

통합 데이터 환경: 개념 및 구현기술

김덕현*

Integrated Data Environment (IDE): Concept & Implementation Technology

Duk-Hyun Kim

ABSTRACT

An Integrated Data Environment (IDE) is the end state of the CALS vision. It refers to an environment where all the data generated from the entire lifecycle activities of a product can be shared by anyone that participates in the management, engineering, manufacturing, or support of the product, although the data are geographically distributed and are maintained in heterogeneous platforms. The primary purpose of this paper is to introduce the fundamental concept, architecture, and core implementation technology of an IDE. The secondary purpose is to suggest some strategic directions to the planning staffs in both government and industry, and to suggest some research issues to the researchers in academia. The reference architecture of an IDE being developed by the US DoD and that of NIIP (National Industrial Information Infrastructure Protocol) being developed by a consortium are discussed. Two principal issues of implementing an integrated database, i.e., distributed object computing including CORBA and multi-database system (MDBS) are reviewed. As a conclusion, the author suggests that daring investment in prototyping an IDE by a nationwide consortium is essential to keep up with the advanced countries in CALS implementation.

* 국방과학연구소 전산부 책임연구원

1. 서론

CALS는 무기체계 또는 제품의 수명주기 활동(즉, 기획/계획, 설계, 자재조달, 제작, 시험평가, 생산, 운용지원 등) 전반에서 생성, 저장, 관리, 유통, 활용되는 각종 기술/관리 자료들의 흐름을 컴퓨터 시스템 및 정보통신망을 이용해서 일관화된(streamlined) 디지털 정보의 흐름으로 변환하기 위한 정부 및 산업체의 공동전략이다. 1985년 미국 국방부에 의해 낙후된 통신시스템과 서류중심의 업무처리 방식에서 생기는 비효율을 제거하기 위한 방안으로 채택된 CALS는 이제 국방차원에서는 무기체계의 획득을 포함한 수명주기 전반에 대한 일관된 지원체제(Continuous Acquisition and Lifecycle Support)로, 민간차원에서는 광속의 거래체제(Commerce At Light Speed)로 발전되고 있다.

통합 데이터 환경(이하 IDE: Integrated Data Environment)은 CALS 구현이 완성된 최종상태를 지칭하는 것으로서 개념적으로는 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.¹ [USDOD] CALS 자체를 '정보와 지식의

공유'라는 비전의 출발점이라 한다면 IDE는 그러한 비전의 총착점이라 할 수 있다. IDE는 당초에는 IWSDB(Integrated Weapon System Database)로 불리웠으나 군수측면에 치우친 개념으로 인식됨을 피하기 위해 개명된 것이다.

본 논문은 IDE의 구현에 관여하게 될 정책 담당자와 산업체의 관리자 및 실무자들에게 IDE의 의의와 개념적 구조, 요소기술 등을 소개하고 학계 및 연구계의 연구자들에게 IDE의 구현에 필요한 몇가지 핵심기술의 연구방향을 제안하기 위한 것이다. IDE의 기본구조는 현재 미국 국방부가 구상하고 있는 IDE 참조모형(Reference Architecture)² [Kidwell & Brazzy, 1996]과, 제조업 중심 가상기업(Virtual Enterprise)의 시험구현을 위해 정부와 민간이 공동추진 중인 NIIP(National Industrial Information Infrastructure Protocol)의 참조모형 [NIIP, 1995]을 중심으로 소개할 것이다. IDE의 구현에 필요한 요소기술들은 사실상 경영, 전산, 통신, 산업공학 및 경영과학 등 다양한 학문분야에 그 뿌리를 두고 있는 것들인 점에서 본 논문에서는 개별기술에 대한 심도있는 분석보다는 전반적 기술구조에 초점을 두어 설명할 것이다. IDE의 중추라 할 수 있는 통합 데이터베이스의 구현을 위한 핵심기술들인 분산객체컴퓨팅(distributed object computing) 기술과 이종의 데이터베이스 관리체제들을 통합하기 위한 다중

¹ "The IDE is defined as the business environment created by the application of existing national and international standards, practices, and technologies to automate the management and exchange of information." in US DoD's

CALS Strategic Overview

² 특정 시스템의 구현에 필요한 요소기술들과 이들의 상호연관성에 대한 개념적 모형

데이터베이스 (multi-database system) 기술에 대해서는 보다 구체적인 분석을 통해 국내에서의 활발한 연구를 촉구하고자 한다.

2. IDE의 개념적 배경

2.1 CALS의 의의

경영전략으로서의 CALS는 하나의 조직체를 구성하는 사업/과제, 기능, 자료, 기술 등을 조직 전체 차원에서 재구축하기 위한 조직통합 (Enterprise Integration) 전략이라 할 수 있다. 조직통합은 기업혁신 (Business Reengineering) 보다 더 포괄적인 개념으로서, 부분별로 폐쇄된 사고방식 ("stovepipe" thinking)을 조직전체 차원의 열린 사고 ("enterprise" thinking)로 전환하기 위해 지식 및 자료의 공유를 위한 정보시스템의 구현과 이에 따른 관리체제의 전반적 재구축을 필요로 한다.

정보시스템으로서의 CALS는 (1) 첨단 정보시스템 기반구조 (infrastructure)와 (2) 제품 획득 (또는 조달) 기획/계획 지원시스템, 설계/제작/제조 자동화 (CAD/CAE/CAM) 시스템, 군수지원/운용지원 시스템 등을 통합한 응용체제로 구성된다. CALS는 다양한 하드웨어, 소프트웨어, 정보통신망들을 논리적으로 상호연동시킬 수 있는 개방형 기반구조를 필요로 한다. 이는 CALS가 대상으로 하는 응용체제들의 대부분이 기능별,

과제별로 독립적으로 구축되어 물리적인 통합이 불가능한 이른바 '자동화의 섬들' (islands of automation)을 이루고 있기 때문이다. 따라서, CALS의 구현전략은 정보통신 기술의 발전과 이에 따른 기존 정보시스템들의 수용능력의 확대에 따라 1단계 목표를 다양한 국제표준들에 의한 자료교환에, 2단계 목표를 통합 데이터베이스 (IDB: integrated database)를 통한 자료공유에 두고 있다.

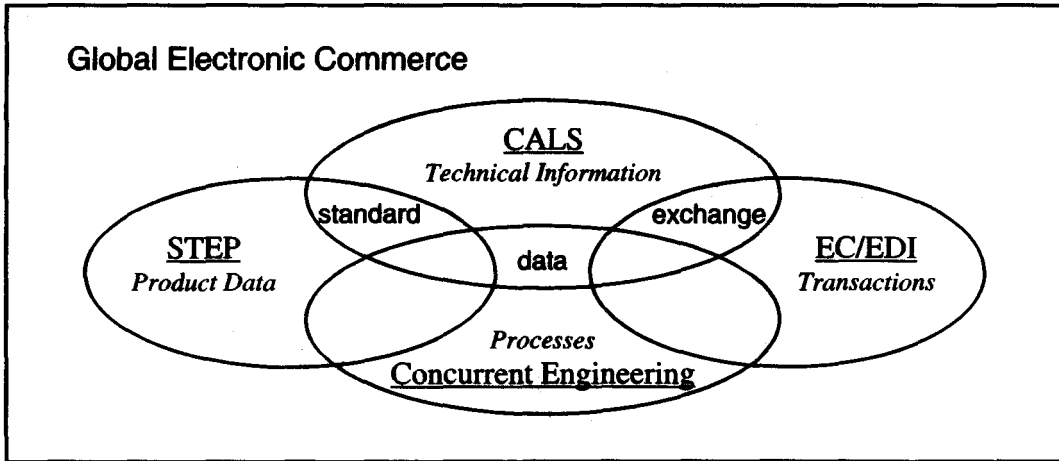
CALS에서 통합 데이터베이스란 지리적으로 분산된 이질적인 데이터베이스들을 논리적으로 연계시킨 자료저장소 (data repository)를 가리키는 사용자 관점의 용어로서 데이터베이스 기술 관점에서는 분산 데이터베이스의 한 형태라 할 수 있다. 다만, 종래의 분산 데이터베이스 연구가 주로 지역적으로 분산된 자료들을 동일한 데이터베이스 관리체제 (DBMS)를 이용해서 연계시키는 것에 초점이 두어졌다면 CALS에서 제기된 통합 데이터베이스는 이종의 DB 시스템 (예들들면, 파일관리 시스템, 관계형 DBMS인 ORACLE, Informix, Sybase 등)들을 연동시킨 것이라는 점에서 다소 차이가 있다.

