

PDM을 위한 하이브리드 데이터베이스 통합 모델에 관한 연구*

이강찬, 이 상, 유정연, 이규철**

A Study on Hybrid Database Integration Model for Product Data Management

Kangchan Lee, Sang Lee, Jeongyeon Yu, Kyuchul Lee

Abstract

In a centralized database system, all system components reside at a single platform. In recent years there has been a rapid trend toward the integration of information systems over multiple sites that are interconnected via a communication network, and users' needs are changed to integration of multiple information sites. Multi database System is one of solutions for integrating distributed heterogeneous databases. However the problems in multi database system are restriction in distributed environment support, limitation in integrating heterogeneous media type data, static integration, and data-only of integration. In order to solve these problems, we propose a hybrid database integration model, HyDIM. HyDIM is used for the integrating legacy multimedia data, adopting CORBA, MDS, and mediator. We demonstrate a prototype system for PDM application domain.

Keyword: database integration, product data management, agent, mediator, multi database system, CORBA

* 본 연구는 충남대학교 소프트웨어연구센터의 기본과제(과제번호 : 98-11-01-02-A-2)와 국방과학연구소의 무기체계 통합 DB 기술 개발 과제(과제번호 : UD980002GD)의 일환으로 수행된 결과임.
** 충남대학교 공과대학 컴퓨터공학과, 데이터베이스 연구실

1. 서론

지난 수 십 년간 정보 시스템의 확산에 따라 각기 다른 데이터 모델을 지원하는 수많은 데이터베이스 관리 소프트웨어가 사용되어 왔다. 정보시스템의 초창기에는 파일 시스템을 사용하여 정보 시스템을 개발하였으며, 시간이 지남에 따라 사용자는 계층형 DBMS, 네트워크 DBMS, 관계형 DBMS, 객체 지향형 DBMS, 객체 관계형 DBMS 과 같은 다양한 모델을 지원하는 DBMS 를 이용하였다.

그러나 현대 컴퓨터 시스템의 발달로 인하여 분산되어 있는 데이터에 대한 통합의 요구가 발생하게 되었으며, 더 나아가 분산에 대한 해결뿐만 아니라 서로 이질적인 데이터에 대한 통합의 요구가 발생하게 되었다. 즉, 기존에 구축되어진 정보시스템에 대하여 통합을 시도함으로써 각각의 데이터 저장소에는 전혀 영향을 끼치지 않으며, 그들 데이터를 통합함으로써 새로운 정보를 창출하는 시도들이 증가하게 되었다.

이러한 시도들은 CALS, 데이터 웨어 하우스, 디지털 도서관, 가상 대학 및 가상 기업등에서 찾아볼 수 있다. 특히 데이터베이스의 통합에 대한 노력은 CALS 의 PDM (product data management) 분야에 더욱 두드러지게 나타난다[Gasoigne, 1995]. PDM 은 제품 전주기에 걸쳐 제품과 관련된 데이터를 관리하고 정보에 대한 추적성(tractability)을

제공하여 제품 개발 프로세스를 관리하는 제품 데이터 관리 체계로써, PDM 을 사용하면 신속하고 정확한 제품 정보 활용 체계를 갖추게 되므로 제품이 시장에 출하되기까지의 시간을 단축하게 된다[PDM, 1997]. 데이터베이스 통합은 PDM 의 근간이 되는 중요한 기술이며, 단순한 통합이 아닌 위치 투명성(Location Transparency) 및 접근 투명성(Access Transparency)을 보장하는 통합이 되어야 한다.

이와 같은 응용은 서로 다른 조직에서 서로 다른 환경을 가지고 있으며, 각 시스템을 구성하는 컴포넌트들이 상이한 하드웨어 및 플랫폼, 운영체제에서 구동 될 때에 이질성이 높아지게 된다. 따라서 서로 다른 데이터베이스를 사용하고 있는 상태에서, 각 데이터에 대한 특별한 조작 없이 원하는 데이터에 대한 통합된 정보를 제공하는 것이 데이터베이스 통합의 목표이다.

이러한 데이터베이스 통합을 위해 많은 연구들이 수행되었다. 초창기의 연구로서 데이터 변환(data conversion)과 게이트웨이(gateway)를 이용한 통합 방법이 제안되었다[WonKim, 1995a].

데이터 변환은 이전에 사용하던 데이터 및 인터페이스(interface)를 새로운 환경에서 사용할 수 있도록 변환하는 방법이다. 데이터 변환의 문제점은 데이터의 변환 비용이 오버헤드가 될 뿐만 아니라 응용 프로그램

까지 바꿔주어야 하는 문제점이 있다. 즉 데이터를 통합하고자 하는 데이터베이스의 데이터 포맷으로 변환하기 용이하지 않을 뿐만 아니라, 데이터를 변환 할 지라도 기존에 사용하던 응용 프로그램을 새로운 시스템에서 사용할 수 있어야 하므로 응용 프로그램도 변환해야 한다.

게이트웨이를 사용하는 방법은 데이터 변환 방법이 아닌 질의 변환 방법으로써 단순히 접근하고자 하는 데이터베이스의 언어로 원래의 질의를 변환하여 수행하는 방법이다. 게이트웨이를 사용하였을 경우의 문제점은 그 첫번째로 트랜잭션 관리 기능이 없다는 것이다. 따라서 데이터베이스의 고유 기능인 동시 수행 제어나 회복 기능을 자유로이 사용할 수 없다. 또한 단순히 질의 변환만 하기 때문에 두 데이터베이스의 스키마의 구조적인 면에서의 유사성이나 스키마 간의 차이를 표현하는 방법을 제공하지 못한다는 것이다.

데이터 변환과 게이트웨이를 이용한 방법은 각기 그 단점을 가지고 있기 때문에 진정한 의미의 데이터베이스 통합이라 볼 수 없다. 데이터 변환과 게이트웨이의 단점을 극복하고 완전한 데이터베이스 기능을 이용할 수 있는 멀티 데이터베이스(MDB : Multi Database)를 사용하고 있다. 그러나 멀티 데이터베이스도 적용 분야에 따라 장단

점을 가지고 있다.

