



CALS 개념의 통합 데이터베이스

김 덕 현*

◆ 목 차 ◆

- | | |
|--------------------|-------------------|
| 1. 서론 | 3. 통합 데이터베이스 구현기술 |
| 2. CALS와 통합 데이터 환경 | 4. 결론 |

1. 서론

CALS는 제품의 수명주기 활동과정에서 산출되는 기술자료의 생산, 관리, 활용을 일관화(streamlining)하기 위한 전략으로서 기존의 다양한 업무기능, 프로젝트, 자료, 그리고 기술들에 대한 조직 전체 차원의 통합을 필요로 한다. 즉, CALS 버전의 구현을 위해서는 Business Reengineering 개념의 경영관리체제 재구축과 Concurrent Engineering 개념의 생산관리체제 재구축, 그리고 이를 뒷받침하기 위한 기존 정보시스템의 통합이 요구된다. 정보시스템의 통합이란 과제는 조직 전체 차원의 정보시스템 모형의 설정이 없이 산발적으로 구축되어 온 다양한 자동화의 섬들(islands of automation) 때문에 생긴 투자의 중복 및 낭비, 정보처리 과정의 비효율, 제공되는 정보의 효용성 미흡 등의 문제들을 해결하기 위한 것이다. CALS의 궁극적 목표가 조직통합(Enterprise Integration)에 있다는 점에서 볼 때 정보시스템의 통합은 CALS 구현을 위한 충분조건이 될 수는 없으나 필요조건이며 전제조건임에는 틀림없다.

정보시스템의 통합은 통합의 중심이 되는 layer에 따라 다양한 기술적 문제들을 포함한다. 즉, 상이한 프로토콜을 갖는 정보통신망의 연계, 상이한 데이터베이스 내지는 파일 관리체제에 의해 구축된 자료들의 통합(:data integration), MIS, OA, CAD, CIM, 문헌정보 검색체계 등 독립적으로 개발된 다양한 응용체계와 이들의 구현에 활용된 소프트웨어 도구들의 연동(: control integration), 사용자 인터페이스 통합(: presentation integration) 등의 문제를 포함한다. [4] CALS의 궁극적 목표가 단순한 메시지의 교환 수준이 아닌 정보의 공유에 있음을 감안할 때, CALS 지향의 정보시스템 통합은 자료 내지는 데이터베이스 layer의 통합을 추구하게 된다.

통합 데이터 환경 (IDE: Integrated Data Environment)란 CALS 버전이 완성된 최종상태를 지칭하는 것으로서 제품의 수명주기활동에 관여하는 각각의 사람들이 다루게 되는 자료들을 지리적 원근이나 하드웨어, 소프트웨어, 정보통신망 등 플랫폼의 상이함에 관계없이 쉽게 생산, 관리, 활용할 수 있는 환경을 의미한다 (The IDE is the business environment created by the application of existing national and international standards, pra-

* 정회원 국방과학연구소 실장

ctices, and technologies to automate the management and exchange of information.). [12] IDE의 핵심은 제품의 설계, 생산, 시험, 운용유지 관련 기술자료들에 대한 통합 데이터베이스이며 이는 물리적으로는 분산되어 있는 자료들을 논리적으로 연결시킨 자료저장소를 의미한다. 즉, 통합 데이터베이스는 사용자 관점의 용어이며 데이터베이스 기술 측면에서는 분산 데이터베이스라 할 수 있다. 다만, 종래의 분산 데이터베이스가 지역적으로 분산된 자료들을 동일한 데이터베이스 관리체제 (DBMS)를 이용해서 연계시키는 것에 초점을 둔 것이었다면, CALS에서 제기된 통합 데이터베이스는 이종의 DB 시스템들 (예를들면, 파일관리 시스템, 관계형 DBMS, 객체지향형 DBMS 등)을 연동시킨 것이라는 점에서 다소 차이가 있다.

본 논문은 IDE의 개념적 구조와 그의 핵심인 통합 데이터베이스 구현에 필요한 요소기술들을 소개하기 위한 것이다. 제2절에서는 IDE의 개념적 구조를 미국 국방부의 IDE 과제 [6]와, 국방부의 지원과 IBM을 포함한 18개 기관의 공동 투자로 추진 중인 NIIP (National Industrial Information Infrastructure Protocol) 과제 [9]의 참조모형 (Reference Architecture)을 통해 소개할 것이다. 제3절에서는 통합 데이터베이스 구현에 필요한 요소 기술들과 일부 핵심기술의 연구동향에 대해 소개할 것이다.