CALS와는 약간 다른 배경 하에서 출발한 Concurrent Engineering (CE)³, STEP (Standard for the

³ 일반적으로 '동시공학'이라 번역되고 있으나 'concurrent'를 '동시'보다는 '일관된'으로 'engineering'을 학문분야를 연상시키는 의미의 '공학'보다는 '제작'으로 보아 '일관제작' 또는 '제작의 일관화'로 번역할 것을 제안함

Exchange of Product model data), 전자자료교환 (EDI: Electronic Data Interchange), 전자 상거래 (EC: Electronic Commerce) 등은 현 시점에서는 CALS 와 상호보완적인 관계에 있는 것으로 이해함이 바람직한 것으로 판단된다. 하나의 정보시스템을 대상 시스템에 포함된 자료 및 프로세스에 대한 자동화체제로 정의할 때, CALS 는 제품의 수명주기를 통해 발생하는 기술자료에, CE 는 제품 수명주기활동 프로세스에 초점을 둔 것이라 할 수 있다. 또한 CALS 가 조직체 내부는 물론 조직체 간에 유통되는 기술자료에 초점을

둔 것임에 반해 EDI 는 조직체간의 거래처리에 필요한 관리자료에 초점을 둔 것인 점에서 차이가 있다. STEP 은 모든 제품자료의 교환을 위한 국제적 표준으로서 CALS 및 EC 를 위한 포괄적 자료교환 표준이 될 것으로 예상되고 있다. 거래라는 개념을 물리적인 조직단위는 물론 논리적인 조직단위 (예: 가상기업) 간에 이루어지는 모든 재화 및 서비스의 유통으로 확대, 해석할 때 (Global) EC 는 CALS, CE, EDI, STEP 등을 포괄하는 개념이 된다. (<그림 1> 참조)



<그림 1> CALS/EC 개념도 (자료원: 미국 ManTech 사)

2.2 통합 데이터 환경의 의의

CALS 구현의 최종상태로 정의되고 있는 IDE 는 CALS 만의 목표라기보다는 정부, 산업체, 교육 기관 등 모든 조직체들이 추구하고 왔으며, 앞으로도 추구하고 갈 기본적인 정보화 목표들 중 하나라 할

수 있다. 필요한 자료 및 지식들을, 필요한 사람들이, 필요한 시기에 활용할 수 있도록 하는 체제를 구축하는 것은 효과적인 행정 서비스와 효과적인 제품 개발/생산/운용지원을 위해, 그리고 효과적인 교육과 연구를 위해 매우 중요한 일이기 때문이다.

이런 점에서 IDE 는 기존의 MIS, OA,

CAD/CAM/CIM 등은 물론, 최근에 관심이 고조되고 있는 EDI 및 EC, CE 및 제품자료 관리체계 (PDM: Product Data Management), 그리고 컴퓨터 지원 협업체제 (CSCW: Computer Supported Cooperative Work) 등의 공통적인 구현목표라 할 수도 있다. 기존 정보시스템들의 구현목표와 IDE의 구현목표 간에 차이가 있다면 전자가 물리적으로 구현된 조직체 내부와 조직체들 간에서 다루어지는 정보를 대상으로 한 중앙집중식의 폐쇄적 체제라면 후자는 지리적, 시간적, 문화적 거리를 초월한 가상기업들 간에서 다루어지는 정보를 대상으로 한 분산 환경 하에서의 개방적 체제라는 점에 있다.

IDE는 CE와 CALS, EC는 물론 가상기업의 효과적 구축을 위한 필수조건이 된다. IDE는 제품의 개발, 생산, 운용유지 등과 관련된 시간과 비용을 단축하고 환경변화에 대한 기민한 대응을 가능케 함으로써 기업의 총체적 경쟁력을 향상시키는 원천이 될 수도 있다.

IDE는 자료 측면에서 보면, 제품의 수명주기 활동 전반에서 다루어지는 획득관리자료, 설계/제작자료, 제조자료, 군수지원/운용지원자료 등을 대상으로 한다. 획득관리자료로는 제품에 대한 RFP/RFQ (Request for Proposal/Quotation), 계약문서, 청구서, 진도보고서 등이; 설계/제작자료로는 설계규격서, 설계도면, 부품목록, 소프트웨어 설명문서 등이; 제조자료로는 BOM (Bill of Material), 자재목록, 장비/공구/치구 목록 등이 포함된다. 또한 군수지원자료로는 군수지원분석 (LSA: Logistics Support

Analysis) 자료, 기술명령서 (technical order), 기술교범 (technical manual), 보급품 목록, 형상관리자료 등이 포함된다. 자료형태로는 숫자, 문자, 문서, 그림, 음성 및 영상 등 멀티미디어 형태의 자료가 대상이 된다.

IDE는 처리기능 측면에서 보면, CAD, 해석 및 시뮬레이션, 2차원 또는 3차원 모델링 등 자료의 준비/작성 기능; 인쇄자료의 이미지 문서화를 위한 스캐닝과 래스터/벡터 그래픽 자료의 변환기능; 자료의 암호화, 압축, 기록, 색인부여 등 자료의 저장 관리 기능; 데이터 모델링, 스키마 관리, 자료의 import/export 등 DB 관리 기능; 조회/질의, 보고서 작성, 전자출판 등 자료의 검색 및 배포기능 등을 포함한다.

IDE는 하드웨어도 정보통신망도 소프트웨어도 아니고 특정 프로세스에 대한 정보시스템이라고도 할 수도 없으며 이들 모두를 포함한 것이라고 봄이 타당하다. 미국이 구상하고 있는 IDE의 구성요소에는 (1) 객체 개념에 입각한 시스템 모델링 (2) 자료의 교환 내지는 공유를 위한 기능, 기술, 인터페이스 등에 대한 표준규격 (3) 운용개념, 그리고 (4) 개발모형 등이 포함된다. 객체 개념의 시스템 모델링은 이질적인 분산 시스템들의 정의와 구현을 위한 방안이며, IDE 운용개념은 사용자 관점의 요구사항을 명확히 정의하기 위한 일종의 시나리오이고, IDE 개발모형은 다양한 구현수단과 운용환경을 가진 기관별 개발자들에게 공통적으로 활용가능한 모형을 제공함으로써 불필요한 노력의 낭비가 없도록 하기 위한 것이다.

IDE의 실체는 미국 국방부의 다음과 같은 IDE 구현전략을 통해 좀더 구체적으로 이해할 수 있다. [Kidwell, 1995] 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 체계로 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 (Local/Wide Area Network) 상에서 운용될 수 있어야 하며 넷째, 그에 따라 클라이

언트-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, 분산환경 하에서 자료의 공동저작과 활용이 가능하도록 지역 및 중앙에서 버전통제 (version control) 및 형상관리 (configuration management) 기능이 제공되어야 한다.