본 논문에서는 먼저 데이터베이스 통합 방법으로 멀티 데이터베이스를 소개하고 PDM에서 멀티 데이터베이스가 적합하지 않음을 지적하고, PDM 환경에서 적합한 데이터베이스 통합 방법에 대하여 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 멀티 데이터베이스 시스템에 대한 소개와 그 기능에 대하여 설명하였으며, 멀티 데이터베이스를 PDM에 적용할 때 발생하는 문제점에 대하여 지적하였다. 3장에서는 PDM 환경에서 데이터베이스 통합을 위한 모델인 HyDIM과 그 구현에 대한 설명이며, 4장에서는 본 논문과 유사한 데이터베이스 통합 방법에 대한 연구에 대하여 설명하였다. 마지막으로 5장에서 결론 및 향후 연구에 대하여 언급하였다.

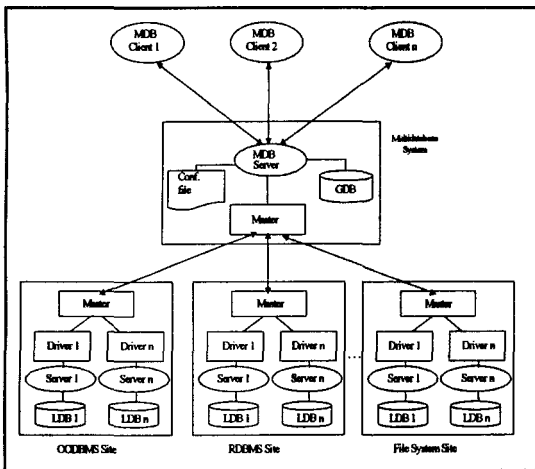
2. 멀티 데이터베이스 시스템 (MDS : Multi Database System)

2.1 시스템 개요

멀티 데이터베이스는 기존에 존재하는 DBMS나 파일 시스템들의 최상위에 존재하는 데이터베이스 시스템으로써 사용자에게는 단일 데이터베이스의 모습으로 보여주는 완전한 데이터베이스의 기능을 갖춘 DBMS이다. 멀티 데이터베이스는 통합하고

자 하는 데이터베이스 간에 통합된 총체적인 스키마 만을 관리하며, 모든 사용자 데이터는 지역 데이터베이스가 관리한다.

또한 멀티 데이터베이스의 트랜잭션은 실제로 데이터를 관리하고 있는 지역 데이터베이스에 의하여 조정되며, 멀티 데이터베이스 시스템은 그 하부구조로 통합 가능한 여러 개의 지역 데이터베이스를 가지고 있으며 멀티 데이터베이스와 지역 데이터베이스는 각각의 드라이버를 통하여 데이터를 전달한다[Calton, 1991].



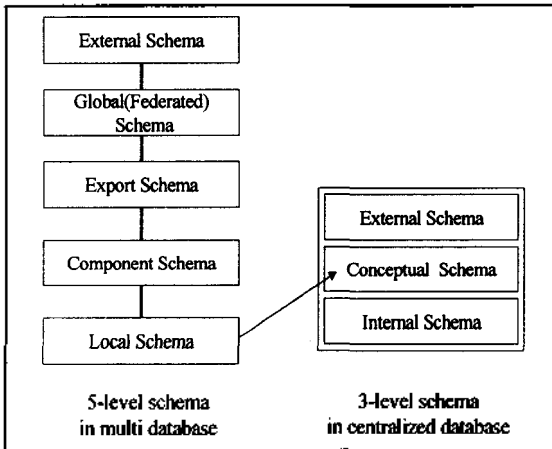
<그림 1> 멀티 데이터베이스 구조

멀티 데이터베이스의 구성 요소는 전역(통합) 데이터베이스를 위한 멀티 데이터베이스 서버, 멀티 데이터베이스 클라이언트와 마스터(master), 드라이버(driver)로 구성되어 있다. 멀티 데이터베이스 서버와 클라이언트는 일반적인 클라이언트-서버 모델의 데이터베이스 관리 시스템과 같은 기능

이고, 마스터는 멀티 데이터베이스 클라이언트와 멀티 데이터베이스 서버 사이, 멀티 데이터베이스 서버와 지역 데이터베이스 드라이버간의 통신을 초기화하며, 전역 데이터베이스와 지역 데이터베이스의 기계 사이에서 상호 연결성(interconnectivity)을 보장하는 역할을 담당한다. 또한 드라이버는 전역 데이터베이스 엔진과 지역 데이터베이스 엔진의 질의 변환을 위한 모듈이다

데이터 변환이나 게이트웨이 방법과 비교해 볼 때, 멀티 데이터베이스는 지역 데이터베이스 스키마간의 차이와 구조적인 유사성을 스키마 매핑(mapping)에 의하여 표현할 수 있다. 두개 이상의 데이터베이스를 통합하게 되면 스키마 충돌과 데이터 충돌이 발생하게 된다. 스키마 충돌과 데이터 충돌은 각각의 지역 데이터베이스의 데이터를 통합할 때에 각각의 스키마 정보, 그리고 데이터에 대한 유사성이나 이질성 때문에 발생하게 된다. 예를 들어 스키마 충돌은 개체(entity) 이름과 어트리뷰트(attribute) 이름이 같은 내용에 대하여 다르게 사용하거나 또는 같은 내용에 대하여 지역 데이터베이스에 따라 개체, 어트리뷰트로 표현하는 경우이다. 또한 데이터 충돌은 같은 데이터를 표현하는데 다른 표기 방식을 사용함으로써 이들을 통합할 때 발생하게 되는 문제점이다.

멀티 데이터베이스 시스템은 지역 DBMS 와 전역 DBMS 의 스키마간에 매핑을 통하여 이러한 문제점을 해결한다. 스키마간에 매핑을 위하여 멀티 데이터베이스는 전형적인 3 단계 구조(외부, 개념, 내부)를 사용하는 일반 DBMS 를 확장하여 5 단계(외부, 전역, 수출, 컴포넌트, 지역) 구조를 사용한다.



<그림 2> 5 단계 스키마와 3 단계 스키마

지역 스키마(local schema)는 각각의 지역 데이터베이스의 개념 스키마이며 각각의 지역 데이터베이스의 데이터 모델을 표현하게 된다. 따라서 다른 지역 데이터베이스의 다른 스키마가 서로 다른 데이터 모델로써 표현이 되었다 할지라도 수용이 가능하다. 시스템에서 통합하기 위해서는 지역 스키마를 공통 데이터 모델로 변환하여야 하는데 이러한 공통 데이터 모델이 컴포넌트 스키마(component schema)이다. 수출

스키마(export schema)는 컴포넌트 스키마를 전역 DBMS 에서 사용할 수 있도록 변환한 것이며, 전역 스키마(global schema)는 다수개의 수출 스키마를 통합함으로써 구성되며, 마지막으로 외부 스키마(external schema)는 특정 사용자나 응용에 사용하도록 보여지는 가장 상위의 스키마이다.

