



CALS에서의 통합 데이터베이스 시스템

변광준

이주대학교 정보 및 컴퓨터공학부 교수

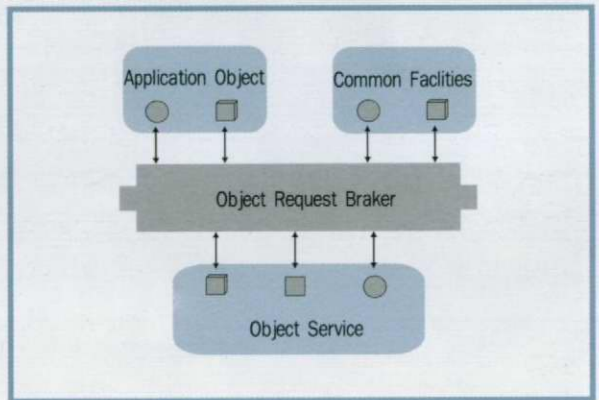
최근 국내에서 관련 활동이 활발하게 진행되고 있는 CALS는 여러 측면에서 데이터베이스 분야와 밀접한 관련이 있다. 그 중에서도 CALS의 구현에 있어 핵심 요소인 통합 데이터베이스 시스템은 이러한 관련성을 단적으로 보여준다 하겠다. 이 글에서는 CALS의 통합 데이터베이스 시스템에 대해 정의, 특징, 구현을 위한 요소 기술, 구현 사례 등을 중심으로 알아보기로 한다.

1. 통합 데이터베이스 시스템의 정의 및 특징

통합 데이터베이스 시스템은 <그림 1>에서 보는 바와 같이 기업에서 필요한 정보를 효율적이고 통합적으로 공유 및 관리하기 위한 체제라 할 수 있으며, 다음과 같은 특징을 가지고 있다.

- 통합 데이터베이스 시스템은 기업내의 현재의 비즈니스 프로세서들사이에서 이들을 위해 구축되는 것이 아니라, 리엔지니어링(reengineering)을 통해 개선된 비즈니스 프로세서들사이에서 구축되어야 한다. 따라서, 리엔지니어링

<그림 1> 통합 데이터베이스 시스템



이 통합 데이터베이스 시스템의 구축에 반드시 선행되어야 한다.

- 일반적으로 기업내에서는 분산된 지역에서 데이터베이스, 파일, CAD, Product Data Management (PDM) 시스템 등의 이질적인 시스템들을 정보 관리에 사용하고 있으며, 이들은 자율적으로 제어되면서 관련 응용 프로그램들을 지원하고 있다. 따라서, 통합 데이터베이스 시스템은 그 동안의 투자를 낭비하지 않기 위해 현재 정보 관리를 담당하는 시스템들을 바탕으로 구축되어야 하며, 이때 각

시스템의 자율성은 최대한 보장하면서 동시에 이질적인 시스템들 사이에서 정보의 공유 및 관리가 이루어질 수 있도록 구축되어야 한다.

- 통합 데이터베이스 시스템은 기존 시스템들의 통합뿐만 아니라 향후 새로운 정보 관리 시스템들의 통합이 용이하도록 확장성을 가지고 있어야 한다. 최근의 컴퓨터 관련 기술의 발전 속도로 볼 때 새로운 시스템의 향후 도입은 충분히 예상할 수 있는 상황이므로 통합 데이터베이스 시스템은 이를 대비해 구축되어야 한다.
- 기업에서 필요한 정보는 일반적인 텍스트 정보 뿐만 아니라 이미지, 오디오, 비디오 등의 멀티미디어 정보도 포함하고 있으므로, 이들의 교환 및 관리가 용이해야 한다. 또한, CALS에서는 정보의 표준화를 매우 중요시 여기며, 정보의 종류에 따라서 표준을 지정해 놓고 있다. 예를 들면, SGML은 문서 정보의 표준이고, STEP은 제품(product) 정보의 표준이다. 따라서, 통합 데이터베이스 시스템은 이러한 표준들을 이해하고 이에 따라 표현된 정보를 관리할 수 있어야 한다.