2. CALS와 통합 데이터 환경

가. IDE의 기본개념

CALS 비전의 최종상태로 정의되고 있는 IDE는 종래의 MIS, OA, CAD/CAM/CIM 등은 물론, 최근에 관심이 고조되고 있는 EDI/EC (Electronic Data Interchange/ Electronic Commerce), CE (Concurrent Engineering), PDM (Product Data Management),

CSCW (Computer Supported Cooperative Work) 등의 공통적인 구현목표라 할 수 있다. 종래의 정보시스템들이 물리적으로 구축된 조직체 내부와 조직체들 간에서 다루어지는 정보에 대한 중앙집중식의 폐쇄적 체제였다면 CALS/IDE는 지리적, 시간적, 문화적 거리를 초월한 가상기업(Virtual Enterprise)들 간에서 다루어지는 정보에 대한 분산환경하에서의 개방적 체제라 할 수 있다.

IDE의 특징은 미국 국방부의 다음과 같은 구현전략을 통해 좀더 구체적으로 이해할 수 있다. [5] 첫째, IDE 하에서 일단 작성된 자료는 (format 변경등의 이유로) 재작성되는 일이 없이 다양한 용도로 활용될 수 있어야 한다 (Create Data Once, Use Many Times.). 둘째, IDE는 상이한 하드웨어, 소프트웨어, 정보통신망의 플랫폼들을 그대로 활용할 수 있는 개방형 체제로 구현되어야 한다. 즉, IDE는 개방형 표준을 지원하는 모든 제품에 의해 구현될 수 있어서 특정회사의 제품/기술에 의존적이지 않아야 한다. 셋째, IDE는 활용주체가 지리적으로 산재된 개인 및 조직체이므로 LAN 및 WAN 상에서 운용될 수 있어야 하며 넷째, 그에 따라 클라이언트-서버 방식의 분산처리가 가능해야 한다. 다섯째, IDE는 제품의 수명주기활동에 관련된 모든 사람들간의 정보공유를 지향하나 개인 및 조직체의 권한에 따라 각각의 자료에 대한 다단계의 접근통제가 가능해야 한다. 여섯째, 분산환경하에서 자료의 공동저작과 활용이 가능하도록 지역 및 중앙에서 버전통제 (version control) 및 형상관리 (configuration management) 기능이 제공되어야 한다.

자료 측면에서 IDE는 제품의 개발관리 자료 (예: RFP/RFQ, 계약문서, 진도보고서), 설계/제작 자료 (예: 규격서, 도면, 소요 자재/부품/치공구 목록), 그리고 운용/지원 자료 (예: 정비명령서, 기술교범) 등을 대상으로 한다. 기능 측면에서 IDE는 자료의 준비/작성 (예: CAD, 해석, 3차원

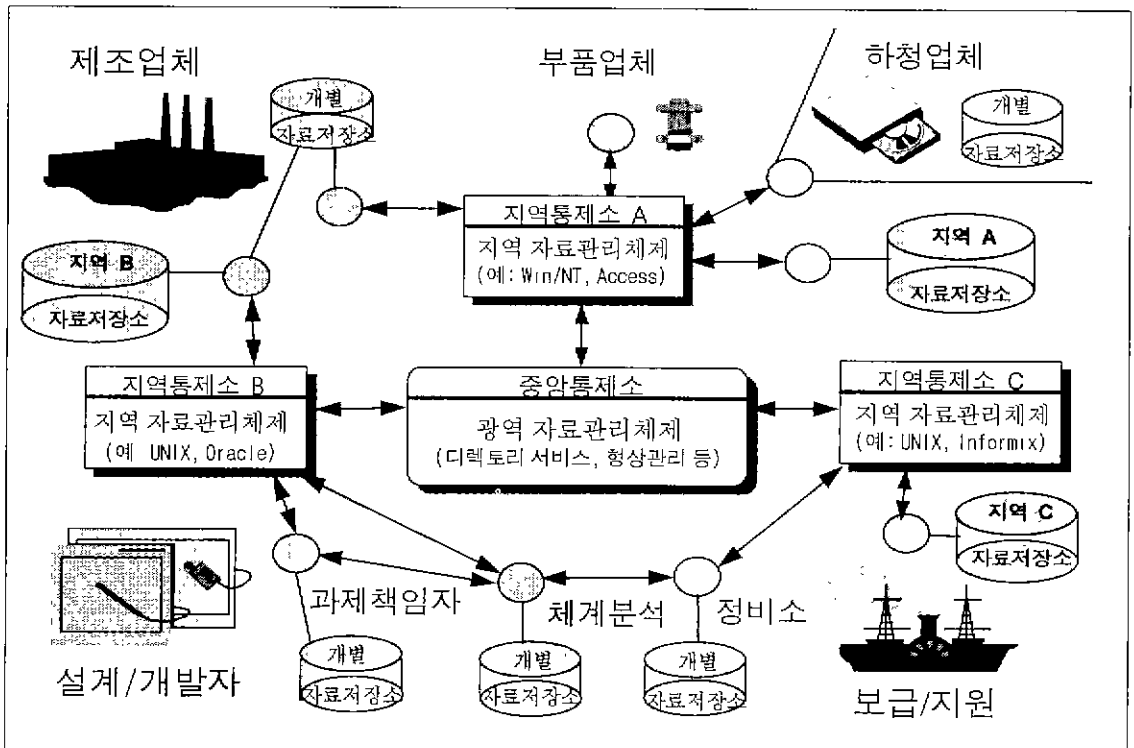
모델링), 자료변환 (예: 스캐닝, 래스터/벡터 그래픽 자료변환), 자료 저장 (예: 암호화, 압축, 색인 부여), DB 구축 (예: 스키마 정의/유지, 자료 생성/갱신), 그리고 자료 검색/배포 (예: 조회/질의, 전자출판) 등을 대상으로 한다.

