

## CORBA 기반의 이질적인 데이터베이스 통합

이주대학교 변광준\*

### 1. 서 론

이질적이고 자율적이며 분산되어 있는 데이터베이스 시스템들 사이에서 각각의 자율성을 최대한 보장하면서 효율적인 정보 공유 및 교환을 가능하게 하는 시스템은 그 동안 데이터베이스 분야의 연방(federated) 데이터베이스 시스템 (또는 멀티데이터베이스 시스템), cooperative information system 등에 관한 연구에서 해결하려는 문제였다. 이러한 통합 시스템의 개발은 대체적으로 네트워크 프로그래밍, 표준화되지 않은 미들웨어 등을 이용해 이질적인 데이터베이스 시스템들 간의 기본적인 연결을 확보하고 이를 기반으로 시스템의 구성 모듈들을 일일이 구현하는 방식을 채택했으며, 많은 경우 객체지향 데이터 모델 및 질의어를 이용하는 등 객체지향 개념에 기반을 두었다. 하지만 구현 작업은 많은 하위 레벨 프로그래밍을 필요로 하는 복잡하고 어려운 과정을 거쳐야 했다.

OMG가 제안한 분산 객체 미들웨어 표준인 CORBA와 이를 기반으로 하는 분산 객체 시스템의 표준 구조인 OMA(Object Management Architecture)[10]는 이질적인 응용 시스템들의 연결을 기존의 방식에 비해 상대적으로 용이하게 할 수 있는 방법을 제공할 뿐만 아니라, 분산 객체 컴퓨팅에 공통적으로 필요할 것으로 예상되는 다양한 서비스들을 제공하고 있다. 따라서, CORBA를 이용해 분산 객체 시스템을 개발할 경우 개념상으로는 구현 작업

의 상당 부분이 간소화되고 손쉽게 이루어지게 되었다. 개발자가 CORBA를 이용한 간단한 상위 레벨 프로그래밍을 통해 이질적인 시스템들을 연결하고 OMA에 정의된 CORBA 기반 서비스들을 가능한 그대로 또는 일부 변경해 이용할 수 있기 때문에, 실질적인 구현은 시스템의 나머지 모듈들에 대해서 이루어진다. 참고로 CORBA를 구현한 제품을 제공하는 회사들은 CORBA 서비스를 구현한 제품들도 제공하고 있다.

CORBA의 이러한 장점은 데이터베이스 분야에도 영향을 끼쳐 이질적인 데이터베이스 시스템들의 통합 문제를 CORBA를 이용한 분산 객체 시스템을 기반으로 해결하려는 연구들을 시도하게 만들었다. 연방 데이터베이스 시스템으로 분류될 수 있는 Middle East Technical University의 MIND 프로젝트[1]와 Dublin City University의 OASIS 프로젝트[2], 연방 데이터베이스 시스템의 제약성을 극복하고 웹에 연결된 수많은 데이터베이스들 사이에서의 정보 공유를 위한 Queensland University of Technology의 WebFINDIT 프로젝트[3, 4], 분산 객체 시스템에 비즈니스 프로세스(business process)들이 통합된 일종의 cooperative information system으로 분류될 수 있는 GTE의 CABS II 프로젝트[5, 6] 등이 데이터베이스 분야에서 보고된 관련 연구들이라 하겠다. 본 기고는 이러한 연구들에서 CORBA 및 CORBA 서비스가 어디에 어떻게 이용되었는지를 알아보는 데에 초점을 맞추고 있으며, 이를 통해 일반적이지는 않지만 데이터베이스 분

\*증신회원

야에서 이질적인 데이터베이스 통합 문제에 CORBA가 이용되고 있는 현황을 설명하고자 한다.