미국 국방부의 최근 CALS 추진전략은 정부와 산업체가 공유할 수 있는 IDE 참조모형 과 이를 국제적 경쟁력을 갖는 상품으로 만들기 위한 참조 구현 (Reference Implementation)의 마련에 중점을 두고 있음을 볼 수 있다. [USDOD] (<표 1> 참조)

<표 1> 미국 국방부의 CALS 추진전략

목 표	세 부 목 표	milestone
정부와 산업간의 원활한 자료교환 체제 구축	<ul style="list-style-type: none"> o 정부와 산업이 공유가능한 정보모형 개발 o 민군 겸용기술 (dual-use technology)의 촉진 o 국내/국제표준과 현실과의 조화 도모 o CALS 개념 및 기술에 대한 교육훈련/지원 	'97.7. CALS 표준의 산업표준화 '98.1. IDE 모형 설정 '98.7. IDE 시작품 구현
기존 기술정보/ 관리정보의 디지털화	<ul style="list-style-type: none"> o 기존 하드웨어 및 소프트웨어의 현대화 o 신규자료는 디지털 형식으로 획득 o 기존자료는 디지털 형식으로 변환 o 모든 거래의 디지털화, 전자화 	'98.7. 완료
기술정보의 통합	<ul style="list-style-type: none"> o 통합 데이터 환경 (IDE) 개발 o 국방정보화 기반구조(DII: Defense Information Infrastructure)와 국가정보화 기반구조 (NII: National ~)와의 연계 o 업무처리방식 및 관리체제의 개선 	'99.7. 완료

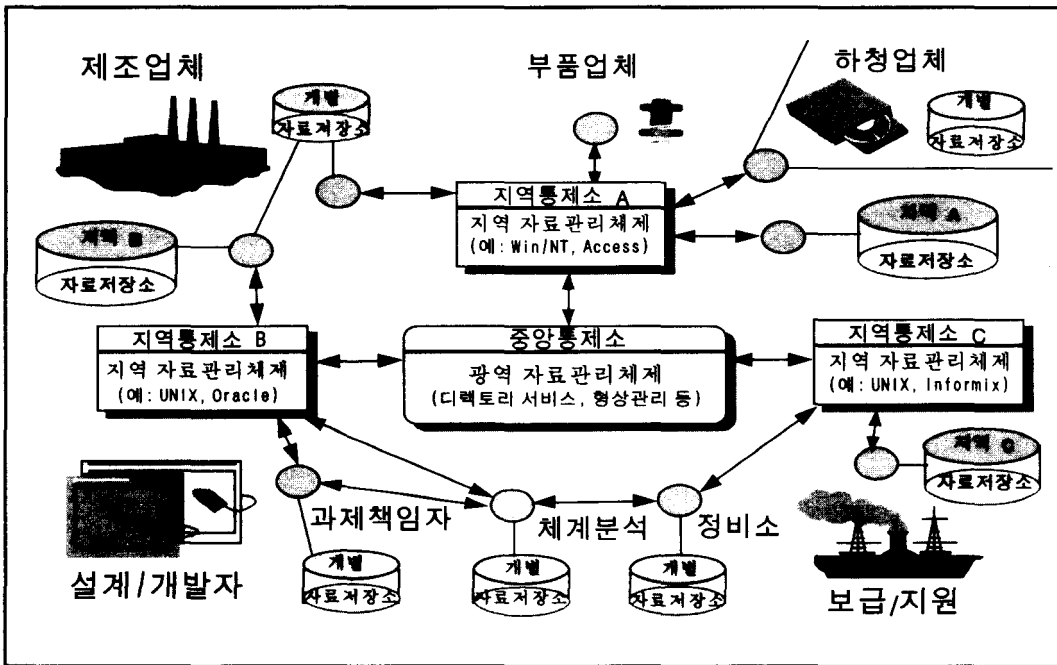
3. IDE 의 기본구조

3.1 미국 국방부의 IDE 구조

미국 국방부의 IDE 과제는 2000년 이전에 그 골격을 완성할 예정으로 West Virginia에 위치한 ManTech Advanced Technology Systems 사를 포함한 10여개사가 참여하고 있으며 현재, 인터넷 기반의 IDE 시제품인 DCN/ICN (Defense/International CALS Network)의 시험운용을 준비하고 있는 단계이다.

IDE의 기본구조는 1994년 7월에 발표된

[WSDB 예비설계 보고서[ManTech, 1994]를 통해 개략적으로 제시된 바 있으며 현재는 컴퓨터 시스템 관점, 자료/정보 관점, 관리체계 관점, 그리고 사용자 관점에서 각각 정의되어 있다.[Kidwell & Brazy, 1996] 컴퓨터 시스템 관점에서의 IDE는 상이한 플랫폼 위에 구축된 정보자원들을 LAN/WAN을 연결하고 3계층, 즉 개별/지역/중앙(local/regional/global) 통제체제를 통해 자료공유를 실현하기 위한 클라이언트-서버 시스템이다. (<그림 2> 참조)



<그림 2> IDE 하에서의 자료관리체계 (자료원: 미국 ManTech 사)

<그림 2>에서 볼 수 있듯이 IDE 하에서 대부분의 자료는 특정 조직체 단위로 관리되는 지역

저장소 (Regional Repository)에 구축되고 개인이나 단위부서 차원에서 빈번히 활용되는 자료들은 활

용성을 높이기 위해 개별저장소 (Local Repository)에 복제된 형태로 유지된다. 중앙통제체제 (Global Control)는 특정 제품 또는 무기체계 단위로 구축되어 물리적으로는 다수의 지역저장소에 위치한 자료들을 논리적으로 통합해 주는 역할을 담당한다. 이러한 3 계층의 자료관리체제는 상이한 플랫폼 위에 구축된 정보자원들을 통합할 수 있는 효과적인 수단으로서 인터넷을 이용한 자료의 공유체제와 유사한 개념이다. 단, IDE는 현재의 인터넷보다 정보통신과 보안 측면에서 훨씬 향상된 품질과 기능을 제공해야 하고, 인터넷에서는 파일단위 자료유통을 지원하지만 IDE에서는 상이한 DB들을 연계한 자료유통을 지원한다.

IDE는 플랫폼의 상이함에서 생기는 문제들을 해결하고 다수의 사람들이 공동작업을 할 수 있게 하기 위해 여러가지 기술적인 문제들의 해결을 필요로 한다. 즉, 지역저장소의 효과적인 자료관리를 위해서는 DB 관리, 보안, 디렉토리 관리, 자료사전 (Data Dictionary) 관리, 버전통제, 프로젝트 관리, 데이터 모델링 등의 서비스가, 중앙통제체제에는 종합적인 디렉토리 관리, 버전제어 및 형상관리와 다중 DB 관리 (Multi-database Management) 등의 서비스가 제공되어야 한다. DB 관리기능은 현재의 관계형 DBMS는 물론 향후의 객체지향(Object-oriented) DBMS, 그리고 DB에 구축된 정형자료와 웹(web)⁴ 방식으로 구축된 비정형

