822	70/03/23	서울	135617	회사원	M	39	00/01/06	서울	김나미	AB001	홍길동	700323-1005522	331-8822	70,03,23	남	서울	135-617	회사원	93,04/30	960-1677	40		광주
PC003	Kim, C, S,	771012-1003311	634-1902	77/10/12	경기	446769	학생	M	32	07/12/11	서울	이금철	AB002	김철수	771012-1003311	325-1302	77,10,12	남	용인	446-769	대학원생	97,08,18	467-8403	25		서울
PC006	Lee, D, S,	610303-1008799	622-3324	61/03/03	서울	142700	공무원	M	48	09/08/21	경기	박나희													
PC007	Kim, C,	880103-1328876	452-8490	null	경기	410777	회사원	M	21	null	경기	김영록	AF010	김찬	880103-1328876	452-8490	88,01,03	남	수원	410-777	대학생	02,07,21	230-7792	15		수원
PC132	Park, M, S,	780714-2550638	null	68/07/14	경기	210050	학생	F	31	04/05/26	경기	김길찬	AZ111	박미선	780714-2550638	477-8901	68,07,14	여	성남	210-050	대학원생	92,09,28	650-2145	23		성남

[그림 5] 개념적 통합 테이블 예

위해서 데이터베이스에 존재하는 모든 속성들을 하나하나 검색하는 것은 효율적이지 않다. 따라서, 다음과 같은 측면들을 고려하여 동치 후보 속성을 우선적으로 탐색하는 것이 바람직하다.

4.4.1 테이블 이름, 속성이름간의 연관성 분석

스키마를 설계하는 과정에서 적절한 테이블 이름과 속성 이름을 사용할 경우, 일반적으로 속성 이름이 같거나 유사하면 동일한 의미를 가질 확률이 높은 것으로 판단할 수 있다. 이름 유사성 분석에 따른 예제 테이블 customer간의 동치 후보 속성은 다음과 같다. <표 2>의 예에서, r_no 속성의 경우, g_num, p_num, o_num와 속성이름이 일치하지는 않지만 'no'와 'num'의 경우 일반적으로 동일한 의미로 사용되기 때문에 후보 속성으로 판단한다.

〈표 2〉 테이블명/속성명 연관성 분석 결과

속성명 (CDB2.customer)	동치 후보 속성 (CDB1.customer)
name	c_name
r_no	g_num, p_num, o_num

4.4.2 속성 타입 및 크기의 일치성 분석

같거나 유사한 타입과 같은 크기를 가지는 속성들의 경우에도 속성 동치 관계에 있을 확률이 상대적으로 높다. 속성 타입과 크기 유사성 분석에 따른 예제 테이블 customer간의 동치 후보 속성은 다음과 같다. <표 3>의 예에서, cd는 c_code와 문자형으로 단위 크기가 같고, job은 j_title과 속성 타입이 정수로 같으며 m_dt는 날짜형으로 b_date, s_date 속성과 같아 동치 후보 관계에 있다고 할 수 있다.

〈표 3〉 속성타입/크기 일치성 분석 결과

속성명 (CDB2.customer)	동치 후보 속성 (CDB1.customer)
cd	c_code
r_no	g_num
h_ph	p_num, o_num
b_dt	b_date, s_date
h_ar	h_reg, b_reg
zip	z_code
job	j_title
ag	p_score
m_dt	b_date, s_date
o_ar	h_reg, b_reg
ref	c_name

4.4.3 속성 데이터 값들의 유사성 분석

속성 이름과 속성 타입 및 크기의 유사성이 발견된 속성들의 실제 데이터 값을 탐색한다. 두 속성에 존재하는 데이터 값들 중 공통된 데이터가 존재하는 정도가 높을수록 속성 동치일 확률이 높다. 데이터 값들의 유사성 분석에 따른 예제 테이블 customer간의 동치 후보 속성은 다음과 같다. <표 4>의 예에서, r_no은 g_num과, h_ph는 p_num과 일치하는 데이터 값이 많으며, b_dt는 b_date와 zip은 z_code와 형식이 조금 상이하지만 역시 일치하는 데이터가 많아 후보 속성으로 선정한다. 또, h_ar속성은 h_reg와 job속성은 j_title과 도메인이 유사하므로 후보 속성으로 선정할 수 있다.