IDE의 참조모형은 1994년 7월에 발표된 IWSDB 예비설계 보고서 [7]를 통해 개략적으로 제시된 바 있으며 컴퓨터 시스템 구성, 자료/정보 모형, 관리 체제, 그리고 사용자를 위한 정보 서비스 등이 정의되어 있다. [6]

나. 미국 국방부의 IDE 참조모형

미국 국방부는 94년부터 CALS 구현의 초점을 IDE의 구현에 두고 있으며 99년까지 그 기본형을 개발할 예정이다. IDE 과제에는 현재 West Virginia에 위치한 ManTech Advanced Technology Systems사를 포함한 10여개사가 참여하고 있으며, 인터넷을 기반으로 한 시제품인 DCN/ICN (Defense/International CALS Network)을 구축, 시험운용 중이다.

컴퓨터 시스템 관점에서의 IDE는 상이한 플랫폼 위에 구축된 정보자원들이 LAN/WAN을 통해 연결된 3계층 구조의 클라이언트-서버 시스템이다. (그림 1. 참조) 제품의 수명주기활동과 관련하여 생산된 기술자료들은 개별(local) 자료저장소 또는 설계/개발자, 제조업체, 하청업체별로 구축된 지역(regional) 자료저장소에 저장, 관리된다. 지역통제소는 관계형 내지는 객체지향형 DBMS, 일반 파일관리체제 등에 의해 구축된 자료들에 대한 디렉토리 관리, 자료사전 관리, 버전



(그림 1) IDE 하에서의 3계층 자료관리체제 (자료원: 미국 ManTech 사)

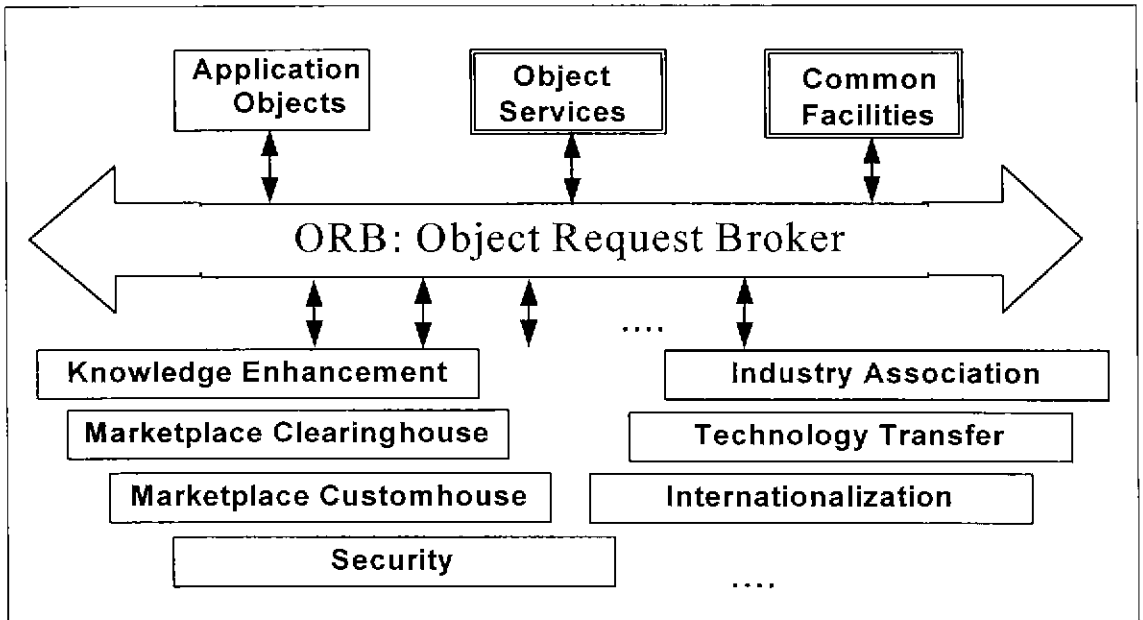
관리, 보안관리, 백업 등의 기능을 수행한다. 중앙 통제소는 지역별 저장소에 저장, 관리된 자료들에 대한 광역(global) 통제를 위한 디렉토리 관리 및 버전 관리 기능을 수행한다. 이와 같은 IDE 환경 하에서 사용자는 원하는 정보가 저장, 관리되고 있는 위치나 형식, 경우에 따라서는 정확한 명칭을 모르더라도 쉽게 필요한 정보에 접근할 수 있게 된다.