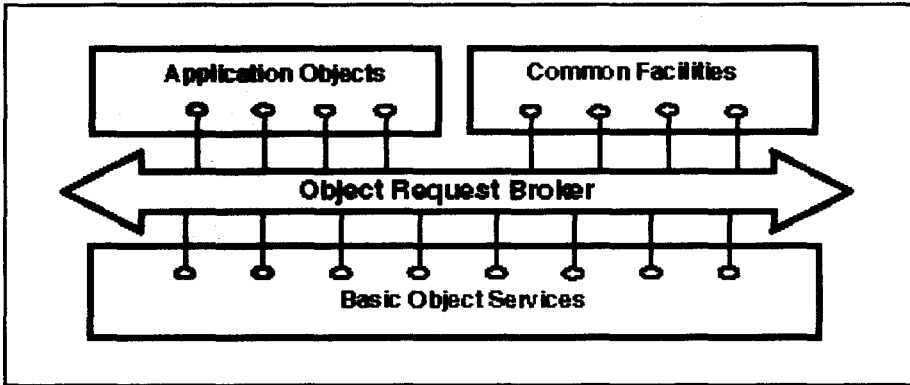
자료들에 대한 통합관리기능이 요구된다. 보안기능은 인가된 사용자의 접근만을 허용하기 위한 인증(authentication), 자료별 보안등급의 설정, 자료자체의 암호화 등을 포함한다. 디렉토리 관리 기능은 산재된 정보자원들의 이름과 실제 저장되어 있는 주소를 연결시켜 주는 기능이며, 자료사전 기능은 각각의 정보자원들에 대한 메타자료(meta-data)--예를들면, 자료자체의 크기, 보안등급--를 관리하기 위한 기능이다. 버전통제는 특정 자료에 변경이 발생했을 때 변경이력을 유지하고 변경결과를 관련 자료들에 전파함으로써 시스템 전체의 일치성을 유지하기 위한 기능이다. 형상관리는 제품의 계층적 분할구조 (WBS: Work Breakdown Structure)에 의한 구성품, 즉 형상항목 (CI: configuration item)들을 식별, 번호를 부여하고 이들의 변경을 검토, 제어하며 형상항목 자체와 변경이력에 대한 정보들을 기록, 유지하는 등의 포괄적 기능이다.

IDE는 소프트웨어 관점에서 보면 자료의 생산/관리/활용에 관여하는 사용자들의 서비스 요청에 대해 이를 직접 처리하는 소프트웨어 (즉, 서버)와 처리된 결과를 사용자에게 보여주는 소프트웨어 (즉, 클라이언트) 그리고 서버와 클라이언트 사이에서 양쪽을 모두 지원하기 위한 미들웨어를 포함하는 계층적 구조를 필요로 한다. 객체지향 기술에 대한 표준화 그룹인 OMG (Object Management Group)의 객체관리모형 표준인 OMA(Object

⁴ 웹(web)이란 World-wide Web의 준말로써 인터넷 상에서 제공되는 다양한 정보서비스 중 전자문서 형식인 HTML (HyperText Markup Language)과 이의 전송규약인

HTTP(HyperText Transfer Protocol)를 이용한 서비스 방식을 가리킴

Management Architecture) [OMG, 1995]는 이와같은 참조) IDE의 기본구조를 뒷받침하고 있다. (<그림 3>

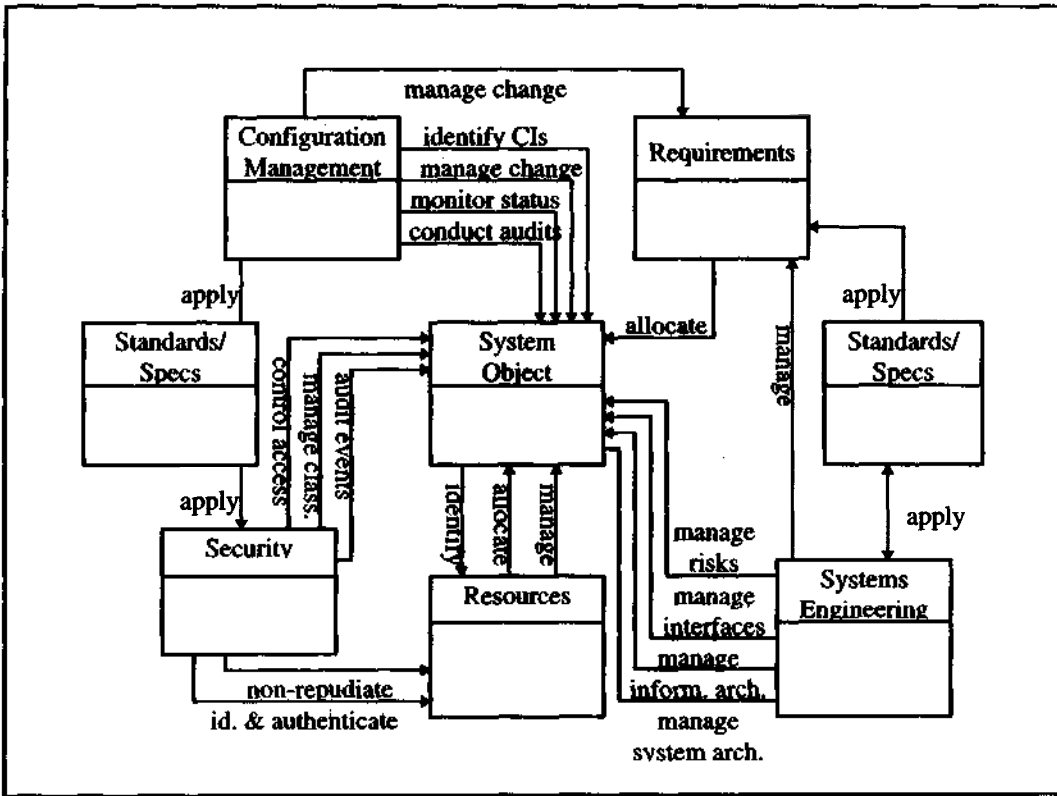


<그림 3> OMG의 OMA 참조모형

<그림 3>에서 'Application Objects'는 사용자가 어떤 작업을 수행하는데 필요한 서비스를 요구하고 처리된 결과를 돌려 받기 위해 작성한 응용 소프트웨어, 'Common Facilities'는 다양한 작업에서 공통적으로 쓰일 수 있는 서비스, 예를들면, 복합 문서처리, 도움말, 규칙처리 기능 등을, 'Object Services'는 모든 객체들에게 기본적으로 필요한 서비스, 예를들면, naming, 논리적 모델링, 물리적 저장, 조회 및 트랜잭션 처리 기능 등을 의미한다. 또한 'ORB (Object Request Broker)'는 application object가 요청한 서비스를 적절한 common facilities와 object services를 통해 처리하고 그 결과를 application object에 돌려 주는 일종의 거간꾼 역할을 하는 소프트웨어를 의미한다. (OMA를 포함한 분산객체컴퓨팅 기술의 구체적인 내용은 4장을 참조바람)

자료/정보 관점의 IDE는 CALS 체제 하에서 필요한 각각의 정보자원들 (즉, entities)과 이들간의 관계들(relationships)로 정의된다. <그림 4>는 데이터 모델링을 위한 CALS의 목시적 표준인 IDEFIX로 표현된 총체적 데이터 모형이다. 그림에서 box는 개체들, 직선과 화살표는 각각 1:1 (one-to-one) 관계, 1:다 (oneto-many) 관계를 의미한다. 'System Object'는 system, subsystem, configuration item, assembly, subassembly, part 등 제품과 구성품들을, 'Requirements'는 각각의 system object의 기능적, 물리적 특성에 대한 요구명세를, 'Resources'는 system object의 개발 및 운용에 필요한 인력, 장비, 도구, 시설, 예산 등의 자원을 의미한다. 'Configuration Management', 'Security', 'Systems Engineering' 등은 IDE의 구축과 운용에 필요한 기능측면의 개체들로서 system object, resources 등에 대한 관리와 제어

를 담당한다. 예를들면, 'Systems Engineering'은 다
수의 'Requirements' 관리와 다수의 'System Object'
에 대한 위험 분석/평가, 개체간 인터페이스 관리,
정보구조 관리 등을 수행한다.



<그림 4> IDE의 총체적 데이터 모형 (자료원: 미국 ManTech 사)

관리체계 관점의 IDE는 자료의 생산, 관리, 유통 관련된 처리방식과 보안기능, 그리고 제품의 개발 및 운용유지 관련 프로세스 자체와 그 과정에서의 검토/승인 프로세스 등 업무처리방식을 포함한다. 즉, IDE 하에서 인쇄문서 중심의 기존 관리체계는 위와 같은 기능들을 중심으로 사실상 재구축된다. 예를들면, 기존 관리체계에서는 자료의 생산자가 그 사본을 필요한 배포처들에 송부함으

로써 동일한 자료가 양쪽 부서에서 중복해서 보관되나, IDE 하에서는 자료의 물리적 이동이 최소화된 가운데 자료에의 접근권한을 가진 사람이라면 누구나 필요한 자료를 즉시 조회, 활용할 수 있는 관리체계가 된다.