2. 요소 기술

통합 데이터베이스 시스템의 구현과 가장 밀접한 관련이 있는 것으로 언급되고 있는 기술로는 연방(federated) 데이터베이스 또는 멀티데이터베이스(multidatabase) 시스템 기술과 분산 객체 관리 기술이 있다.

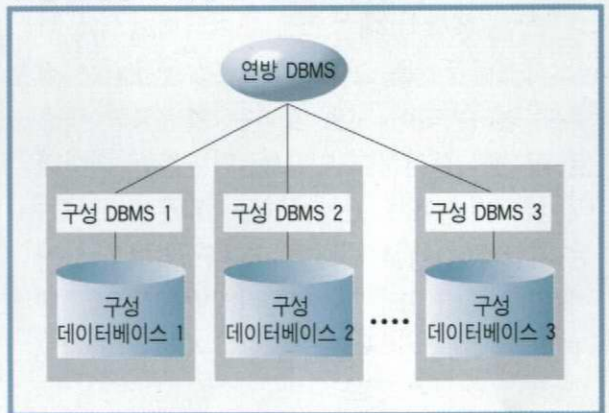
2.1. 연방 데이터베이스 시스템 기술

연방 데이터베이스 시스템 기술은 이질적인 데이터베이스 및 파일 시스템들 사이에서 각 시

스템의 자율성을 최대한 보장하면서 정보의 교환 및 공유를 가능하게 해 주는 일종의 분산 데이터베이스 시스템의 구축에 중점을 두고 있다. 이 기술은 그 동안 데이터베이스 분야에서 발전되어 왔으며, 많은 연구 개발을 통해 다양한 시스템들이 제시되었고 최근에는 UniSQL/M이라는 상용화된 제품이 소개되기도 하였다. 이들은 <그림 2>에서 보는 바와 같이 이질적인 구성(component) 데이터베이스 시스템들과 이들을 제어하는 연방 DBMS로 이루어진 공통적인 구조를 가지고 있다.

구성 데이터베이스 시스템은 기존에 사용하고 있던 데이터베이스 시스템에서 DBMS를 연방 DBMS와의 상호작용이 가능하도록 확장한 형태를 취하고 있다. 구성 데이터베이스 시스템들이 일반적으로 이질적인 데이터 모델과 데이터베이스 언어를 사용하므로, 연방 DBMS는 이들을 모두 표현하고 이해할 수 있는 공통 데이터 모델과 데이터베이스 언어를 사용하며, 대부분의 경우 다른 모델들에 비해 모델링 파워가 크고 확장성이 뛰어난 객체지향 모델과 언어를 사용한다. 연방 데이터베이스 시스템에서는 구성 데이터베이스 시스템들의 자율성이 지켜지므로, 지역(local) 질의와 전역(global) 질의의 두 종류의 질의가 존재한다.

<그림 2> 연방 데이터베이스 시스템의 구조





지역 질의는 하나의 구성 데이터베이스 시스템에만 관련된 질의로서 해당 DBMS에 의해 처리되고, 전역 질의는 여러 구성 데이터베이스 시스템들에 동시에 관련된 질의로서 연방 DBMS와 해당 구성 DBMS들과의 상호 작용에 의해 처리된다.

지금까지 제시된 연방 데이터베이스 시스템들은 이와 같은 공통점외에 각기 고유한 특징을 가지고 있고, 이를 기초로 몇 개의 그룹으로 구분될 수 있다. Sheth와 Larson이 1990년 ACM Computing Surveys에 발표한 구분 방법에 따르면 연방 데이터베이스 시스템은 연방의 연결 정도에 따라 tightly-coupled와 loosely-coupled 방식으로 구분되고, tightly-coupled는 다시 연방의 갯수에 따라 단일 연방 (single federation) 과 복수 연방 (multiple federation) 방식으로 구분된다.