UniSQL/M[UniSQL, 1994]에서는 스키마 충돌과 데이터 충돌을 해결하기 위한 방법으로 다음 그림과 같은 뷰(view) 정의를 확장한 멀티 데이터베이스 언어로써 매핑 관계를 형성한다.

```

CREATE VCLASS          virtual_class name
SIGNATURE             attr_def_list
AS SELECT             selection_list
FROM                  entity_spec_list
WHERE                 search_conditions,
...
SELECT                selection_list
FROM                  entity_spec_list
WHERE                 search_conditions

Entity_spec_list ::=      cdb_entity_name
[variable] {, cdb_entity_name [variable] }
Cdb_entity_name ::= [cdb name] entity_name
    
```

<그림 3> UniSQL/M 언어

UniSQL/M 의 언어는 전역 스키마의 가상 클래스와 지역 스키마의 명세로 구분되며 가상 클래스의 시그니처(signature)에는 전역 스키마의 어트리뷰트를 명세한다. 또한 지역 스키마 명세에는 통합하고자 하는

각각의 지역 데이터베이스의 내용을 기록한다.

UniSQL/M에서는 그림 3과 같은 언어를 제공함으로써 사용자로 하여금 스키마 충돌이 발생하였을 경우 사용자가 언어로 충돌을 해결할 수 있는 방법을 제공하며, 통합 시 발생하는 충돌의 해결은 사용자가 언어로써 피할 수 있도록 표현함으로써 해결된다.

2.2 PDM에서의 멀티 데이터베이스 시스템의 문제점

멀티 데이터베이스가 데이터 변환 및 게이트웨이 방법의 문제점을 어느 정도 해결할 수 있지만 PDM에서 관리하는 데이터를 통합하기에는 충분하지 않다. PDM 데이터베이스 통합을 위한 문제점은 다음과 같다.

- 제한된 분산 환경 제공 : 멀티 데이터베이스는 DBMS의 이질성만 고려하였을 뿐, H/W, 또는 운영 체제 등의 운영 환경의 이질성을 고려하지 않았다. 특히 멀티 데이터베이스 환경에서는 드라이버가 반드시 필요한데 멀티 데이터베이스 제공 회사에서 제공하는 드라이버의 수가 주요한 DBMS에 국한되어 있어 사용이 용이하지 않다.
- 제한된 데이터베이스 통합 : 스키마 통합은 서로 연관 있는 스키마만이 가능할 뿐이지 이질적인 데이터베이스 통합은 용이하지 않다. 즉 멀티 데이터베이스에서는 서로 같은 정보이지만 스키마 이름이 다르거나 어트리뷰트 이름이 다른 경우에 유용하지만, 완전히 상이한 데이터, 즉 다른 미디어 데이터인 문서, 도면 등의 자연스런 통합은 매우 어렵다.
- 정적 통합 방법 : 멀티 데이터베이스는 한번 스키마가 통합되면 통합된 스키마가 변하지 않는다. 즉 지역 데이터베이스의 구성을 변경하거나 또는 전역 데이터베이스의 스키마 구성을 변경하거나, 지역 데이터베이스의 데이터베이스 구조가 변경하게 되면 멀티 데이터베이스의 환경 파일을 변경하고, 변경된 환경 파일(configuration file)의 내용을 반영하기 위하여 MDB 서버를 다시 구동해야 한다.
- 데이터만 통합 : 멀티 데이터베이스는 단지 데이터만 통합하는 방법이다. 즉, 그 데이터베이스를 사용하는 시스템에서 제공하는 서비스는 제공하지 못한다. 그러나 현재 데이터베이스 통합은 그 사용이 데이터 통합 뿐만 아니라 그 서비스들인 어플리케이션까지 확대되고 있다. 진정한 의미에서의 스키마 통합은 데이터와 어플리케이션 프로그램

램과 동시에 아무런 손실 없는 완전한 통합이어야 한다.

3. 하이브리드 데이터베이스 통합 모델 (HyDIM)

3.1 HyDIM의 목적

본 논문에서 하이브리드 데이터베이스 통합 모델(HyDIM : Hybrid Database Integration Model)은 기존의 데이터베이스 통합 방법의 문제점을 해결하는 새로운 데이터베이스 통합 방법을 제안한다. 본 논문에서 제안하는 HyDIM의 주요 아이디어는 다음과 같다.

- HyDIM을 이용하여 통합하면 데이터 모델, DBMS 타입, 그리고 운영체제 및 하드웨어와 같은 기존의 데이터베이스 통합에서 문제점을 해결한다. 이를 위하여 HyDIM은 산업계에서 분산 환경 시스템 통합의 표준 인프라(infrastructure)가 되는 CORBA를 사용하였다. CORBA를 통하여 사용자는 원하는 정보가 어디에 위치하는지 또는 그 정보가 위치하는 컴퓨터가 어떠한 소프트웨어를 사용하는지에 투명하게 원하는 정보에 접근할 수 있다.
- HyDIM은 다양한 미디어 데이터 타입을 지원한다. PDM과 같은 응용에서

는 다양한 미디어 타입을 이용하게 되므로 그 데이터 통합도 다양한 미디어 타입을 지원하여야 한다.

- HyDIM은 새로운 데이터베이스에 대하여 즉각적으로 통합할 수 있어야 한다. MDS는 통합 이후에 그 통합에 대한 정보를 변경하기 어려울 뿐만 아니라, 변경 자체가 런타임(run-time)에 이루어질 수 없다. 그러나 HyDIM은 MDS의 단점을 극복하여 스키마 통합에 관한 정보를 동적으로 변경이 가능해야 하며, 이를 위하여 규칙(rule)으로 표현하게 된다. HyDIM은 두번째, 세번째의 요구사항을 위하여 미디어이터를 사용한다. 미디어이터는 지식을 기반으로 한 소프트웨어 모듈로서 다양한 데이터 소스들을 동적으로 중재하게 된다.
- 마지막으로, HyDIM에서 이루고자 하는 통합의 범위는 데이터 뿐만 아니라 어플리케이션이다. 본 논문에서는 어플리케이션의 의미를 데이터에 접근하기 위한 인터페이스라고 정한다. HyDIM의 미디어이터는 데이터 뿐만 아니라 어플리케이션간에 관계성을 규칙으로 표현할 수 있다.