사용자 관점의 IDE는 하드웨어, 소프트웨어, 정보통신 프로토콜, 자료의 표현 및 저장형식 등에

구매발음이 없이 필요한 자료 및 정보에 쉽게 접근할 수 있으며, 다른 부서/기관/나라에 있는 사람들과 협업을 할 수 있는 환경을 의미한다. 예를 들면, 특정 제품을 개발 중인 설계자가 IDE 하에서 사무자동화 및 CAD 소프트웨어를 활용해서 문서나 도면을 완성하면 그 내용은 관련자들의 전자적 검토과정을 거쳐 지역저장소에 등록되고 등록된 정보는 지역 및 중앙 통제체계에 구축된 디렉토리 자료사전에 포함된다. 지역저장소에 있는 자료들은 필요 시, 방화벽(firewall)을 통해 외부 사용자의 접근이 봉쇄될 수도 있으나 그렇지 않은 경우 타 기관의 사용자는 자료의 명칭만을 시스템에 알려 줌으로써 원하는 자료를 획득할 수 있게 된다.

미국 국방부가 정의한 IDE 는 <그림 3>의 OMA 와 같은 구조로서 ORB 위에 Knowledge Enhancement, Security, Marketplace Clearinghouse, Marketplace Customhouse, Industry Association, Technology Transfer, Internationalization 등의 서비스들을 얹은 형태이다.

Knowledge Enhancement Services 는 인공지능 기술을 응용한 자료처리의 지능화 서비스들로서 문제영역에 대한 학습, 정보의 검색/종합 등 정보수집, 전자우편, 뉴스 등을 통해 입수된 자료에 대한 여과; 정보자원의 소재에 대한 디렉토리 유지; 시스템 내부 및 외부에서 발생한 각종 이벤트들에 대한 모니터링과 관련자들에 대한 변경사항 통지; 워크플로우의 자동화를 위한 지능적 처리; 표준, 기술, 소프트웨어, 도면, 기타 조직전체 정보자원들의 특성에 따른 분류 기능 등을 포함한다. Security

Services 는 자료 및 컴퓨터 시스템의 보안유지를 위한 것으로서 사용자 식별 및 인증, 접근제어, 이벤트 모니터링, 부인거부, 시스템 가용성 유지 등의 기능을 포함한다. Marketplace Clearinghouse Services 는 다양한 형태의 조직체 내부 및 조직체 간의 거래처리를 지원하기 위한 것으로서 EDI를 위한 주문서, 송장 등 자료의 교환형식 변환; 자금의 이체/지불처리; 제품 카타로그 관리; 광고색인; 납품요청 (RFP/RFQ)의 공고; 관련업체 목록; 암호화 키의 종합통제 ("escrow") 기능 등을 포함한다. Marketplace Customhouse Services 는 각종 정보자원의 품질관리를 위한 것으로서 소프트웨어 제품 및 프로세스의 품질관리; 정보자체의 일치성 유지, 보다 적합한 정보의 추출, 자료사전의 유지 기능 등을 포함한다. Industry Association Services 는 산업간 협력활동을 지원하기 위한 것으로서 관련 기관 및 단체와 개인 등에 관한 정보검색용 디렉토리 유지, 각종 인쇄물 및 전자문서에 대한 조회; 제품의 조달, 제작, 제조, 시험, 운용지원 등 기업활동에서의 모범사례 소개; 고객 및 납품업체로부터의 의견수집 기능 등을 포함한다. Technology Transfer Services 는 기업활동을 통해 축적되는 기술의 관리와 전파를 위한 것으로서 기술개발 관련 연구기관 및 컨소시엄, 수행 중 과제 등에 대한 디렉토리 유지; 기술현황에 대한 모니터링; 제작, 제조, 운용지원 등 관련 기술들의 분류; 기술의 개발자 및 사용자로부터의 의견 수집 기능 등을 포함한다. Internationalization Services 는 기업활동의 국제화를 지원하기 위한 것으로서 국제적 기업활동에 대한

정기간행물, 통계자료; 무역관련 국가별 규제, 관세, 선적, 기술표준, 수입면허, 수출통제, 수출입 전망; 특허, 상표 등 지적재산권 등 관련 정보의 제공기능을 포함한다.

IDE는 미국경우, CALS가 완성되는 2010년경까지 추진될 장기적 과제로서 2000년까지는 각종 시범사업을 통한 개념설계와 요소기술의 확보가 진행될 예정이다. 현재 진행 중인 IDE 시범사업으로는 미국 육군 탱크사령부(TACOM)가 추진 중인 전장기동화체계(CMS: Combat Mobility System)를 위한 종이없는 프로젝트 관리체계 구축계획과, 정보통신업체와 은행들 간의 전자 상거래를 위한 CommerceNet 계획 (<http://www.commerce.net> 참조) 등이 있다.

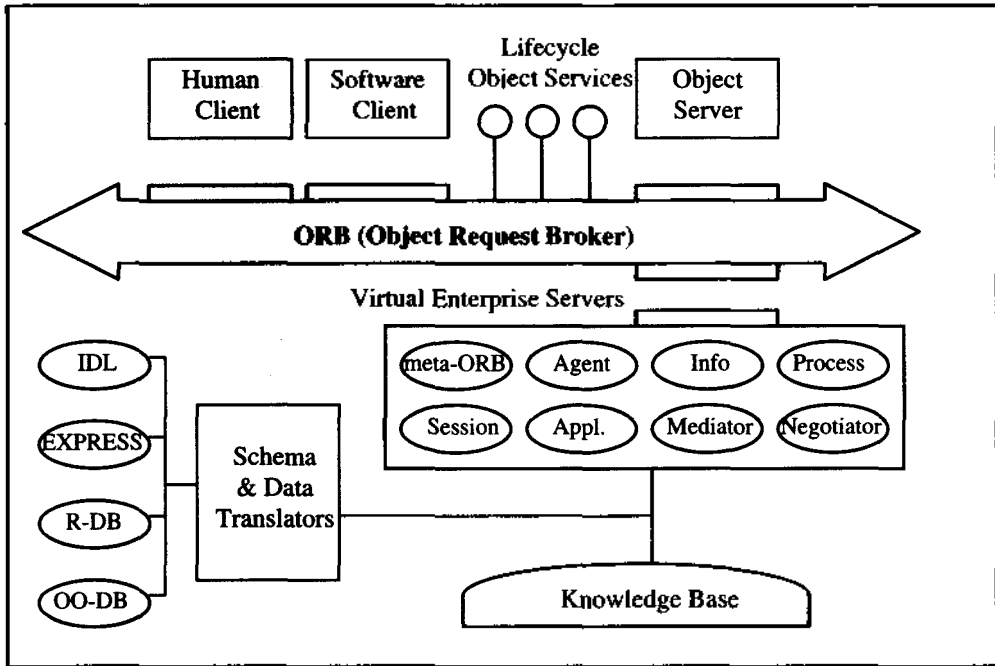
3.2 미국 NIIP의 IDE 구조

NIIP (National Industrial Information Infrastructure Protocol) [NIIP, 1995]는 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 ARPA (Advanced Research Projects Agency)에 의해 국가정보화 기반구조(NII: National Information Infrastructure)의 핵심구성품으로 정의되어 과제화되었으며 미 공군의 Mantech office에 의해 관리되고 있다.