## 2. MIND

터키의 METU(Middle East Technical University)에서 개발한 시스템인 MIND (METU INteroperable Database system)은 CORBA를 기반으로 하는 연방 데이터베이스 시스템이다. MIND는 기존의 연방 데이터베이스 시스템들 대부분이 그렇듯이[7, 8], 이질적인 데이터베이스들의 스키마를 공통 데이터 모델로 변환해 통합하고 통합된 전역(global) 스키마를 이용해 복수의 데이터베이스에 대한 전역 질의를 공통 질의어로 작성해 처리할 수 있는 기능을 제공함으로써 일종의 분산 데이터베이스 시스템을 구축하는 방식을 채택했다. MIND에서는 공통 데이터 모델과 공통 질의어로서 ODMG-93[9]에서 설명하는 객체 데이터 모델과 SQL과 유사한 객체 질의어인 OQL을 구현해 이용했다.

MIND의 시스템 구조는 그림 1에서 보는 바와 같이 LDA(Local Database Agent)와 GDA(Global Database Agent), 그리고 Client

ent로 구성된다. LDA는 각 LD(Local Database)에 대해 하나씩 존재하고, LD가 제공하는 export 스키마를 공통 데이터 모델로 표현된 형태로 가지고 있으며, OQL과 LD의 질의어 사이에서 변환을 담당한다. GDA는 하나 이상 존재할 수 있고 하나 이상의 LDA들을 관리하며, 전역 스키마를 가지고 있고 Global Query Processor와 Global Transaction Manager를 포함한다. Client는 GDA가 제공하는 동일한 인터페이스를 통해 이질적인 데이터베이스 시스템(LD)들에 대한 질의를 요청한다. CORBA는 Client와 GDA 사이에서, 그리고 GDA와 LDA 사이에서 상호 작용을 가능하게 함으로써, 결국 이질적인 데이터베이스 시스템들간의 연결을 제공한다.

MIND의 구현은 현재 관계형 DBMS인 Sybase와 Oracle, METU에서 개발한 객체지향 DBMS인 MOOD(METU OODBMS)가 CORBA 제품인 DEC사의 ObjectBroker로 연결된 환경을 기반으로 이루어졌다. 구현 작업에서 ObjectBroker는 Client, GDA, LDA 사이에서의 연결에 이용되었으나 ObjectBroker가 제공하는 CORBA 서비스가 GDA 및 LDA를 구성하는 주요 모듈들(예를 들면, GDA의 Global Transaction Manager)을 구

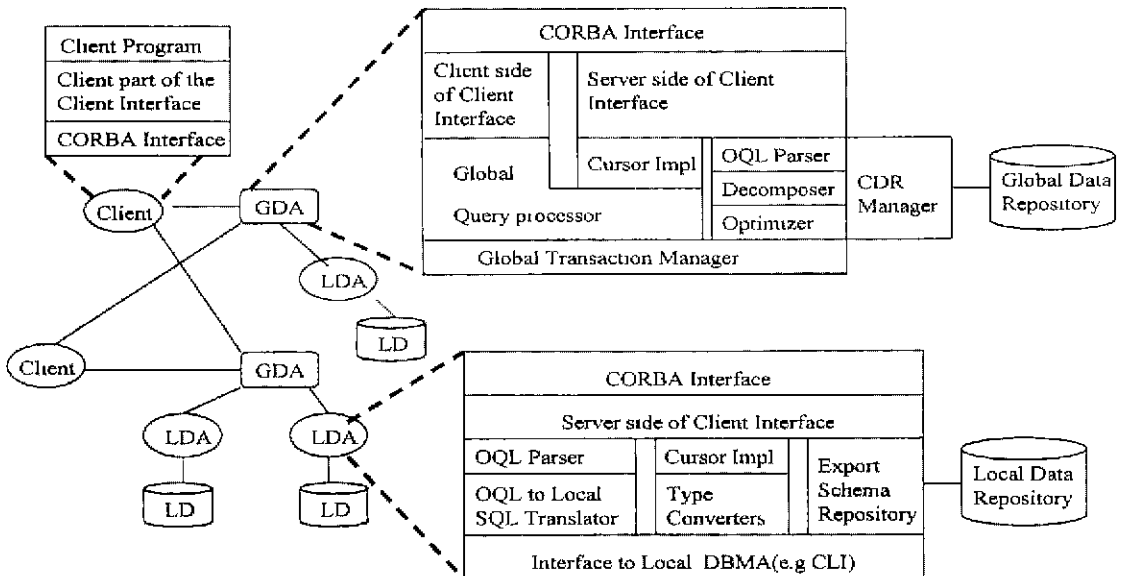


그림 1 MIND의 시스템 구조[1]

현하는 데는 이용되지 못했다.