즉, HyDIM은 MDS의 장점, 미디어이터, 그리고 CORBA를 채택하였다. HyDIM은 MDS, CORBA, 미디어이터를 조화롭게 이

융합으로써 데이터베이스 통합의 문제점을 해결하였으며, 다수개의 분산 이질 데이터베이스의 완전한 통합을 이룰 수 있다.

3.2 HyDIM의 주요 기술

3.2.1 CORBA

CORBA(Common Object Request Broker Architecture)는 OMG(Object Management Group)에서 1990년 가을에 제안한 것으로 이기종간에 상이한 분산 환경에서의 상호 연동성을 제공하고 다중 객체 시스템간에 완전한 상호 연결을 위하여 제안된 것으로 객체간에 서비스를 요청하고 그것에 대한 응답의 투명한 메커니즘을 제공한다[OMG, 1993].

CORBA는 80년대 중반 이후 널리 알려진 객체 지향 방식의 높은 생산성, 확장성 등을 채택하고 있어 최근에는 관련 상용 제품이 널리 보급되고 있다. CORBA는 객체의 상호 연결 버스 역할을 수행하는 ORB를 중심으로 응용 서비스를 담당하는 응용 객체(application objects), 응용에 관계없이 객체와 관련된 서비스를 담당하는 객체 서비스(object services), 그리고 대부분의 응용에 공통적으로 필요한 공통 기능(common facilities)으로 구성되어 있다.

CORBA의 기본구조는 클라이언트(Client)와 서버(Object Implementation), ORB(Object

Request Broker)등으로 구성되어 있다. 클라이언트는 동적 주문 API(Dynamic invocation API)를 사용하여 컴파일된 IDL(Interface Definition Language) 스텝(stub)을 거치거나 처리 중에 서비스를 정적으로 불러낸다. 서버는 정적 골격(static skeleton)을 사용하여 클라이언트 요구를 오브젝트 수행 프로그램에 연결하거나, 새로운 동적 골격 인터페이스를 사용하여 런타임으로 보내기도 한다.

HyDIM의 구현에서 서버는 CORBA IDL로 표현된 인터페이스를 정의한다. 클라이언트와 서버 사이의 데이터 교환은 IDL 구조와 시퀀스 등으로 정의된 자료구조를 통하여 이루어진다. IDL은 IDL 컴파일러로 컴파일되어 클라이언트와 서버의 소스 코드로 변환되며, 변환된 소스 코드에 메소드(method)에 대한 코드를 추가로 구현하였다.

클라이언트는 객체 참조(object reference)를 통하여 객체간 통신을 하게 된다. 객체 참조에 연산이 발생하게 되면 연산에 사용되는 파라미터(parameter)가 서버에 전달되고, 실제적인 객체가 그 파라미터를 받아 수행하게 된다. 그리고 그 결과는 다시 클라이언트에게 전달된다.

3.2.2 미디어이터(mediator)

미디어이터는 어떠한 모듈로부터 원하는 정보를 추출하여 상위 계층의 모듈에서 그 정보를 사용할 수 있도록 도와주는 소프트웨어 모듈이다[Wiederhold, 1992].

데이터베이스 통합의 관점에서 보면, 미디어이터는 이질적이고 다양한 정보 소스를 통합할 때 발생하는 문제점을 규칙을 이용한 지식 기반 데이터베이스를 사용하여 해결하는 방법을 의미한다.

멀티 데이터베이스 시스템과 비교하여 사용자는 미디어이터에 규칙을 동적으로 추가하거나 삭제할 수 있다. 멀티 데이터베이스 시스템에서는 스키마 통합 정보가 전역 스키마에 저장되어 있으며, 만약 이를 변경하려면 전역 데이터베이스의 환경을 모두 바꿔 주어야만 한다. 그러나 규칙을 이용하여 지역 데이터베이스의 정보를 표현하게 되면 동적으로 변경이 가능하다.

HyDIM 에서의 규칙은 미디어이터의 에이전트 사이의 메시지 기술 형태이다. 지역 데이터베이스의 모든 정보는 규칙으로 표현되고, 만약 충돌이 발생하게 되면 규칙을 이용하여 충돌을 해결한다. HyDIM 에서 사용하는 규칙은 IF-then 규칙이다.

일반적으로 If-then 규칙은 If 또는 선행(antecedent) 부분과 then 또는 후행(consequent)

부분과 같이 두 부분으로 나뉘어 진다. 규칙에서 조건은 절(clause)로 불린다. 만약 규칙에서 모든 선행절이 참이면, 후행절이 참이 되므로 후행절에 해당하는 새로운 정보가 추가되는 것이다.

R1 : If X1 = A and X2 = B then X3 = C1

예를 들어 위와 같은 규칙이 있을 경우 R1 은 규칙 이름이 되고, If 절의 내용(X1 = A and X2 = B)이 참이 되면, then 이후의 내용도 참이 되므로 새로운 정보(X3 = C1)를 추가할 수 있는 것이다.

3.2.3 데이터 특성

HyDIM 은 일반 데이터베이스에 저장되어 있는 구조화된 데이터가 아닌 비 구조화된 자료에 대한 데이터베이스의 통합에 주안점을 두고 있다. 따라서 데이터베이스가 저장하고 있는 데이터가 어떤 특성을 가지느냐가 매우 중요한 문제이다. 본 논문에서는 HyDIM 을 이용하여 구조화된 자료(예를 들어, 상품 가격 정보, 등)뿐만 아니라 이미지 데이터베이스와 문서 데이터베이스와 같은 다양한 미디어 타입의 데이터베이스에 대한 통합을 위한 모델을 제안하였다.

데이터의 표준으로 사용되는 것은 STEP (Standard Exchange for Product Data)과 XML

(Extensible Markup Language)로 각각은 그래픽 데이터 포맷과 텍스트 데이터를 위한 표준이다.