현재 제안된 IDE는 (1) LAN/WAN을 포함한 인터넷 방식의 정보통신망 위에서 (2) 제품자료의 국제표준으로 자리잡고 있는 STEP으로 표현된 자료들을 관계형 내지는 객체지향 DBMS들에 의해 저장, 관리하고 (3) 객체지향기술에 대한 국제적 표준화 그룹인 OMG의 분산객체관리 규격인 CORBA (Common Object Request Broker Architecture)의 지원도구를 활용해서 이질적 분산 시스템들의 상호연동을 가능하게 하여 (4) 사용자들에게 지능적 정보처리, 보안, 거래처리(Clearing-house), 품질관리(Customhouse), 산업간 협력 및 국

제화, 기술이전 등의 서비스를 제공하는 계층들로 구성되어 있다. (그림 2. 참조)

다. 미국 제조업 중심의 NIIP 참조모형

NIIP는 최신의 정보기술 및 제품을 이용해서 제조업 중심의 가상기업 (I-VE: Industry Virtual Enterprise)을 시험구현해 보기 위해 IBM을 포함한 18개 기관이 1994년 9월부터 2년씩 2단계에 걸쳐 추진하고 있는 과제를 말한다. NIIP는 DARPA (Defense Advanced Research Projects Agency)에 의해 국가정보화 기반구조 (NI: National Information Infrastructure)의 핵심구성품으로 정의되어 과제화 되었으며 미 공군의 Mantech office에 의해 관리되고 있다. NIIP의 목표는 (1) 미국의 제조업들을 대상으로 분산환경 하의 상이한 컴퓨팅 자원들을 통합할 수 있는 개방형 기반구조를 설정하고 (2) NIIP 참조구조를 문서화해서 공개함으로써



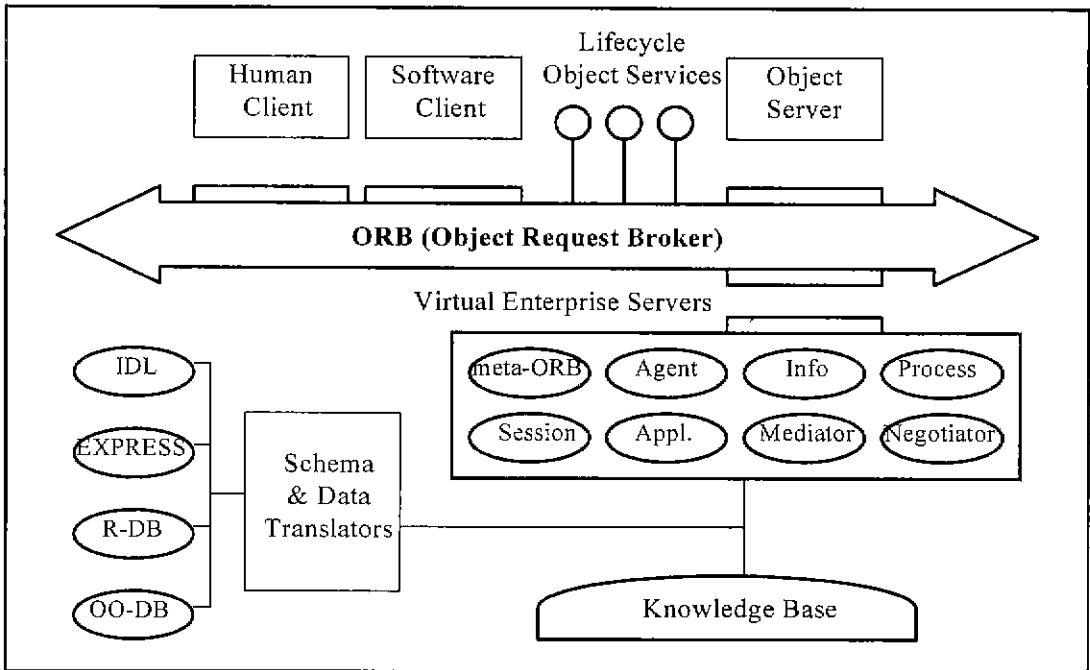
(그림 2) 미국 국방부의 IDE 참조모형 (자료원: 미국 ManTech사)