〈표 4〉 속성 값 유사성 분석 결과

속성명 (CDB2.customer)	동치 후보 속성 (CDB1.customer)
r_no	g_num
h_ph	p_num
b_dt	b_date
h_ar	h_reg
zip	z_code
job	j_title

다음은 선정된 동치 후보 속성들을 대상으로 개념적 통합 테이블의 통합키를 매개로 하여 연관 속성간의 데이터 사상 규칙을 결정한다. 속성 동치를 판단하기 위해서는 통합 테이블로부터 앞서 기술한 동치 후보 속성들에 대해 동일한 통합키 값을 갖는 두 후보 속성들 간의 함수적 의미 동치 관계가 있는지 판단하여 일정한 변환 함수 관계가 있는 경우, 속성 동치를 판단한다. 그 뒤, 제시된 다양한 데이터 이질성의 유형중 어디에 속하는지를 결정하고 통합키, 함수 종속성, 속성 이름, 속성의 도메인(타입과 크기), 속성의 유사성 등의 메타정보가 없더라도 실 데이터를 검증함으로써 사상 규칙을 정의하고 의미적으로 동일한 속성을 최종적으로 확정한다.

다음 <표 5>는 속성 동치 관계에 있는 속성들을 찾고 그에 대한 사상 규칙을 결정한 예를 보여준다.

데이터 이질성중 다중언어에 의한 이질성 범주에 속한 속성들의 동치 관계를 발견한 예로 $dataSem_R$ (c_name) = $dataSem_R$(name)는 한글 고객명과 영문 고객명으로 의미가 같되 표현 언어가 서로 다르게 표현된 의미적으로 동일한 속성으로 관련 질의에 대해 한글명과 영문명사이의 매핑을 통한 통합 검색이 이루어져야 한다. 데이터 이질성중 범주화에 의한 이질성 범주에 속한 속성들의 동치 관계를 발견한 예로 $dataSem_R$(j_title) = $dataSem_R$(job)는 한쪽은 대학생과 대학원생으로 세분화된 신분정보를, 다른 한쪽은 일반적인 학생으로만 신분을 표현한 의미적으로 동일한 속성으로 관련 질의에 대해서 관련 속성 데이터들간의 표현 방식을 일반화 또는 상세화함으로써 적절한 통합 검색이 이루어져야 한다. 데이터 이질성중 코드화에 의한 이질성 범주에 속한 속성들의 동치 관계를 발견한 예는 다양하며 그 중에서 $dataSem_R$ (c_code) = $dataSem_R$(cd)의 경우, 내부적으로 사용하는 고객관리를 위한 식별 코드로 통합키를 통한 일대일 매핑을 통해서만 동치 관계에 있음을 확인할 있으며 관련 질의에 대해서도 통합키를 통한 대응 코드 정보의 변환을 통해 검색 결과가 제공되어야 한다.

속성 비동치 관계에 있는 속성들에 대한 예로는 $dataSem_R$(p_score) ≠ $dataSem_R$(ag)의 경우, 정수형으로 속성 타입과 도메인은 유사하지만 한쪽은 포인트 점수를, 한쪽은 나이 정보를 표현하기 때문에 의미적으로 같지 않다. 즉, 상호간에 속성 값을 서로 대치할 수 없으므로 동치 관계로 적절하

〈표 5〉 속성동치 및 사상규칙 분석 결과

속성 동치 내용	사상 규칙(비사상 이유)	이질성 유형	속성 동치 유형
dataSemR(c_code) = dataSemR(cd)	상이한 사용자코드간 일대일매핑	코드	완전/부분
dataSemR(c_name) = dataSemR(name)	한글명과 영문명간 일대일 매핑	언어	완전/부분
dataSemR(p_num) = dataSemR(p_ph)	전화번호 표시방식간 매핑	코드	완전/부분
dataSemR(b_date) = dataSemR(b_dt)	생년월일 표시방식간 매핑	코드	완전/부분
dataSemR(sx) = dataSemR(gd)	성별 한영약어표현방식간 매핑	언어	완전/전체
dataSemR(h_reg) = dataSemR(h_ar)	지역광역 단위표현방식간 매핑	범주	불완전/포함
dataSemR(z_code) = dataSemR(zip)	집우편번호 표시방식간 매핑	코드	완전/부분
dataSemR(j_title) = dataSemR(job)	직업 단위표현방식간 매핑	범주	불완전/포함
dataSemR(c_name) ≠ dataSemR(ref)	(가입자와 추천인 의미상이)	-	-
dataSemR(s_date) ≠ dataSemR(m_dt)	(가입날짜와 결혼날짜 의미상이)	-	-
dataSemR(o_num) ≠ dataSemR(h_ph)	(직장번호와 집전화번호 의미상이)	-	-
dataSemR(p_score) ≠ dataSemR(ag)	(포인트점수와 나이 의미상이)	-	-
dataSemR(b_reg) ≠ dataSemR(o_ar)	(출생지와 직장지역명 의미상이)	-	-