STEP은 ISO의 TC184/SC4에서 개발한 제품 데이터의 표현과 데이터 교환을 위한 표준이다[Owen, 1997]. STEP의 목표는 여러 나라들이 각자의 표준을 따로 가지고 있는 것보다는 모든 산업에서 제품 생명 주기의 각 측면을 다루는 더 우수한 표준을 하나로 만드는 것이다. STEP의 구성 요소는 EXPRESS 데이터 명세 언어와 구현 방법, 그리고 표준 데이터 모델로 이루어진다. EXPRESS는 저장할 데이터에 대한 자료 구조와 제약조건을 정의하기 위해 사용되는 데이터 명세 언어이다. 구현 방법은 실제적인 데이터 값과 데이터 모델을 구분하는 것으로서, 물리적 파일 형식과 SDAI(Standard Data Access Interface)의 두 가지 접근 방법이 있다. 마지막으로 데이터 모델은 EXPRESS로 기술된 STEP 데이터를 교환하고 공유하기 위하여 사용되는 모델링 방법이다. 본 논문에서는 STEP을 이용하는 방법은 DBMS에 EXPRESS로 기술된 제품 도면 데이터를 변환하여 저장하고 STEP의 표준 인터페이스인 SDAI로 접근하여 복원하는 방법이다[SDAI].

HyDIM의 또 다른 주요 데이터 표현 방법은 XML이다. XML은 현재 W3C에서

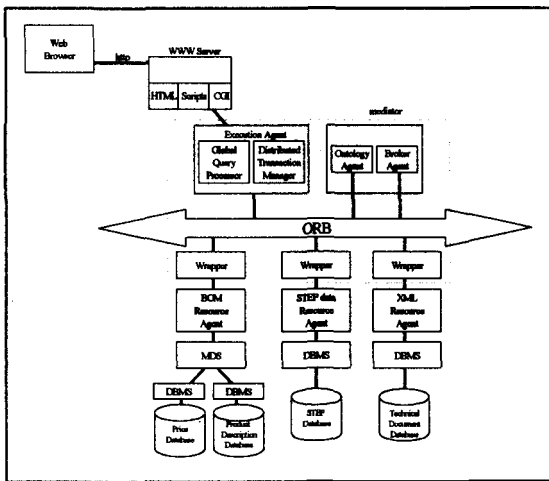
제안하고 있는 차세대 인터넷 표준 문서에 대한 규약이다[XML, 1997]. XML은 SGML(Standard Generalized Markup Language)의 복잡성과 SGML 시스템의 고가성으로 인한 기술적, 경제적인 측면을 완화하면서 HTML(Hyper Text Markup Language)이 갖는 한계점을 탈피함으로써 인터넷 환경에서 보다 편리한 방법으로 다양한 기능을 제공할 수 있도록 링크(link)나 포맷팅(formatting) 기술을 통합한 언어이다[XML, 1998]. 1996년 하반기에 제안된 XML은 현재 그 발전 속도와 툴들의 개발에 박차를 가하고 있으며 W3C에서는 XML, XLL(Xpointer, XLink), XSL을 그 표준 스펙으로 반영할 예정이다.

XML은 HTML과는 달리 사용자가 자신이 표현하고자 하는 문서 구조를 DTD(Document Type Definition)에 정의함으로써 사용하는 태그 셋을 설정할 수 있으며, 문서 작성은 지정한 태그 셋을 이용하여 하게 된다. 태그 셋은 DTD를 조작함으로써 추가, 삭제 할 수 있다.

또한 W3C는 XML 문서 인스턴스(instance)에 대한 접근 방법으로 DOM(Document Object Model)도 제안하고 있다[DOM, 1998]. DOM은 XML과 HTML 문서에 대한 플랫폼 중립적이고, 언어 중립적인 인터페이스로써 사용자는 DOM 인터페이스를 사용하여 동적으로 문서, 문서 구조, 문서 스타일에 대한 접근 및 수정이 가능하다.

3.3 시스템 구조

그림 4는 HyDIM을 기반으로 한 프로토타입 시스템 구조를 나타낸다. 이 시스템은 MDS, 미디어이터와 자원 에이전트로 이루어져 있다.



<그림 4> HyDIM을 기반으로 한 PDM을 위한 프로토타입(prototype) 시스템

본 시스템의 주요 기능은 다음과 같다.

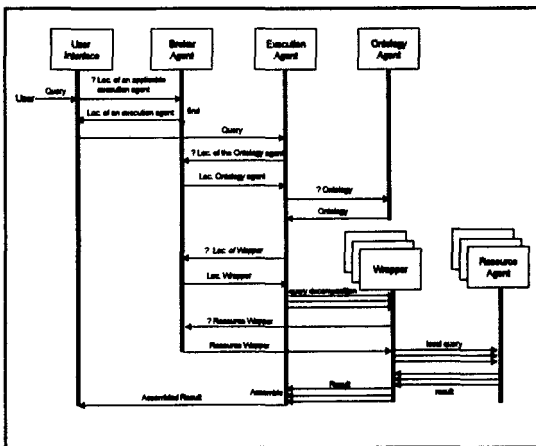
- 미디어이터는 지식을 기반으로 데이터베이스의 스키마 정보를 가지고 있으며 규칙(rule)을 통하여 서로 다른 데이터베이스의 내용을 통합할 수 있다. 미디어이터는 실행 에이전트, 온톨로지 에이전트, 브로커 에이전트, 자원 에이전트로 구성된다.
- 실행 에이전트(execution agent) : 전역 질의 처리기 및 분산 트랜잭션 관리기로 구성되며, 사용자 질의를 처리하고 전반적인 트랜잭션 관리를 한다.
- 온톨로지 에이전트(ontology agent) : 지역 데이터베이스의 서로 다른 스키마 정보를 전역 데이터베이스 스키마로 매핑하는 역할을 담당한다.
- 브로커 에이전트(broker agent) : 정보 자원에 대한 위치 정보를 파악하고 있다
- 자원 에이전트(resource agent) : 각 지역 데이터베이스에 대한 인터페이스 역할을 담당한다. 예를 들어, STEP 데이터에 대한 자원 에이전트는 SDAI 함수로 구성되어 있으며, 사용자는 자원 에이전트에서 제공하는 인터페이스로 복잡한 CAD 데이터에 대한 접근을 쉽게 할 수 있다.
- 래퍼(wrapper)는 멀티 데이터베이스 또는 일반 DMBS와 CORBA와 연동이 가능하도록 해 주는 모듈이다. 래퍼는 또한 전역 질의를 자원 에이전트가 사

- 멀티 데이터베이스는 유사한 정보들의 통합을 위한 것으로서, 이질적이지 않은 정보의 통합을 담당하게 된다. 그림 4의 MDS는 가격 정보와 제품 설명 데이터베이스에 대한 통합을 하고 있으며 이와 같은 데이터베이스는 유사한 스키마 구조를 가지고 있고 문자 숫자(alpha-numeric) 데이터를 포함하고 있다.