써 관심있는 기관들이 이를 활용할 수 있도록 하며 (3) 가상기업의 구현을 위한 효과적인 기술들을 선정, 개선, 통합해서 NIIP를 구현하고 (4) 민수 및 군수산업에의 시험적용을 통해 그 타당성을 입증하며 (5) NIIP 관련 기술 및 제품을 국제적 경쟁력을 갖는 상품으로 개발하고 (6) 기존 정보시스템들에서 운용 중인 자료들의 변환방안을 마련하여 (7) NIIP 기술이 최종사용자는 물론 소프트웨어 공급업체, 시스템 통합 업체 등에서 광범위하게 채택될 수 있게 함에 있다.

Framework Initiative) 등에서 다루어 지고 있는 워크플로우 및 작업관리 기술 등을 통합한 형태이다. (그림 3. 참조)

NIIP 경우는 컨소시엄에 참여하고 있는 기관들이 보유하고 있는 기술 및 상품을 기반으로 하고 있기 때문에 국방부의 IDE에 비해 개방성이 상대적으로 부족한 것으로 판단된다.

3. 통합 데이터베이스 구현기술



(그림 3) NIIP의 참조모형

NIIP는 IETF (Internet Engineering Task Force)의 인터넷 기술, OMG의 CORBA, ISO의 제품자료 표준인STEP/EXPRESS 기술, 그리고 WFMC (Workflow Management Coalition), DARPA의 I3 (Intelligent Integration of Information), CFI (CAD

미국 국방부의 IDE와 NIIP의 IDE는 공통적으로 (1) 기존 업무체제의 분석, 설계, 재구축을 위한 시스템/정보 모델링 기술 (예: IDEF, OMT, EXPRESS-G), (2) 인터넷을 포함한 정보통신망의 구축 및 활용기술 (예: LAN/WAN, 정보/컴퓨터

보안, EDI, 멀티미디어 전자우편, 화상회의), (3) 제품자료 표준인 STEP 응용기술 (예: EXPRESS, Application Profiles, STEP 구축방법론, VRML), (4) 분산형 및 지능형 DBMS 구현기술 (예: 다중 DB 언어, DB 스키마의 통합/절충, 지식베이스와 DB와의 통합), (5) 분산객체컴퓨팅 지원 소프트웨어 구현기술 (예: CORBA & ORB, Object Adaptor 또는 bridge), (6) 지능형 응용 소프트웨어 구현기술 (예: 워크플로우 처리 및 협업지원 agent, 정보 검색/여과용 agent, KQML²) 등의 핵심기술들을 필요로 하는 것으로 분석된다. 제한된 지면을 통해 상기한 요소기술들 각각에 대해 언급하기가 곤란한 점에서 이하에서는 분산객체 컴퓨팅 기술과 다중 DB 기술의 의의와 개략적 연구방향만을 소개하고자 한다.

가. 분산객체 컴퓨팅

분산컴퓨팅이란 하나의 application을 여러개의 컴퓨터 시스템을 통해 처리하는 방식을 말한다. 분산컴퓨팅은 크게 보아 3단계를 거쳐 발전되어 온 것으로 볼 수 있다.[2] 초기단계는 일련의 메시지를 지리적으로 떨어진 시스템 간에 주고 받을 수 있게 한 stream-oriented 방식이며, 2단계는 원격지의 컴퓨터 시스템에 특정 작업을 의뢰하는 RPC (Remote Procedure Call) 방식이다. RPC 방식은 OSF사의 DCE (Distributed Computing Environment), SunSoft사의 ONC 및 ONC+ 등의 제품에서 지원된다. DCE는 RPC, 분산 디렉토리, 스레드, 표준시간의 동기화, MIT에서 개발한 Kerberos 인증시스템과 사용자 계정관리 등을 포함한 보안기능, 파일과 디렉토리를 복수 개의 서버에 중첩시켜서 가용성을 높이기 위한 분산 파일 시스템, diskless system 등의 기능을 제공한다.