### 3. OASIS

아일랜드의 Dublin City University에서 진행 중인 시스템인 OASIS(ODMG Architecture for Specification of Interoperable Systems) 역시 CORBA를 기반으로 하는 연방 데이터베이스 시스템이다. OASIS는 MIND와 마찬가지로 전역 스키마 방식을 채택하고 있으며, 공통 데이터 모델로는 ODMG-93의 객체지향 데이터 모델을, 공통 질의어로는 ODMG-93의 OQL을 연방 데이터베이스에 적용할 수 있게 일부 변경한 MOQL을 구현해 이용한다.

OASIS의 시스템 구조는 그림 2에서 보는 바와 같이 Local Mediator와 MDB Server로 구성된다. Local Mediator는 MIND의 LDA와 거의 동일한 역할을 담당하지만, LDA와는 달리 해당 데이터베이스의 전체 스키마를 공통 데이터 모델로 표현한 component 스키마와 이 스키마로부터 정의된 하나 이상의 export 스키마들을 가지고 있다. MDB Server는 MIND의 GDA와 동일한 역할을 담당하지만, GDA와는 달리 현재 상태의 OASIS에서는 하나의 MDB Server가 존재한다. 그림 2에서 보이지는 않지만 MIND의 Client와 동일한 역할을 하는 일종의 클라이언트 프로그램이 MDB

Server를 이용한다. CORBA는 MIND에서와 유사하게 클라이언트 프로그램과 MDB Server 사이에서, MDB Server와 Local Mediator 사이에서 상호 작용을 가능하게 한다.

OASIS는 진행 중인 프로젝트이며 현재 상태의 구현은 그림 2에서 보는 바와 같이 객체지향 DBMS인 Versant, 관계형 DBMS, legacy 시스템이 CORBA 제품인 Iona사의 Orbix로 연결된 환경에서 이루어졌고, Orbix는 MIND의 경우에서와 마찬가지로 MDB Server, Local Mediator, 클라이언트 프로그램 사이에서의 연결에 대해서만 주로 이용되고 있다.

### 4. WebFINDIT

호주의 Queensland University of Technology에서 개발한 시스템인 WebFINDIT은 웹에 연결된 데이터베이스들 사이에서 효율적이고 용이한 정보 공유를 가능하게 하기 위한 실험적인 시스템이다. 기존의 연방 데이터베이스 시스템들이 기반으로 하고 있는 스키마 통합은 데이터베이스의 수가 단지 몇 개에 불과한 경우에도 상당히 어려우며 데이터베이스의 수가 많아지면 불가능하게 되기 때문에, 수많은 웹 데이터베이스들간의 정보 공유에 적용하기에는 어려움이 있다. WebFINDIT은 이에 착안해 사용자가 관련있는 웹 데이터베이스들을 Coalition이라는 단위로 동적으로 그룹화하고 Coalition간에 또는 Coalition과 데이터베이스간 서비스(Service Link로 표현)를 역시 동적으로 정의할 수 있게 하며, 이러한 Coalition 및 Service Link를 이용해 많은 수의 웹 데이터베이스들을 효과적으로 이용할 수 있게 한다.