용할 수 있는 질의로 변환하여 주며, 질의 결과를 전역 질의 처리기에 전달한다.

3.4 질의 수행 과정

본 논문에서 제안하고 있는 HyDIM 에 따른 질의 시나리오는 그림 5 와 같다.



<그림 5> HyDIM 의 질의 수행 과정

- 사용자 인터페이스는 웹 브라우저를 통한 폼 형태가 되며, 사용자는 찾기 원하는 키워드를 입력하게 된다.
- 웹 서버의 CGI 프로그램은 사용자의 키워드를 받아서 브로커 에이전트에게 실행 에이전트에 대한 위치를 물어보게 된다. 브로커 에이전트는 그 결과를 웹 서버의 CGI 프로그램에게 전달한다.
- CGI 프로그램은 브로커 에이전트로부터

터 받은 위치 정보를 참조하여 실행 에이전트에게 사용자의 질의를 그대로 전달하게 된다. 실행 에이전트는 사용자 질의를 래퍼에게 전달하기 위하여 일단 온톨로지 에이전트에게 사용자 질의, 즉 키워드를 전달하게 된다.

- 온톨로지 에이전트는 사용자 질의를 각각의 지역 데이터베이스에서 어떤 식으로 사용하고 있는지 알고 있기 때문에 지역 데이터베이스의 context 에 맞게 변환하여 실행 에이전트에 전달한다. 만약 사용자 질의를 온톨로지로 변환할 때 대상 온톨로지가 여러 개 있게 되면 사용자로 하여금 온톨로지를 선택하게 한다. 예를 들어 사용자 질의가 “gear” 이라면 그 gear 가 각각의 지역 데이터베이스에서는 어떠한 방법으로 식별하는지 변환(BOM에서는 “gear”, image 데이터베이스에서는 gear 의 part number, 기술문서 데이터베이스에서는 “gears”로 변환) 하는 것이다. 그리고 그 gear 가 자동차에 사용되는 기어, 비행기에 사용되는 기어 등과 같이 여러 개 있다면, 사용자로 하여금 해당 온톨로지를 선택하게 한다.
- 실행 에이전트는 질의를 수행하기 전에 래퍼의 위치를 브로커 에이전트에 물어보고, 그 결과를 참조하여 해당 래퍼에게 사용자 질의를 변형하여 전달한다.

- 각각의 래퍼는 질의를 수행한 후, 그 결과를 실행 에이전트에 반환한다.
- 실행 에이전트는 모든 래퍼로부터 결과를 받아 취합하여 사용자 인터페이스로 전달한다.

이와 같은 질의 시나리오에 따라 단계별로 사용자 질의가 수행되게 된다.

3.5 구현

현재 하이브리드 데이터베이스 통합을 위한 구현은 다음과 같은 서로 다른 4개의 데이터베이스를 구성하였다. 이는 제품 데이터에 대한 관리를 위한 모형으로 제품의 가격 및 설명, 도면 및 매뉴얼을 각각 저장한다. 각각의 데이터베이스에 대한 DBMS는 다음과 같다.

- 가격 데이터베이스 : Oracle
- 제품 설명 데이터베이스 : UniSQL/X
- 제품 도면 데이터베이스 (STEP) : Oracle
- 기술 문서 데이터베이스 (XML) : Fulcrum

CAD와 같은 데이터가 저장된 도면 데이터베이스는 멀티미디어 데이터인 STEP 데이터를 저장하고 있으며, 매뉴얼 데이터베이스는 XML로 기술된 제품에 대한 전

자 문서가 저장된다. 현재 구현을 위하여 각각 Oracle과 Fulcrum을 사용하였으며 다른 DBMS의 사용도 가능하다.

가격 데이터베이스와 제품 설명 데이터베이스의 통합을 위한 멀티 데이터베이스 시스템으로는 UniSQL/M을 사용하였다. 또한 프로그래밍 환경으로는 모두 Java 언어[Java, 1998]를 사용하여 프로그램 하였으며, ORB는 현재 자바 바인딩(binding)이 가능한 OrbixWeb 3.0[OrbixWeb, 1998]을 이용하였다. 미디에이터는 자바 에이전트를 이용하여 구현하였다[Joseph, 1997].

이와 같은 하이브리드 데이터베이스 통합모델에 따라 유사한 스키마를 가진 데이터베이스의 통합은 멀티 데이터베이스를 사용하며, 서로 다른 스키마 정보를 가지고 있는 데이터베이스의 통합은 미디에이터를 통하게 된다.

4. 관련 연구

지금까지 분산되어 있고 이질적인 데이터베이스의 통합에 대한 연구는 지속적으로 진행되어 오고 있다. 그러한 연구는 멀티 데이터베이스의 경우에 있어서는 기능적인 면에 치우쳐 왔다. 즉, 멀티 데이터베이스에서 가져야 하는 기능인 위치 투명성 및 접근 투명성의 보장, 분산 트랜잭션과 동

시 수행의 지원, 구조 및 데이터 충돌 문제의 해결, 분산 데이터베이스의 데이터 모델링, 멀티 데이터베이스의 질의 언어 등 특정 데이터를 가지고 특정 DBMS에 대하여 통합하는 실제로 사용하기 위한 시제품 이전에 연구 중심의 결과들이었다[Weiyi, 1995; Calton, 1991; Yuri, 1995].

따라서 이러한 연구 결과는 적은 양의 데이터에는 비교적 좋은 성과가 있었으나 현실세계의 분산되어 있는 데이터에 대하여 다양한 요구사항이 모두 반영되기는 어려웠다.

더욱이 멀티 데이터베이스 연구 분야 중에서 멀티미디어 데이터의 통합 방법에 대한 연구는 전혀 이루어지지 않았다. 따라서 멀티 데이터베이스를 이용하는 응용 분야는 그래픽, 이미지, 텍스트, 오디오등과 같은 대용량의 멀티 미디어 분야의 응용보다는 단순한 문자숫자 데이터에 대한 통합을 이용하는 분야에 사용되었다.

현재까지 멀티 데이터베이스를 이용한 데이터 통합 방법을 구현한 시스템으로 Papyrus, UniSQL/M, MIND 를 소개하겠다.