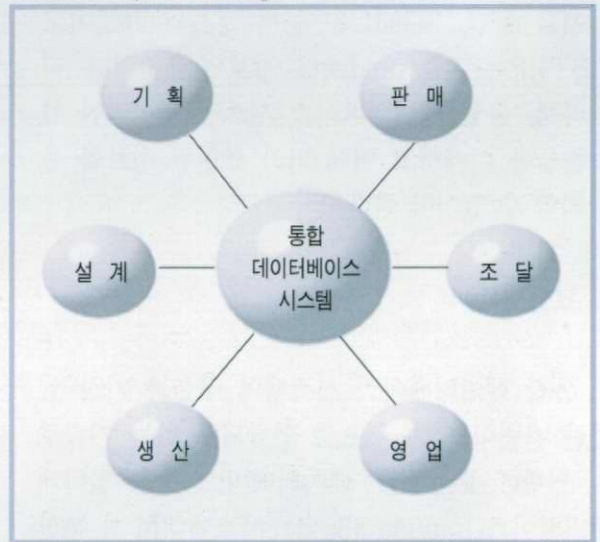
각 방식에 대한 설명은 이 글의 성격상 생략하겠고 자세한 내용은 다음의 논문을 참조하기 바란다. (A. Sheth and J. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", ACM Computing Surveys, Vol. 22, No. 3., 1990.)

- 하지만 각 방식은 기능상 제약성을 가지고 있어 부분적인 해결책만을 제시하고 있을 뿐이다. 또한, 통합의 대상으로 데이터베이스와 파일 시스템들만을 고려하고 있고, 공유되는 정보도 멀티미디어 정보 보다는 일반적인 텍스트 정보에 한정하고 있으며, 일반적인 데이터베이스 기술이기 때문에 CALS에서 요구하는 표준화된 정보에 대한 고려도 미비한 실정 이므로, CALS의 통합 데이터베이스 시스템 구현에 그대로 적용하기에는 문제점을 가지고 있다.

2.2. 분산 객체 관리 기술

분산 객체 관리 기술은 데이터베이스 시스템을 포함하는 일반적인 응용 시스템들을 대상으로 하고, 이들 사이에 존재하는 이질성을 극복하고 정보 교환이 가능하게 해 주는 체제의 개발에 중점을 두고 있다. 이 기술에 대해서는 그 동안 많은 연구 개발이 진행되어 왔으나, CALS의 통합 데이터베이스 시스템에 관련되어서는 OMG가 제안한 표준안들인 Object Management Architecture (OMA), Common Object Request Broker Architecture (CORBA), Core 객체 모델이 자주 언급되고 있다. OMA는 <그림 3>에서와 같이 Object Request Broker(ORB), Object Services, Common Facilities, Application Objects로 이루어지며, 각각의 특징은 다음과 같다.

<그림 3> Object Management Architecture (OMA)