NIIP의 목표는 (1) 미국의 제조업들을 대상으

로 분산환경 하의 상이한 컴퓨팅 자원들을 통합할 수 있는 개방형 기반구조를 설정하고 (2) NIIP 참조구조를 문서화해서 공개함으로써 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.

NIIP는 IETF (Internet Engineering Task Force)의 인터넷 기술, OMG의 분산객체관리 모형 표준인 CORBA (Common Object Request Broker Architecture), ISO의 제품자료 표준인 STEP, 그리고 WFMC (Workflow Management Coalition), ARPA의 I³ (Intelligent Integration of Information), CFI (CAD Framework Initiative) 등의 워크플로우 및 작업관리 기술 등을 통합한 형태가 될 것이다. [그림 5]의 NIIP 참조모형은 OMA의 바탕 위에 I-VE를 위한 NIIP 고유의 common facilities를 얻은 것으로서 국방부의 IDE 참조모형과 유사한 한 형태임을 알 수 있다. 이는 서론에서 지적한 바와 같이 IDE의 자체가 CALS 고유의 것이라기 보다는 미래지향적 컴퓨팅 시스템들이 보다 효율적이면서 효과적인 기업활동을 지원하기 위해 공통적으로 추구하고 있는 컴퓨터, 통신, 소프트웨어 기술의 결집체이기 때문이다.



<그림 5> NIIP의 참조모형

<표 2> NIIP의 구성요소 및 기능

서비스시스템	메타클래스	기능
Project M'gmt	NIIP Desktop	다수의 VE 자원관리를 위한 human interface (I/F)
	Agent	VE 조직구조, 자원관리 등을 위한 non-human I/F
I-VE Services	Internet Tools	WWW 접속, 원격회의, 보안
Knowledge & Rule M'gmt	STEP Services	SDAI (Standard Data Access Interface)의 지원
	VE Monitor	NIIP 프로토콜과 Appl. Objects 의 binding 지원
Task & Session M'gmt	VE KBMS	Active Semantic Network DB
	Session	사용자 인증, 자원배분, 트랜잭션 처리 등
	Workflow	워크플로우 처리
	Data M'gmt	스키마 관리, 버전 및 형상관리, 자료복제 등
	Application	application-to-data, data-to-application 지원
	Access	CORBA 및 non-CORBA objects 간의 접속 지원

Decision	Mediator	semantic ambiguity 의 조정, 정보요약 및 종합
Support	Negotiator	다수 객체들 간의 conflicts 를 조정

<표 2>는 NIIP 를 구성하는 5 개의 서브시스템과 이들을 필요한 서비스별로 분할한 13 개의 메타클래스(meta-class)들의 기능을 요약한 것이다. Decision Support 서브시스템의 하부 모듈인 mediator 는 공유할 자료의 명칭이나 내용, 프로그램 인수 등이 관련된 사용자나 클라이언트 소프트웨어들에서 각각 다르게 정의되어 있는 경우, 이들을 변환없이 그대로 사용할 수 있도록 중간에서 조정하는 역할을, negotiator 는 하나의 트랜잭션을 처리하는데 관련된 다수의 agent⁵ 소프트웨어들 간에 생기는 처리방식 상의 차이를 조정하는 역할을 담당한다.

3.3 IDE 참조모형들의 검토결과

미국 국방부의 IDE 와 NIIP 의 IDE 는 공히 이질적인 분산시스템들의 통합수단으로서 OMG 의 OMA 참조모형을 채택하고 있다. 즉, 다양한 플랫폼들 위에서 운용되고 있는 응용체제들을 미들웨어인 ORB 를 이용해서 상호연동시킴으로서 제품의 수명주기활동에 참여하는 개인 및 조직체를 위한 다양한 정보서비스를 제공하고자 한다. NIIP 경우는 컨소시엄에 참여하고 있는 기관들이 보유하고 있는 기술 및 상품을 기반으로 하고 있기 때문에

국방부의 IDE 에 비해 개방성이 상대적으로 부족한 것으로 판단된다. 국방부의 IDE 과제가 장기적 기술개발 과제라면 NIIP 는 단기 내지는 중기적 문제해결 과제라 할 수 있겠다.

국방부의 IDE 와 NIIP 의 IDE 는 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, VRML⁶), (4) 분산형 및 지능형 DBMS 구현기술 (예: 다중 DB 언어, DB 스키마의 통합/결중, 지식베이스와 DB 와의 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, KQML⁷) 등의 핵심기술들을 필요로 하는 것으로 분석된다.

⁵ 특정 임무의 수행을 위해 자주적으로 작동되는 지능형 프로그램을 가리킴

⁶ Virtual Reality Modeling Language

⁷ Knowledge Query Manipulation Language

4. 통합 데이터베이스 구현기술

4.1 분산객체컴퓨팅 (distributed object computing) 기술

컴퓨팅 환경과 통합의 문제

오늘날 대부분 조직체의 정보시스템 부문은 (1) 보유하고 있는 정보자원들의 다양성과 이질성이 심화되고 (2) 독립적인 운영을 필요로 하는 하드웨어, 소프트웨어들이 지리적으로 분산되어 있으며 (3) 이들 정보자원들의 특성과 상호연관성이 끊임없이 변화하는 상황 하에서 최적의 컴퓨팅 환경을 구축하기 위해 부심하고 있다. 효과적인 조직운영을 위해서는 각종 정보자원들을 통합함으로써 자원낭비와 업무처리 상 비효율을 제거함이 매우 중요하다.

정보자원의 통합은 정보자원들 간의 상호접속성 내지는 상호연동성을 확보하는 방향으로 추진되고 있는데 이는 정보처리 기술측면의 여러가지 어려운 문제들을 야기하게 된다. 상호접속성(interconnectivity)이란 둘 이상의 정보자원들 간에 메시지를 송/수신할 수 있는 수준을 의미하며 상호연동성(interoperability)이란 둘 이상의 정보자원들이 주어진 작업을 상호협력해서 실행할 수 있는 수준을 의미한다.[Nicol 등, 1993] 현재의 정보시스템 기술은 상호접속성의 문제는 해결하고 있으나 완벽한 상호연동성은 아직 제공하지 못한 상태이다.

정보자원들의 통합을 위해 기술적으로 해결되어야 할 과제들로는 정보자원의 (1) 이질성(heterogeneity)에 대처하기 위한 고유명칭 부여, 비밀자료 보호, 공개 가능 자료의 공유, 상이한 표현형식의 변환, 시스템 규모 차이의 조정 (2) 독립성(autonomy) 보장을 위한 회계 및 자원관리 (3) 분산성(distribution)에 대처하기 위한 표준시간 동기화, 자료의 일치성 유지, 장애복구 (4) 변화성(evolution)에 대처하기 위한 버전 및 형상관리, 스키마 변경, 개체간 상호의존성들에 함축된 다양한 의미(semantics) 내지는 메타지식(meta-knowledge)의 모델링 등이 있다.

분산컴퓨팅의 발전단계

분산컴퓨팅이란 하나의 application 을 여러개의 컴퓨터 시스템을 통해 처리하는 방식을 말한다. 클라이언트-서버 컴퓨팅은 분산컴퓨팅의 한 형태라 할 수 있는 것으로서 하나의 application 을 여러 개의 작업(task)들로 나눈 뒤 각각의 작업을 가장 잘 처리할 수 있는 시스템에게 의뢰하고 그 결과를 돌려 받아 종합하는 방식을 말한다.

분산컴퓨팅은 크게 보아 3 단계를 거쳐 발전되어 온 것으로 볼 수 있다.[이동만, 1995] 초기단계는 일련의 메시지를 지리적으로 떨어진 시스템 간에 주고 받을 수 있는 'stream-oriented' 방식으로 TCP/IP 프로토콜을 이용한 ftp, telnet 등의 application 이 그 활용 예이다. 2단계는 원격지의 컴퓨터 시스템에 특정 작업을 의뢰하는 'RPC (Remote Procedure Call)' 방식이다. RPC 방식은 OSF 사의 DCE (Distributed Computing Environment), SunSoft 사의 ONC 및 ONC+ 등의 제품에서 지원된

⁸ (예) 제품, 프로세스, 장비, 도구, 인원, 예산; data, knowledge, decision-support models; 컴퓨터 하드웨어, 운영체제(OS), 프로그래밍 언어, DBMS 등

다.