Pegasus[Rafi, 1991]는 다양한 데이터 모델을 통합하는 정보 시스템으로써 데이터 모델로는 Iris 데이터 모델을 채택하였다. 또한 질의 언어로는 SQL을 확장한 HOSQL을 이용하였으며, producer type, unifying type 의

가상 클래스를 사용하여 의미 충돌, 이름 충돌, OID 충돌을 해결한다. 그러나 Pegasus는 트랜잭션 처리에 관한 기능을 제공하지 못하는 단점이 있다.

UniSQL/M[UniSQL, 1994]은 UniSQL/X와 SQL을 기반으로 한 멀티 데이터베이스 시스템으로써 데이터 모델로는 객체-관계 모델을 사용하였다. SQL을 기반으로 구현되어 있기 때문에 스키마 충돌도 SQL 언어로써 해결하며 2 단계 실행 프로토콜을 지원한다.

1990년대 와서는 멀티 데이터베이스에 CORBA를 이용한 방법이 제안되었다. MIND (Metu INteroperable Database management systems)[METU, 1996]는 기존에 존재하는 관계 데이터베이스와 객체지향 데이터베이스의 통합을 목적으로 하는 멀티 데이터베이스로서, DEC의 ObjectBroker와 METU Object Oriented DBMS[MOOD, 1994]와 Oracle을 통합하는 시스템이다. 데이터 모델은 ODMG-93[ODMG, 1993] compliant common data model을 사용하였으며 SQL을 기반으로 하는 전역 질의 언어를 제공하였다. 또한 스키마 충돌은 OMG의 Object Definition Language(ODL)을 기반으로 해결된다. 또한 MIND는 전역 질의 최적화를 통하여 서브 질의에 대한 병렬 처리가 가능하다. MIND 시스템은 분산 트랜잭션의 처리를 위해

CORBA 의 상위에 트랜잭션 매니저를 두었고, 지역 트랜잭션과 전역 트랜잭션으로 구성되는 2 레벨 nested 트랜잭션 모델을 가진다. 그러나 MIND 는 통합할 수 있는 데이터베이스의 수가 적으며, 다른 멀티 데이터베이스와 마찬가지로 멀티미디어 데이터에 대한 처리가 전혀 없다.

미디어이터를 이용한 개발은 1990 년대 중반에 활발하게 이루어지고 있는데 대표적인 시스템으로는 Garlic, IDIMS 등이 있다.

Garlic[Mary, 1997]은 완전한 미디어이터라기보다는 통합된 뷰를 제공하는 미들웨어(middleware) 시스템으로써 은닉화 된 데이터 소스와 미들웨어 간의 중재를 하는 래퍼를 통하여 데이터 통합이 이루어진다. Garlic 의 데이터 모델과 프로그래밍 인터페이스는 ODMG 를 기반으로 한 ODL 로 되어 있다. 그러나 멀티데이터베이스와는 달리 Garlic 은 래퍼가 완벽한 지역 시스템에 대한 인터페이스를 제공하기 때문에 일반적인 데이터베이스의 통합 뿐만 아니라 이미지 데이터베이스나 아주 복잡하게 구성된 멀티미디어 데이터베이스도 통합이 가능하다.

IDIMS(INEEL Data Integration Mediation System)[Bhujanga, 1997]는 다수의 이질 데이터 자원들로부터 검색되는 데이터들의 통

합을 목적으로 하는 미디어이터 방식의 통합 시스템으로써 미디어이터와 래퍼의 도메인(domain)에 대한 공통의 정보 표현 모델로 ODMG 의 확장된 버전인 ODL 을 사용한다. 또한 모든 래퍼와 미디어이터가 공유하여 사용하는 공통의 서비스 인터페이스를 두어 시스템의 확장성을 제공한다. IDIMS 에서는 공통의 질의 표현 방법을 제공하기 위하여 새로운 질의 구조인 Query Exchange Model (QEM)을 제안하였는데 QEM 은 질의를 구조적으로 표현하여, 데이터의 구조 정보를 제공한다. IDIMS 에서는 다수의 데이터 자원들로부터의 데이터 통합을 용이하게 하기 위하여 간단하고, 유연성 있는 데이터 표현 모델인 Object Exchange Model(OEM)을 공통의 데이터 모델로 채택하였다. 그러나 IDIMS 는 이질 데이터들에 대한 통합을 위해 설계되었으나 현재는 객체지향 데이터베이스와 관계형 데이터베이스만을 통합 대상으로 하고 있고, 파일 시스템이나 다른 매체로 구성된 자원에 대해서는 통합을 제공하지 못하고 있다.

이와 같이 데이터베이스 통합에 대한 연구는 많이 이루어지고 있으나 각기 그 적용 대상이 다르며, 통합 범위가 다르기 때문에 특정 응용에 필요한 통합 시스템을 구축하기 매우 어렵다. 또한 PDM 분야에서 필요한 데이터 베이스 통합 기능을 충족하기에 각 시스템이 각기 장단점을 가지

고 있다. 따라서 본 논문은 PDM 에 필요한 데이터를 통합하기에 적합한 모델이라 볼 수 있다.

5. 결론

본 논문은 멀티 데이터베이스 시스템으로 PDM 과 같이 이종의 다양한 미디어 타입의 데이터를 분산된 환경에서 통합할 때 발생하는 문제점에 대하여 지적하고 이러한 문제점을 해결하기 위한 방법으로 HyDIM 을 제안하였다.

HyDIM 은 CORBA, 멀티 데이터베이스 시스템, 미디어이터를 사용하는 대용량 멀티 미디어 데이터에 대한 통합을 위한 모델이며 디지털 도서관, CALS, 인터넷 정보 자원등과 같이 기존에 이미 존재하는 대용량 멀티 미디어 데이터에 대한 통합에 적합하다.

마지막으로 HyDIM 에 대한 향후 연구 계획은

- 현재 에이전트에 많은 기능을 부여하여 보다 인텔리전트한 에이전트로 개선
- 현재의 PDM 에 적용한 미디어이터를 보다 다양한 미디어 타입을 지원하는 범용 미디어이터의 구현
- 보다 다양하고 복잡한 사용자 질의를 처리할 수 있도록 시스템 확장
- STEP 데이터를 VRML 이나 와이어 프레임(wire frame) 형태로 보여줄 수 있도록 사용자 인터페이스 개선

등이다.