WebFINDIT의 구조는 그림 3에서 보는 바와 같이 Browser, Query Processor, Co-database Server, Broker, Information Source Interface로 구성된다. Browser는 사용자가 WebFINDIT을 이용할 수 있게 하는 인터페이스 역할을 하며, 사용자로 하여금 현재 정의되어 있는 Coalition 및 Service Link에 대한 정보를 Co-database로부터 얻고 이를 바탕으로 특정 웹 데이터베이스(그림 3에서

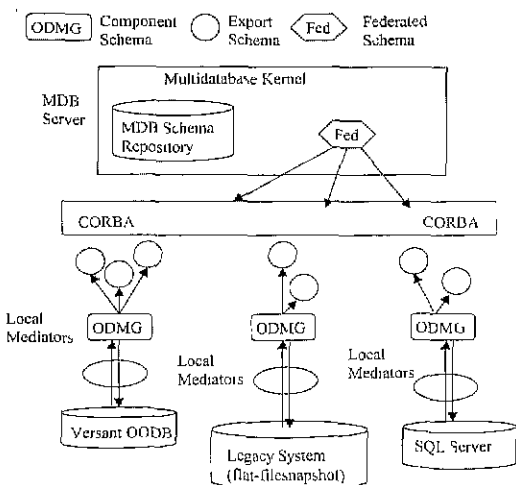


그림 2 OASIS의 시스템 구조[2]

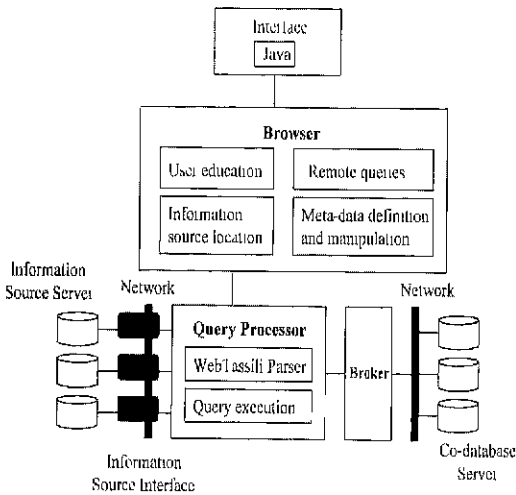


그림 3 WebFINDIT의 시스템 구조[3]

는 Information Source)를 찾아내서 질의를 하고 그 결과를 확인할 수 있도록 한다. 또한, 사용자가 Coalition 및 Service Link를 동적으로 정의해 Co-database에 저장할 수 있도록 한다. Query Processor는 사용자로부터 Co-database에 대한 요청 및 Information Source에 대한 요청을 받아, 전자는 Broker를 통해 Co-database Server를 접근하도록 해 주고 후자는 Information Source Interface를 통해 Information Source Server를 접근하도록 해 준다. Co-database Server는 Coalition과 여기에 속한 Information Source들, 그리고 Service Link 등에 대한 메타 데이터를 저장한다. Broker는 Query Processor와 Co-database Server간의 연결을, Information Source Interface는 Query Processor와 Information Source Server간의 연결을 담당한다.

WebFINDIT의 프로토타입은 Information Source Server로는 객체지향 DBMS인 ONTOS와 ObjectStore, 관계형 DBMS인 Oracle, 객체 관계형 DBMS인 UniSQL이 이용되고 Information Source Interface는 JDBC 및 ODBC를 이용해 구현되었고, Co-database Server로는 ObjectStore가 이용되고 Broker는 SUN사의 CORBA 제품인 NEO를 이용해 구현되었으며, Browser와 Query

Processor 등 기타 구성 모듈들은 Java로 구현되었다. 실제 구현 작업에서 NEO는 MIND 및 OASIS의 경우에서와는 달리 WebFINDIT의 두 구성 모듈인 Query Processor와 Co-database Server간의 연결에만 이용되었다.

### 5. CABS II

미국의 GTE에서 Billing 업무를 지원하기 위해 개발한 시스템인 CABS II(Carrier Access Billing System II)는 이질적이고 자율적인 정보 시스템들이 협력(cooperation)을 통해 공통의 목표들을 성취할 수 있는 cooperative information system의 일종이다. CABS II는 OMG의 OMA에 기반을 두고 여기에 비즈니스 프로세스들이 통합된 형태를 취하고 있다. 그림 4에서 Object Request Broker와 이에 연결된 Object Services, Common Facilities, Application Objects가 OMA 형태를 따르며, 위에 보이는 그래프는 비즈니스 프로세스를 나타내고 있다.