3단계인 분산객체컴퓨팅 (distributed object computing)은 종래의 분산컴퓨팅 (distributed computing) 방식이 상호접속성(interconnectivity)의 보장 수준에 머물렀던 것을 객체지향 컴퓨팅(object-oriented computing) 방식의 도입을 통해 상호연동성(interoperability)의 보장 수준으로 발전시킨 것이다. 분산객체컴퓨팅은 이질적이며 분산된 다양한 정보 자원들을 각각 독립된 객체로 보고 이들 객체들이 표준 인터페이스--예를들면, CORBA의 IDL (Interface Definition Language)--를 통해 상호연동시키기 위한 기술이라 할 수 있다.

분산객체컴퓨팅의 개념적 모형으로는 OMG (Object Management Group)의 OMA (Object Management Architecture)와 ISO 및 CCITT의 ODP (Open Distributed Computing), Microsoft사의 COM (Component Object Model)/ OLE (Object Linking & Embedding), Next사의 NextStep, 그리고 OSF사의 DME (Distributed Management Environment) 등이 있다. [8] OMA는 정보자원들 간의 표준적인 인터페이스 정의 언어로서 IDL을 제공하고 wrapper³를 통해 상이한 application들의 통합을 지원한다. 객체의 생성, 소멸, 저장, 이름부여, 이벤트 통지, 트랜잭션 처리, 동시성 제어 등을 위한 Object Services는 COSS (Common Object Service Specification)로 정의되어 있고, ORB는 CORBA (Common Object Request Broker Architecture)라는 업계 표준규격으로 정의되어 있으며 이를 지원하는 다양한 제품들이 출하되어 있다. CORBA는 91. 12.에 revision 1.1이, 93. 12.에 revision 1.2가, 그리고 95. 7.에 revision 2.0이 발표되었다. CORBA 2.0은 상이한 ORB들간의 상호연동성을 보장하기 위한 GIOP (General Inter-ORB Protocol), IIOP (Internet IOP), ESIOP (Environment Specific IOP) 등과 타 객체 시스템 (예를들면, Microsoft사의 COM/OLE)과의 연동을 위한 bridge들을 정의하고 있다. [10]

¹ Virtual Reality Modeling Language

² Knowledge Query Manipulation Language

³ legacy system을 위한 인터페이스

OSF사의 DME는 다양한 객체시스템들에 대한 중립적 기능과 CORBA보다 더 완벽한 객체지향 방식의 지원을 위해 설계되었다. DME의 설계목적은 기존 시스템들이 (1) 일관성 없는 사용자 인터페이스를 제공하고 있으며 (2) 시스템 운영 프로토콜의 상이함으로 인한 상호연동성의 한계를 갖고 있고 (3) 적용 업무의 규모에 따른 확장/축소가 불가능한 점들을 해결하고자 함에 있었다. 그 결과 DME는 (1) Motif를 기반으로 한 일관된 사용자 인터페이스를 제공하고 (2) 시스템 운영을 위한 SNMP (Simple Network Management Protocol), CMIP (Common Management Information Protocol), DCE RPC 등 국제적 표준 프로토콜을 지원하며 (3) 규모의 차이에 따라 신축이 가능하도록 node, cell, enterprise로 구성되는 3-tiered hierarchy를 제공한다. DME는 이질적인 객체시스템들의 상호연동성 지원뿐만 아니라 네트워크 시스템 자체의 효과적 운영을 위한 소프트웨어의 자동설치, 시스템 성능의 모니터링, 시스템 구성 관리 등 기능을 포함하는 점에서 value-added CORBA라 할 수 있다. [8]

나. 다중 데이터베이스 시스템 (MDBS: Multi-database System)

상이한 DB 내지는 DBMS들의 통합은 단순하게는 gateway 개념의 소프트웨어를 중간에 개재시키는 방법이 있겠으나 분산환경 하에서는 처리효율의 문제를 감안할 때 바람직한 방법이라 할 수 없다. 분산객체컴퓨팅에서 DB의 통합은 크게 두가지 방향, 즉 전체 시스템을 구성함에 있어서 상이한 DBMS 자체를 구조적으로 통합하는 방안과 상이한 DBMS의 스키마(schema)들을 논리적으로 통합하는 방안이 있다.[8] 전자는 DBMS를 포함한 상이한 객체시스템들을 통합해주는 효과적 수단인 CORBA에 의해 가능하나 이는 완벽한 상호연동성을 구현하는데는 한계가