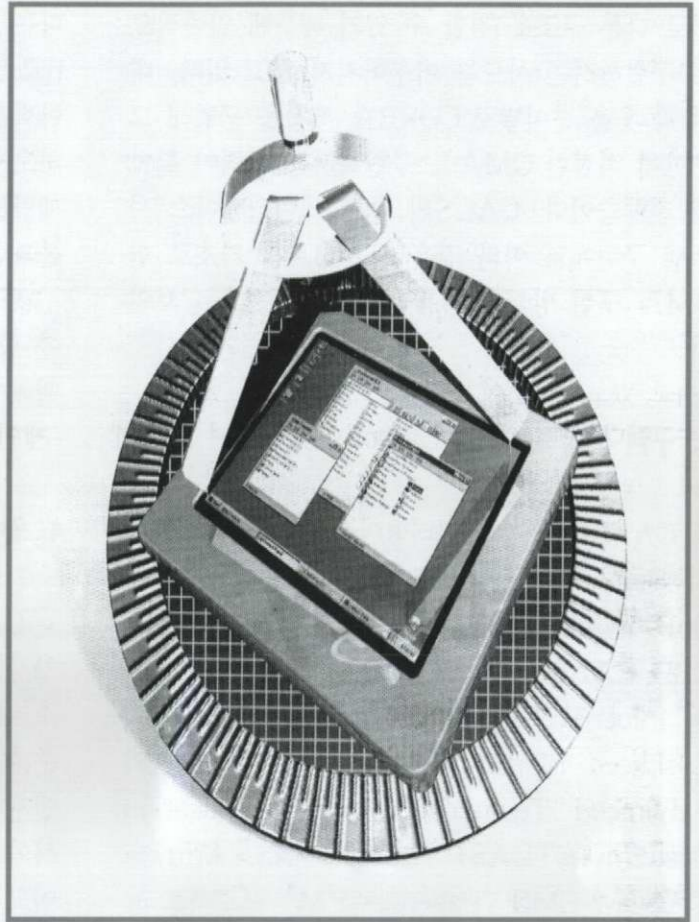


- ORB는 분산 환경에서 객체들이 위치 및 시스템 등에 투명하게 서비스에 대한 request를 보내고 이에 대한 response를 받을 수 있게 해 주는 체제이다. 따라서, ORB는 서로 다른 환경에 있는 이질적인 시스템들사이에서의 상호

작용성(interoperability)을 제공함으로써, 이들사이의 유연한 연결을 가능하게 해 준다. OMG는 이러한 ORB가 갖추어야 할 요건들을 담은 specification을 발표해 오고 있으며, 이 specification이 Common Object Request Broker Architecture라고 이름이 붙여졌기 때문에 ORB는 흔히 CORBA라고 불린다. 이 CORBA는 그 동안 계속 보강되어 현재에는 revision 2.0에 이르고 있다.

- Object Services는 ORB를 보강하여 분산 객체 관리의 근간을 이룬다. ORB가 객체의 invocation을 위한 최소의 기능만을 포함하기 때문에, 분산 객체 관리를 위해 공통적으로 필요한 다른 기능들은 모두 Object Services에 포함되어 있다. 이러한 기능들에는 naming, events, persistence, relationships, concurrency, transactions, security, versions, 질의 등을 처리 및 관리하는 기능들이 속한다.
- Common Facilities와 Application Objects는 ORB와 Object Services로 구성된 시스템을 이용하는 객체들로 간주할 수 있으며, Application Objects는 application들을 나타내고, Common Facilities는 출력, 문서 관리, 전자우편 기능같이 여러 응용 분야들에 공통적으로 필요한 기능들을 나타내며 이들은 특정 응용 분야에 맞게 변경 및 조절될 수도 있다.

OMG는 OMA와 CORBA 외에도 이들이 다루는 객체를 정의하는 Core 객체 모델을 제시했다. 이 모델은 거의 모든 객체지향 모델에서 볼 수 있는 객체, 오퍼레이션, 타입 및 서브타입 등



의 기본적인 개념들만을 제공한다.

하지만, 이 모델은 특정 application을 위한 개념들을 포함시켜 확장될 수 있으며, 예를 들어 Object Database Management Group (ODMG)에서 정한 객체지향 데이터베이스의 데이터 모델 표준인 ODMG-93은 OMG의 Core 객체 모델에 애트리뷰트, 트랜잭션 등의 데이터베이스 관련 개념들을 포함시켜 확장한 형태를 취하고 있다.

OMG가 제안한 표준안들중에서 Core 객체 모델에 기초한 CORBA의 구현은 현재 활발하게 진행되고 있으며, Iona의 Orbix, DEC의 Object Broker, Expersoft의 PowerBroker, IBM의 SOM, HP의 ORB+, Sun의 DOE/DOMF가 구현된 제품들중의 일부이다. OMA는 응용 분야



별로 다른 구조를 취할 수 있기 때문에 일반적으로 구현하려는 시도는 이루어지지 않고 있다. 대신 각 응용 분야별로 CORBA 제품을 구입해 그 분야에 적절한 OMA를 구현하려는 노력이 활발히 진행 중이다. CALS의 통합 데이터베이스 시스템 구현에도 이와 같은 CORBA를 기초로 한 OMA 구현 방식을 적용할 수 있을 것으로 보인다.