DCE는 RPC, 분산 디렉토리, 스레드⁹, 표준시 간의 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. 이 방식은 1단계 방식과는 달리 location transparency는 보장하고 있으나 시스템 간의 연동방식이 단순해서 복잡한 프로그램을 작성하기 어렵고 보다 다양한 분산처리 기능을 제공하지 못하는 점에서 한계가 있다.

분산컴퓨팅 기술의 3단계는 앞에서 언급한 분산객체컴퓨팅으로 이 단계에서는 하나의 application을 분산된 시스템들이 공유하는 개념의 상호연동성 보장을 목표로 한다.

분산객체컴퓨팅

분산객체컴퓨팅(distributed object computing)이란 위와 같은 컴퓨팅 환경 하에서의 기술적 문제 해결에 초점을 둔 최선의 패러다임으로서 종래의 분산컴퓨팅(distributed computing)방식이 상호접속성의 보장 수준에 머물렀던 것을 객체지향 컴퓨팅(object-oriented computing)방식의 도입을 통해 상호연동성의 보장 수준으로 발전시킨 것이다.

객체지향 방식은 캡슐화(encapsulation), 클래스 계층(class hierarchy 또는 type lattice), 특성의 계승(inheritance) 등의 유용한 수단을 통해 (1) 현실세계

를 독립된 객체들의 집합으로 묶으로써 보다 자연스러운 모델링과 모듈화된 시스템의 구현을 가능케 하고 (2) 외부 객체와는 한정된 인터페이스를 통해서만 접속이 가능토록 함으로써 information hiding과 시스템의 신뢰성을 높여 주며 (3) 계층화된 클래스를 통해 사용자의 관점에 따른 다양한 추상화(abstraction)를 제공함으로써 문제의 복잡성을 감소시켜 주는 등의 잇점을 갖고 있다. 결국, 분산객체컴퓨팅은 이질적이며 분산된 다양한 정보 자원들을 각각 독립된 객체로 보고 이들 객체들이 표준 인터페이스-에블들면, CORBA의 IDL(Interface Definition Language)-를 통해 상호연동시키기 위한 기술이 된다.

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.[Nicol 등, 1993]

OMA & CORBA

OMA는 제3장의 [그림 3]을 통해 소개한 바와 같이 정보자원들 간의 표준적인 인터페이스 정의언어로서 IDL을 제공하고 wrapper(즉, legacy system을 위한 인터페이스)를 통해 상이한 application들의 통합을 지원한다. 객체의 생성, 소멸, 저장, 이륙부여, 이벤트 통지, 트랜잭션 처리, 동시성 제어 등을 위한 Object Services는 COSS(Common Object Service Specification)로 정의되어 있

⁹ 스레드(thread)란 하나의 작업을 수행하는 프로그램을 여러 개로 분할한 것을 말하며 복수 개의 스레드를 실행시키는 기법을 멀티스레딩(multi-threading)이라 한다.

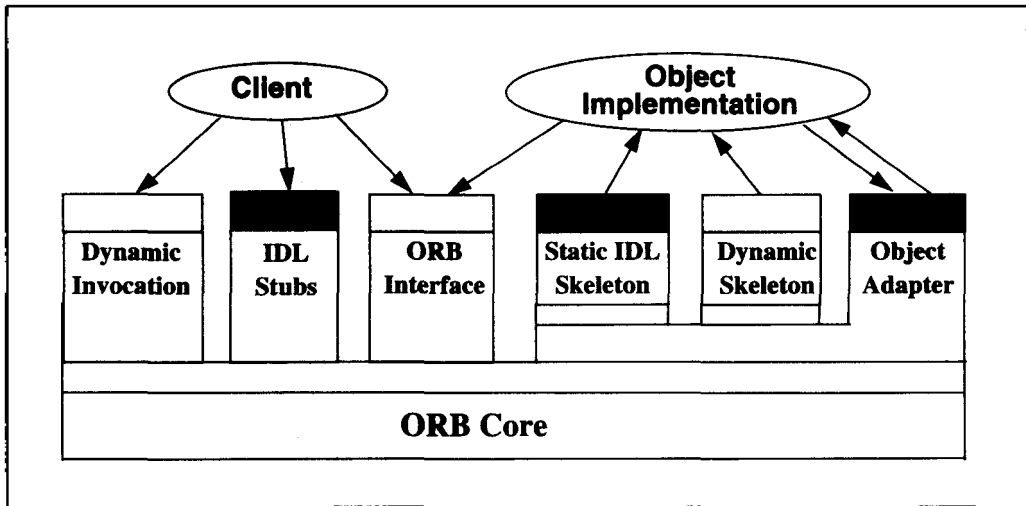
고, ORB 는 CORBA (Common Object Request Broker Architecture)라는 업계 표준규격으로 정의되어 있으며, 이를 지원하는 다양한 제품들이 출하되어 있다.

CORBA 는 지리적으로 분산된 다수의 이질적 객체 시스템들간의 상호연동성을 보장한 가운데 서비스의 요청과 처리결과의 반환과정이 사용자에게는 보이지 않도록 하기 위한 표준규격으로서 '91. 12.에 revision 1.1 이, '93. 12.에 revision 1.2 가, 그리고 '95. 7.에 revision 2.0 이 발표되었다. CORBA 2.0 은 상이한 ORB 들간의 상호연동성을 보장하기 위한 (1) 필수적 요구사항으로 GIOP (General Inter-ORB Protocol)과 TCP/IP 의 지원을 위한 IIOP (Internet IOP)를, (2) 선택적 요구사항으로는 DCE 등과의 연동을 위한 ESIO (Environment Specific IOP) 를, 그리고 기타의 객체 시스템 (예를들면,

Microsoft 사의 COM/OLE)과의 연동을 위한 bridge 들을 정의하고 있다. CORBA 2.0 에서는 또한 객체의 이식성을 높이기 위한 기능들과, C 외에도 C++, Smalltalk 등 프로그래밍 언어와의 결합을 지원하고 있다. [OMG, 1995]

CORBA 를 구현한 제품으로는 IONA 사의 Orbix, DEC 사의 ObjectBroker, Expersoft 사의 PowerBroker, IBM 사의 SOM/DSOM 등이 있다. Orbix 는 분산객체시스템 시장을 주도하고 있는 제품으로서 온라인 거래처리용 도구인 Tuxedo, 객체 지향 DBMS 인 ObjectStore 등과의 연동이 가능한 면에서 활용성이 높은 것으로 평가되고 있다.[박재현, 1996]

<그림 6>은 CORBA 의 구성요소들과 이들간의 인터페이스들을 보인 것이다.