있다. 왜냐하면 CORBA의 초점이 자료에 함축된 의미요소(semantic) 보다는 자료를 표현하는 형식(syntax)에 있는 점에서 다수의 작업들을 상호연동시키는데 필요한 충분한 정보의 공유가 어렵기 때문이다. 완벽한 상호연동성의 구현을 위해서는 상이한 DBMS에 정의된 스키마에 함축된 의미들을 상호보완하고 조정, 통합하는 과정이 필수적이다.

MDBS란 이미 존재하는, 독립적이면서도 이질적인 (autonomous and heterogeneous) DB 시스템들을 통합한(confederated) 시스템이다. [Pitoura등, 1995] MDBS의 통합은 전통적으로 두 가지 접근방식, 즉 non-federated and federated approaches에 의해 추구되어 왔다. Non-federated approach는 DB 스키마에 대한 통합보다는 다종의 DBMS를 공통적으로 정의, 조작할 수 있는 multi-DB language를 개발, 활용하는 방식이다. Federated approach는 (1) 기존의 상이한 DB들로부터 통합에 필요한 스키마를 각각 추출해서 component schema를 만들고 (2) 이를 통합 대상으로 분할해서 export schema들을 만든 다음에 (3) 이들을 논리적으로 결합해서 federated schema (또는 global schema)를 만드는 방식이다. DB 스키마들을 통합하기 위한 schema integration [Batini등, 1986] 논리는 관계형 DB 내에서는 어느 정도 정의된 바 있으나 이종의 DB 시스템들의 경우는 아직 충분한 연구가 없는 상황이다.

이종의 DB 스키마들을 효과적으로 통합하기 위해서는 충분한 표현력을 가지면서도 구현의 편의성이 보장될 수 있는 단순함을 가진 표준 데이터 모형 (CDM: Canonical data model 또는 Common Data Model)의 개발이 필수적이다. CDM은 (1) 표현력 면에서는 유형화/실체화 (classification/instantiation), 일반화/특수화 (generalization/specialization), 집계화/분할 (aggregation/decomposition) 등 다양한 추상화 수단은 물론 행위적 특성과 제약조건들을 포함시킬 수 있는 수단을 가짐으로써 충분한 확장성을 가져야 하며 (2) 구현의 용이성을 위해서는

모델링 도구가 가급적 단일개념-예를들면, 객체 또는 함수-이 되는 것이 바람직하고 스키마 통합을 위한 다양한 연산자-예를들면, union, subtraction, Cartesian product 등-들을 가질 수 있어야 한다.[11].

객체지향 데이터 모형은 통합 DB의 CDM이 되기 위한 강력한 표현력과 구현 용이성을 갖추고 있으나 몇가지 측면에서의 개선이 요구된다. 즉, (1) 의미론적 데이터 모형 내지는 지식기반 데이터 모형들-예를들면, DAPLEX, VODAK 등-이 제공하는 것 보다 강력한 semantic constructs의 개발, (2) 다수의 실체(instance)들이 갖고 있는 공통적 특성에 대한 규격으로서의 타입과 실체들의 집합으로서의 클래스에 대한 구분, (3) 메타자료 및 메타지식과 객체 특성의 상향계승(upward inheritance) 등을 정의할 수 있는 도구의 개발 등이 요구된다.

지금까지 MDBS에 대한 여러가지 연구와 시험 구현이 이루어 졌으며 대표적인 것으로는 HP사 DB 연구실의 Pegasus, GMD-IPSI의 ViewSystem, UniSQL사의 UniSQL/M, GTE사의 DOMS (Distributed Object Management System), MCC사의 Carnot, Georgia 대학의 FIB 등이 있다. [Pitoura 등, 1995] Pegasus는 functional data model인 Iris를 기반으로 해서 HOSQL이라는 multi-DB 언어를 구현한 시스템이고, ViewSystem은 VODAK이라는 데이터 모형과 Smalltalk 언어에 기반한 DB 개발/운영환경으로서 스키마 통합을 위한 다양한 연산자들을 포함하고 있다. UniSQL/M은 SQL을 확장한 SQL/M을 통해 다종의 DBMS에 접근할 수 있게 한 시스템이다. DOMS는 FROOM (Functional, Relational, Object-Oriented data Model)이라는 CDM을 바탕으로 DB는 물론 하이퍼미디어 시스템과 다양한 application 들을 통합할 수 있게 한 시스템이다. Carnot은 규칙에 기반한 스키마 통합방식을 구현하고 있으며 DB, 전문가 시스템, 지식베이스, 워크플로우 등을 논리적으로 통합할 수 있는 시스템이다.