3. 구현 사례

CALS의 통합 데이터베이스 시스템을 구현한 사례는 주로 미국에서 보고되고 있다. 예를 들면, DoD의 Joint CALS (JCALS) 프로젝트, OSD CALS IWSDDB Project Task 2 Team의 Pilot IWSDDB Demonstration 프로젝트, Lockheed Martin Aeronautical System사의 Advanced Technical/Logistic Application System (ATLAS) 프로젝트, U.S. Army의 TACOM CMS "Paperless PM" 프로젝트 등에서 프로젝트의 일부로서 통합 데이터베이스 시스템을 구현했다고 보고되었다. 국내에서는 아직 보고된 적이 없으며, 통상 산업부가 선정한 10개의 CALS 시범 업체들중 일부에서 프로젝트의 일부로서 통합 데이터베이스 시스템을 구축하려는 것으로 알려져 있다.

현재까지의 구현 사례 및 계획들은 시스템의 구조 및 기능에 대해 자세하게 언급하고 있지 않아 각각을 면밀히 분석하기는 어렵지만, 각 사례별로 ad-hoc한 구조 및 기능을 채택하고 있는 것으로 보인다. 이를 반영하듯이 연방 데이터베이스 시스템 및 분산 객체 관리 기술 (특히 CORBA)을 이용해 통합 데이터베이스 시스템을 구현해야 한다는 제안이 최근 국내외에서 논문, 보고서, 학술 회의 등을 통해 발표되고 있다. 연방 데이터베이스 시스템을 이용해 구현해야 한

다는 제안들 중 미국에서 발표된 일부는 파일럿 (pilot) 시스템의 형태로 구현이 이루어졌으나, 아직은 실험적인 단계일 뿐 실제 활용되지는 못하고 있다. CORBA를 이용해 구현해야 한다는 제안들로 아직은 실제 구현보다는 제안 단계에 불과한 실정이다.

또한, 연방 데이터베이스 시스템과 CORBA를 함께 이용해 통합 데이터베이스 시스템을 구현해야 한다는 제안도 발표되고 있지만, 이 역시 구체적인 방법을 제시하지는 못했다.

4. 맺음말

CALS는 데이터베이스 분야와 많은 관련성을 가지고 있으며, 특히 CALS의 통합 데이터베이스 시스템은 이와 같은 관련성을 단적으로 보여 준다. 통합 데이터베이스 시스템은 기업에서 리엔지니어링을 통해 개선된 비즈니스 프로세스들 사이에서 다양한 종류의 표준화된 정보를 효율적이고 통합적으로 공유 및 관리할 수 있게 해 주는 체제라 할 수 있다. 이 시스템의 구현을 위한 요소기술로는 연방 데이터베이스 시스템 및 분산 객체 관리 기술이 최근 자주 언급되고 있다. 연방 데이터베이스 시스템에 대해 데이터베이스 분야에서 그 동안 제안됐던 많은 방법들과, 분산 객체 관리에 대해 OMG가 제안한 표준들인 OMA, CORBA, 객체 모델 등이 CALS와 관련해 언급되고 있다.

현재까지 보고된 통합 데이터베이스 시스템의 구현은 각 사례별로 ad-hoc한 방식에 따라 이루어진 것으로 보이며, 이를 반영하듯 연방 데이터베이스 시스템 및 CORBA를 이용하는 구현 방식이 국내외에서 제안되어 있다. 하지만, 아직 실제 구현에 적용할 수 있을 정도로 구체적인 안이 제시되진 못했다. DC