다고 할 수 없다.

지금까지의 예들은 데이터 이질성중 다중언어, 범주, 코드화로 인한 이질성에 따른 동치 속성들로써 적절한 대치를 통해 상호간에 사상이 가능하지만 문자형 데이터가 아닌 수치 데이터의 경우 단위 이질성을 갖는 경우가 상당히 많다. 그 예로 각 Fee 테이블의 time 속성간에 $dataSem_R(time)$ = $dataSem_R(time)$ 속성 동치가 경우를 들 수 있으며 한쪽은 사용시간을 초단위로, 다른 한쪽은 분단위로 표현하고 있기 때문에 분초간의 단위변환을 통해 요청된 질의에 대한 통합 검색 결과를 제공할 수 있다.

논문에서 제시한 속성 동치를 통한 데이터 이질성 해결 방안은 속성값들 사이의 코드 이질성, 언어 이질성, 단위 이질성, 범주 이질성의 4가지 데이터 이질성 유형을 극복하고 데이터를 통합하는데 효율적이며 메타 정보가 없거나 충분치 않는 경우에도 적용이 가능하고 특히, 코드명, 지역명, 인명 등과 같은 온톨로지를 사용하더라도 해결하기 어려운 데이터 통합에 이점을 갖는다. 그러나, 데이터 통합 이전에 스키마 구조의 통합이 선행되어야만 한다는 기존 전제가 있어 독립적인 해결방법이라기 보다는 기존 구조적 통합과 결합되어 적용하는 것이 바람직하며 통합 대상인 테이블에 충분한 실데이터가 존재해야만 정확한 데이터 통합이 가능하다는 제한점이 있다.

5. 결 론

본 논문에서는 테이블 내의 데이터 값들에 대한 특성을 조사하여 이러한 정보를 바탕으로 멀티데이터베이스 스키마 통합 시에 발생할 수 있는 데이터 이질성을 해결하는 방법을 제안하였다. 정보통합 분야에서 이러한 의미적 통합을 이루려는 노력은 지금까지 많이 진행되어 왔다. 그러나, 주로 구조적 통합에 초점이 맞추어져 있으며 데이터 이질성에 대한 추가적 고려 없이는 완전한 의미적 통합은 불가능하다. 본 논문에서는 데이터 이질성 문제에 대해 새롭게 정의하고 이를 해결하기 위한 방안으로 속성 동치 개념을 사용하여 데이터 값들 사이의 의미 동치성을 분석하고, 데이터 이질성을 해결하는 4단계 과정을 제시하였다.

제안한 방법은 스키마에 대한 메타 정보(도메인, 타입, 제약조건 등)에 의존적인 기존 방법들과는 달리 데이터베이스의 실 데이터의 연관관계 분석을 통해 데이터 이질성을 해결함으로써 고유 명사나 서로 다른 데이터 코드 체계를 갖는 데이터와 같이 직접적인 데이터 값의 기계적 분석을 통해서만 해결될 수 있는 경우에 효과적인 방법이다.

본 논문에서 제안하는 속성 의미 동치에 기반한 통합 방법은 기존 구조적, 의미적 통합방법을 대체하기보다는 보완할 수 있는 데이터의 이질성 해결 방법으로 기존의 구조중심의 스키마 통합 접근 방법과 결합되어 데이터베이스 통합 과정 자동화에 적용될 수 있다.

참 고 문 헌

[1] Won Kim and Jungyun Seo, "Classifying Schematic and Data Heterogeneity in Multidatabase Systems", *IEEE Computer*, Vol. 24, No.2(1991), pp.12-18.