4. 결론

본 논문에서는 CALS의 최종목표인 통합 데이터 환경 (IDE: Integrated Data Environment)의 의미와 참조모형들, 그리고 요소기술들 중 분산객체컴퓨팅 기술과 다중 데이터베이스 시스템 기술에 대해 소개하였다.

CALS의 도입에 있어서 선진국보다 적어도 10년이 뒤진 우리나라 입장에서 선진국 수준을 따라 잡기 위해서는 IDE의 구현을 위한 연구개발의 조속한 착수가 정책적, 기술적 측면에서 매우 중요한 문제라 본다. 이는 첫째, IDE는 CALS의 궁극적 목표 그 자체이기 때문이며 둘째, CALS 표준 (예: SGML, CGM, IGES, IETM 등)에 의한 정보의 교환이라는 중간단계 목표달성에 필요한 대부분의 기술 및 제품들을 선진국이 이미 독/과점하고 있는 상태에서 유사제품의 모방개발에 치중하는 것은 국제경쟁력면에서 승산이 없다고 보기 때문이다. IDE는 산업체의 실무자들은 물론 학계, 연구계의 다양한 전문가들의 광범위한 참여가 필수적인 과제인 점에서 IDE의 시범구현을 위한 산.학.연의 컨소시움 구성과 이의 활발한 활동을 지원할 수 있는 정부와 업계의 과감한 투자가 요구된다.

참고문헌

- [1] 김덕현, 통합 데이터 환경: 개념 및 구현기술, 한국 CALS/EC 학회지, 제1권, 제1호, 1996, pp. 69-92
- [2] 이동만, Distributed Objects and Component Integration: CORBA, KRNET 95 특강자료집, 1995. 7.
- [3] C. Batini, M. Lenzerini, and S. Navathe, A Comparative Analysis of Methodologies for Database Schema Integration, ACM Computing Surveys, v.8, n.4, 1986. 12., pp.323-364

[4] M. Chen, and R. Norman, A Framework for Integrated CASE, *IEEE Software*, March, pp.18-22

[5] R. Kidwell, Integrated Data Environment, *CALS KOREA 95 튜토리얼 교재 및 발표자료집*, 1995. 10.

[6] R. Kidwell and J. Brazy, CALS and EC/EDI (IDE 중심), 국방과학연구소 해외전문가 초빙 기술세미나 교재, 1996. 6.

[7] ManTech International Co., Preliminary Integrated Weapon System Database(TWSDB) Implementation Strategy Paper, Version 2, 1994. 7.
* <http://www.mantech-wva.com>

[8] J. Nicol, C. Wilkes, and F. Manola, Object-orientation in Heterogeneous Distributed Computing Systems, *IEEE Computer*, v.26, n.6, 1993. 6., pp. 57-67

[9] NIIP Consortium, NIIP (Cycle 0, Revision 6), NIIP Consortium, 1995. 11.
* <http://www.niip.org>

[10] OMG, The Common Object Request Broker: Architecture and Specification, Revision 2.0, 1995
* <http://www.omg.org>

[11] E. Pitoura, O. Bukhres, and A. Elmagamid, Object-orientation in Multidatabase Systems, *ACM Computing Surveys*, v.27, n.2, 1995. 6., pp. 141-195

[12] U.S. DoD, CALS Strategic Overview, U.S. DoD
* <http://www.acq.osd.mil/cals/>



김 덕 현

1976년 서울대학교 산업공학과(공학사)
1986년 KAIST 경영학과 졸업(공학석사)
1993년 KAIST 경영학과 졸업(공학박사)

1996년-현재 한국 CALS/EC 학회 데이터베이스 연구위원회 위원장
1976년-현재 국방과학연구소 (현재 전산부 전산개발 실장)
관심분야: 분산객체관리, 객체지향 시스템 모델링, 인터넷

'97 춘계학술 발표대회 논문 모집 안내

- 일 시 : 1997. 4. 12 (토)
- 장 소 : 한남대학교 (대전)
- 논문마감 : 1997년 3월 14일 (금) 까지

* 자세한 사항은 학회지 뒷면 논문모집 참조