CABS II가 OMA에 기초하기 때문에 구현은 당연히 CORBA 제품 및 CORBA 서비스 제품들을 최대한 활용하는 방식으로 시작되었다. 하지만 여러 대형(large-scale) 분산 객체 시스템에 성공적으로 이용되었기 때문에 CABS II에 채택된 TCSI사의 Object Services

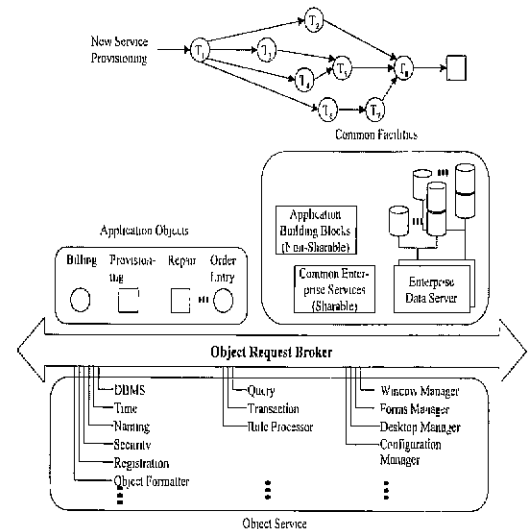


그림 4 CABS II의 개념적인 시스템 구조[5]

Package도 일부 서비스들만을 제공하는 관계로 실제 구현 작업에서는 많은 서비스들을 새롭게 구현해야 했다.

## 6. 결 론

OMG의 표준인 CORBA와 OMA는 이질적인 응용 시스템들간의 용이한 연결을 가능하게 하고 분산 객체 컴퓨팅에 필요한 다양한 서비스를 제공한다. 따라서, 이를 이용해 분산 객체 시스템을 개발할 경우 CORBA 서비스들을 일부 변경하거나 CORBA가 제공하지 않는 모듈들만 구현하면 되기 때문에 구현 작업의 부담이 상당 부분 덜어지게 된다. 이와 같은 장점으로 인해 CORBA는 데이터베이스 분야에서 이질적인 데이터베이스 통합 문제의 해결에 적용되는 것이 고려되었고 CORBA를 이용한 몇몇 실험적인 시스템들이 보고되었으며, 본 기고에서는 이러한 시스템들에 대해서 알아보았다.

CORBA 및 OMA는 표준안 상태로는 분산 객체 컴퓨팅을 위한 다양한 서비스들을 제공하는 것으로 되어 있으나, 표준안을 실제로 구현한 제품들은 이질적인 시스템들을 연결하는 CORBA의 본래 기능은 모두 제공하지만 OMA에 정의된 CORBA 서비스들은 극히 일부 서비스들을 제외하고는 아직까지 제공하지 못하고 있는 실정이다. 이로 인해 본 기고에서 알아본 데이터베이스 분야의 시스템들에서도 CORBA는 주로 이질적인 데이터베이스 시스템들의 연결을 손쉽게 구축하기 위해 사용되었고 각 시스템을 구성하는 모듈들은 CORBA 서비스와 관계없이 별도로 구현되었다.

현재의 CORBA 제품 상태로는 일반적인 분산 객체 시스템은 물론 데이터베이스 분야의 이질적인 데이터베이스 통합 시스템에 보편적으로 활발히 적용되기에 어려움이 있는 실정이고, 전문 지식을 가지고 있는 유능한 개발자들을 보유하고 있는 경우에만 효율적인 시스템 개발이 이루어질 수 있는 실정이다. 하지만 CORBA 및 OMA의 표준안 중 미비한 부분에 대해서는 보완이 진행되고 확정된 부분에서 CORBA 제품에 구현되지 않은 부분은 제품별

로 구현이 진행중이며, CORBA를 기반으로 하는 시스템의 개발, 운용, 관리를 쉽게 할 수 있는 도구(tool)에 대한 개발도 진행되고 있다. 이러한 개발 노력이 결실을 거두어 다양한 서비스들을 제공하는 CORBA 제품들과 관련 도구들이 나오면, CORBA가 가지는 개념상의 상대적인 장점으로 인해 CORBA를 기반으로 하는 다양한 형태의 이질적인 데이터베이스 통합 시스템들이 개발될 것으로 예상된다.