[2] Litwin, W., "MSQL : A Multidatabase Language", *Information Science*, 1989.

[3] Lifer Michael, Kim Won and Sagiv Yehoshua, "Querying Object-Oriented Databases", *ACM SIGMOD*, Vol.21, No.2(1992), pp.393-402.

[4] Frederic Gingras and Laks V. S. Lakshmanan, "nD-SQL : A Multi-dimensional Language for Interoperability and OLAP", *Proceedings of the 21th VLDB Conference*, 1998, pp.134-145.

[5] Soon M. Chung and Pyeong S. Mah, "Schema Integration for Multidatabases Using the Unified Relational and Object-Oriented Model", *Proceedings of the 23rd ACM An-*

nual Conference, 1995, pp.208-215.

[6] Cheng Hian Goh, Stephane Bressan, Stuart Madnick, and Michael Siegel, "Context Interchange : New Features and Formalism for the Intelligent Integration of Information", *ACM Transactions on Information Systems*, Vol.17, No.3(1999), pp.270-293.

[7] Adam Farquhar, Richard Fikes, Wnada Pratt, and James Rice, "Collaborative Ontology Construction for Information Integration", *Technical Report*, Dept. of Computer Science, Stanford University, Knowledge Systems Laboratory, 1995.

[8] Ramez Elamsri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Benjaming/Cummings, 1994.

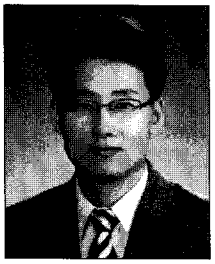
[9] Vipul Kashyap and Amit Sheth, "Semantic and Schematic Similarities between Database Objects : A Context-based approach", *VLDB Journals*, Vol.5, No.4(1996), pp.276-304.

[10] Silvana Castano, Valeria De Antonellis, and Sabrina De Capitani di Vimercati, "Global Viewing of Heterogeneous Data Sources", *IEEE Transactions on Knowledge and Data Engineering*, Vol.13, No.2(2001), pp.277-297.

[11] Marek Rusunkiewicz, Amit Sheth, and George Karabatis, "Specifying Interdatabase Dependencies in a Multidatabase Environment", *IEEE Computer*, Vol.24, No.12(1991), pp.46-53.

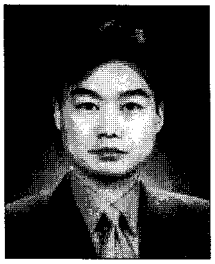
[12] Laks V. S. Lakshmanan, Feredoon Sadri, and Iyer N. Subramanian, "Logic and Algebraic Language for Interoperability in Multidatabase Systems", *Journal of Logic Programming*, Vol.33, No.2(1997), pp.101-149.

◈ 저 자 소 개 ◈



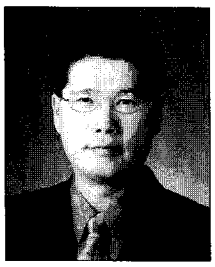
박 성 진 (sjpakr@hs.ac.kr)

고려대학교에서 전산학 석사를 하고 동 대학에서 분산 데이터베이스 시스템으로 박사학위를 취득하였으며 한국전자통신연구원에서 선임연구원으로 근무하다 현재 한신대학교 컴퓨터공학부 교수로 재직중이다. 주요 연구 관심분야는 데이터 웨어하우징, 지식관리, 데이터 마이닝, 웹컨텐츠 통합 등이다.



박 성 공 (skpark89@dreamwiz.com)

고려대학교에서 전산학 석사를 하고 현재 고려대학교 전산학 박사과정에 재학 중이다. 주요 연구 관심분야는 데이터베이스, 센서네트워크, 데이터 통합, 메타데이터 등이며, 관련 학회에 다수 논문을 실었다.



박 화 규 (hkpark1@sch.ac.kr)

캘리포니아주립대 및 오클라호마주립대에서 대학원과정을 수료하고, KAIST에서 경영공학박사를 취득하였다. 한국과학기술연구원 및 한국전자통신연구원에서 선임연구원을 거쳐 현재 순천향대학 의료과학대 교수로 재직중이며, 주요 연구분야는 고객관계관리, u-healthcare, 서비스 경영정보체계등이다.