참고문헌

- [Bhujanga, 1997] Bhujanga Panchanpagesan, Joshua Hui, Gio Wiederhold, Stephan Erickson, Lynn Dean, "The INEEL Data Integration Mediation System", May 1997,
<http://id.inel.gov/idim/paper.html>
- [Calton, 1991] Calton Pu, Avraham Leff, Shu-Wie F. Chen, "Heterogeneous and Autonomous Transaction Processing", *IEEE Computer* 24(12), pp64,-72, 1991.
- [DOM, 1998] Lauren Wood, Jared Sorensen, Steve Byrne, Robert S. Sutor, Vidur Apparao, Scott Isaacs, Gavin Nicol, Mike Champion, Document Object Model Specification,
<http://www.w3.org/TR/WD-DOM/>
- [Gasoinne, 1995] B. Gascoigne, "PDM : The Essential Technology for Concurrent Engineering", MCB University Press, Volume 2, November, 1995.
<http://www.pdmic.com/articles/artetfce.html>
- [Java, 1998] Sun Microsystems, Inc., The Java™ Development Kit (JDK™), version 1.1.5, 1998,
<http://java.sun.com/products/jdk/>
- [Joseph, 1997] Joseph P., Jennifer Bigus, "Constructing Intelligent Agents with Java™ : A programmer's Guide to Smarter Applications", Wiler Computer Publishing, 1997
- [METU, 1996] Dogac, A., et. Al., "METU Interoperable Database System", Demo Description, *Proc. ACM SIGMOD Intl. Conf. On Management of Data*, June 1996.
<http://www.srdc.metu.edu.tr/mind/>
- [MOOD, 1994] Dogac, A., et. Al., "METU Object Oriented Database System", Demo Description, *Proc. ACM SIGMOD Intl. Conf. On Management of Data*, May 1994.
- [Mary, 1997] Mary Tork Roth, Peter M. Schwarz, "Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources", *VLDB '97*, 266-275, 1997.
<http://www.almaden.ibm.com/cs/garlic/homepage.html>
- [ODMG, 1993] Cattel, R.G.G, "The Object Database Standard : ODMG-93", Morgan Kauffmann Publishers, 1994
- [OMG, 1993] Object Management Group, "The Common Object Request Broker : Architecture and Specification", Revision 1.2, OMG Document Number 93-12-43, Draft, December 1993.
- [OrbixWeb, 1998] IONA Technologies, OrbixWEB™ 3.0, 1998,

<http://www.iona.com/products/internet/orbixweb/>

- [Owen, 1997] Jon Owen, STEP : An Introduction, Information Geometers, 1997
- [PDM, 1997] "Introduction to PDM", <http://www.pdmic.com/intropdm.html>, 1997.
- [Rafi, 1991] Rafi Ahmed, Philippe De Smedt, Weimin Du, William Kent, Mohammad A. Ketabchi, Witold Litwin, Abbas Rafii, Ming-Chien Shan, "The Pegasus Heterogeneous Multidatabase System", *IEEE Computer* 24(12), pp19-27, December 1991.
- [SDAI] The Standard Data Access Interface (SDAI), ISO 10303-22
- [UniSQL, 1994] UniSQL/M Manual, UniSQL Inc., 1994.
- [Weiyi, 1995] Weiyi Meng, Clement Yu, "Query Processing in Multidatabase System", *Modern Database Systems; The Object Model, Interoperability, and Beyond*, pp. 552-572, Addison-Wesley Publishing Company, 1995.
- [Wiederhold, 1992] Gio Wiederhold, "Mediators in the Architecture of Future Information Systems," *The IEEE Computer Magazine*, Mar, 1992.
- [WonKim, 1991] Won Kim, Jungyun Seo, "Classifying Schematic and Data Heterogeneity in Multidatabase Systems", *IEEE Computer*, 24(12), pp.12-18, December 1991.
- [WonKim, 1995a] Won Kim, "Introduction to Part 2: Technology for Interpreting Legacy Databases," *Modern Database Systems; The Object Model, Interoperability, and Beyond*, pp. 515-520, Addison-Wesley Publishing Company, 1995.
- [WonKim, 1995b] Won Kim, Injun Choi, Sunit Gala, Mark Scheevel, "On Resolving Schematic Heterogeneity in Multidatabase Systems", *Modern Database Systems; The Object Model, Interoperability, and Beyond*, pp. 521-550, Addison-Wesley Publishing Company, 1995.
- [XML, 1997] Dan Connolly, XML : Principles, Tools, and Techniques, *World Wide Web Journal*, Volume 2, Number 4, Winter, 1997
- [XML, 1998] Tim Bray and C.M. Sperberg-McQueen, "Extensible Markup Language (XML): Part I. Syntax", World Wide Web Consortium Recommendations, 1998. 2. 10, <http://www.w3.org/TR/REC-xml>
- [Yuri, 1995] Yuri Breitbart, Hector Garcia-Molina, Avi Silberschatz, "Transaction Management in Multidatabase Systems", *Modern Database Systems; The Object Model, Interoperability, and Beyond*, pp. 573-591, Addison-Wesley Publishing Company, 1995.

저자소개

이강찬 (dolphin@comeng.chungnam.ac.kr)

충남대학교 컴퓨터공학과 졸업 (학사, 석사)

현재 충남대학교 컴퓨터공학과 박사과정 재학 중

관심 분야 : 미디어이터, 멀티 데이터베이스, STEP, XML, CORBA, 에이전트

이 상 (slee@comeng.chungnam.ac.kr)

충남대학교 컴퓨터공학과 졸업 (학사)

현재 충남대학교 컴퓨터공학과 석사과정 재학 중

관심 분야 : 미디어이터, CORBA, 에이전트

유정연 (jyyou@comeng.chungnam.ac.kr)

배재대학교 컴퓨터공학과 졸업 (학사)

현재 충남대학교 컴퓨터공학과 석사과정 재학 중

관심 분야 : STEP, CORBA, 멀티 데이터베이스

이규철 (kcllee@comeng.chungnam.ac.kr)

서울대학교 컴퓨터공학과 졸업(학사, 석사, 박사)

미국 Syracuse University, CASE Center 객원 교수

미국 IBM Almaden Research Center 객원 연구원

현재 충남대학교 공과대학 컴퓨터공학과 부교수

관심 분야 : 데이터베이스 통합, 문서 관리 시스템, STEP, SGML, XML, CORBA, 에이전트