## 참고문헌

- [1] Dogac, A., et. al., "METU Object-Oriented Database System", SIGMOD Record, Vol. 24, No. 3, Sept. 1995.
- [2] Roantree, M., Murphy, J., and Hassebring W., "The OASIS Multidatabase Prototype", SIGMOD Record, Vol. 28, No. 1, Mar. 1999.
- [3] Benatallah, B. and Bouguettaya, A., "Data Sharing on the Web", International Conference on Enterprise Distributed Object Computing, Oct. 1997.
- [4] Bouguettaya, A., et. al., "Using Java and CORBA for Implementing Internet Databases", IEEE International Conference on Data Engineering, Mar. 1999.
- [5] Brodie, M.L., "The Emperor's Clothes are Object Oriented and Distributed", in Cooperative Information Systems Trends and Directions, Papazoglou, M.P. and Schlageter, G., Eds., Academic Press, 1998.
- [6] Manola, F., et. al., "Supporting Cooperation in Enterprise-Scale Distributed Object Systems", in Cooperative Information Systems Trends and Directions, Papazoglou, M.P. and Schlageter, G., Eds., Academic Press, 1998.
- [7] Sheth, A. and Larson, J., "Federated Database Systems for Managing Distributed, Heterogeneous and Autono-

mous Databases”, ACM Computing Surveys, Vol. 22, No. 3, 1990.

- [8] Bukhres, O. and Elmagarmid, A., Eds., Object-Oriented Multidatabase Systems, Prentice Hall, 1996.
- [9] Cattell, R. and Barry, D., Eds., The Object Database Standard: ODMG 2.0, Morgan Kaufmann, 1997.
- [10] Orfali, R., Harkey, D., and Edwards, J., The Essential CORBA: System Integration Using Distributed Objects, Wiley, 1996.



**변 광 준**

1985 서울대학교 컴퓨터공학(학사)  
 1987 Pennsylvania State University 전산학(석사)  
 1993 University of Southern California 전산학(박사)  
 1994~현재 아주대학교 정보 및 컴퓨터공학부 부교수  
 관심분야: 객체지향 데이터베이스, 연립 멀티미디어 데이터베이스, cooperative information system, 분산 객체 컴퓨팅

E-mail : byeon@madang.ajou.ac.kr

**'99 정례회의 및 편집위원회 연간일정표**

월 별	정 려 회 의		편 집 위 원 회	
	상임이사회	정레이사회	논 문 지	학 회 지
1월	8일(금) 17:00		29일(금) 16:00, 전체회의	20일(금) 16:30
2월	5일(금) 16:00	26일(금) 18:00	26일(금) 16:00	19일(금) 16:30
3월	5일(금) 16:00		26일(금) 16:00, 전체회의	19일(금) 16:30
4월	9일(금) 16:00	16일(금) 17:00	30일(금) 16:00	23일(금) 19:00
5월	7일(금) 16:00		28일(금) 16:00, 전체회의	21일(금) 16:30
6월	11일(금) 16:00	25일(금) 17:00		18일(금) 16:30
7월	9일(금) 16:00		2일(금) 16:00 30일(금) 16:00, 전체회의	16일(금) 16:30
8월	6일(금) 16:00	10일(금) 17:00	27일(금) 16:00	20일(금) 16:30
9월	3일(금) 16:00		17일(금) 16:00, 전체회의	17일(금) 16:30
10월	8일(금) 16:00	15일(금) 17:00	29일(금) 16:00	22일(금) 19:00
11월	5일(금) 16:00		26일(금) 16:00, 전체회의	19일(금) 16:30
12월	3일(금) 16:00	17일(금) 17:00	24일(금) 16:00	17일(금) 16:30

※ 회의일정은 사정에 따라 변경될 수 있음.