<그림 6> CORBA Components and Interfaces

DME (Distributed Management Environment)

OSF사의 DME는 다양한 객체시스템들에 대한 중립적 기능과 CORBA보다 더 완벽한 객체지향 방식의 지원을 위해 설계되었다. DME의 설계 목적은 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 신속이 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 구성관리 등 기능을 포함하는 점에서 value-added CORBA라 할 수 있다. [Nicol 등, 1993]

4.2 다중 데이터베이스 시스템 (MDBS: Multi-database System)

소프트웨어 및 DB 통합 대안

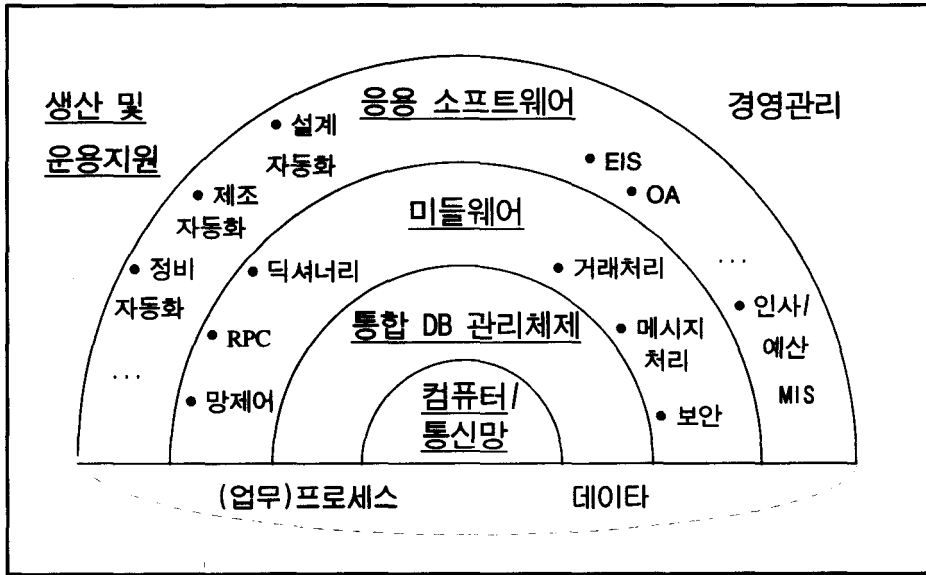
<그림 7>을 일반적인 정보시스템의 계층구조로 볼 때, 이질성과 분산성을 갖는 소프트웨어들의 통합은 크게 보아 세가지 layer에서 고려할 수 있다.

즉, 첫째는 응용 소프트웨어에서 다루는 표현논리 (presentation logic)를 통합하는 것이고 둘째는 미들웨어들이 다루는 제어논리(control logic)를 통합하는 것이며 셋째는 조직전반을 지원하는 소프트웨어에서 다루어지는 자료들을 통합하는 것이다. MS Windows의 GUI (Graphic User Interface)는 표현논리 통합의 한 예이며, OMG의 CORBA는 제어논리 통합의 한 예이다. 자료통합의 예는 이종 소프트웨어 시스템 간의 파일교환 (import/export), CASE (Computer-Aided Software Engineering) repository 등이다. 자료는 표현논리나 제어논리에 비해 그 수명이 길고 안정적이라는 점을 감안할 때, 소프트웨어 통합의 궁극적 목표는 자료의 통합에 두어져야 할 것이다. CALS가 여타의 정보시스템들과 다른 점 중 하나는 이러한 관점에서 프로세스 보다는 자료에 초점을 둔 것이라 하겠다.

상이한 DB 내지는 DBMS들의 통합은 단순하게는 gateway 개념의 소프트웨어를 중간에 개재시키는 방법도 있겠으나 분산환경 하에서는 처리효율의 문제를 감안할 때 바람직한 방법이라 할 수 없다. 분산객체컴퓨팅에서 DB의 통합은 크게 두가지 방향, 즉 전체 시스템을 구성함에 있어서 상이한 DBMS들을 구조적으로 통합하는 방안과 상이한 DBMS의 스키마(schema)들을 논리적으로 통합하는 방안이 있다.[Nicol 등, 1993] 전자는 DBMS를 포함한 상이한 객체시스템들을 통합해 주는 효과적 수단인 CORBA에 의해 가능하나 이는 완벽한 상호연동성을 구현하는데는 한계가 있다. 왜냐하면 CORBA의 초점은 자료에 함축된 의미(semantics) 보다는 자료를 표현하는 형식(syntax)에 있는 점에

서 다수의 작업들을 상호연동시키는데 필요한 충분한 정보의 공유가 어렵기 때문이다. 완벽한 상호연동성의 구현을 위해서는 상이한 DBMS 에 정의

된 스키마에 함축된 의미들을 상호보완하고 조정, 통합하는 과정이 필수적이다



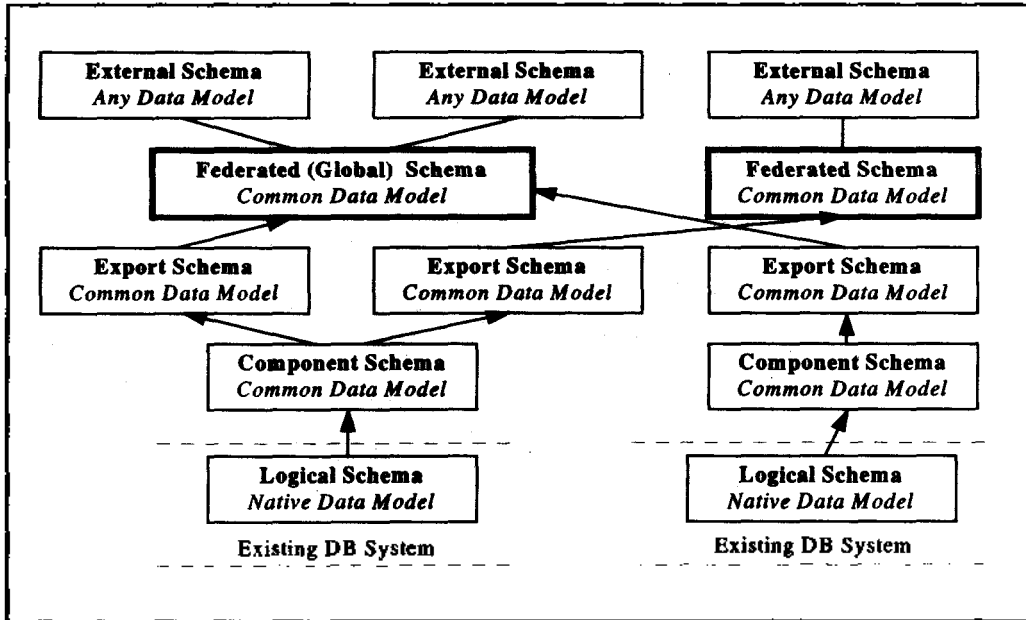
<그림 7> 일반적 정보시스템의 계층구조

다중 DB 시스템 (MDBS)의 의의 및 구조 [Pitoura 등, 1995]

MDBS란 이미 존재하는, 독립적이면서도 이질적인 (autonomous and heterogeneous) DB 시스템들을 통합한(confederated) 시스템이다. DB 시스템들 간의 이질성은 적용된 데이터 모형--예를들면, 계층형, 관계형, 객체지향 데이터 모형-- 또는 집의어의 차이에서 비롯되거나 트랜잭션 처리, 동시성 제어 등을 실행하는 방식 상의 차이에서 비롯된다.

MDBS의 통합은 전통적으로 두 가지 접근방식, 즉 non-federated and federated approaches에 의해 추구되어 왔다. Non-federated approach는 DB 스키마

에 대한 통합보다는 다종의 DBMS를 공통적으로 정의, 조작할 수 있는 multi-DB language를 개발, 활용하는 방식이다. Multi-DB language는 기존의 프로그래밍 언어를 확장하거나 관계형 DB 표준언어인 SQL (Structured Query Language)를 확장한 형태가 된다. Federated approach는 (1) 기존의 상이한 DB들로부터 통합에 필요한 스키마를 각각 추출해서 component schema를 만들고 (2) 이를 통합 대상별로 분할해서 export schema들을 만든 다음에 (3) 이들을 논리적으로 결합해서 federated schema (또는 global schema)를 만드는 방식이다. (<그림 8> 참조)



<그림 8> MDBS의 5계층 Schema

Schema Integration

Federated schema는 중앙 컴퓨터 시스템에 들 수도 있고 각 지역에 복제된 형태로 유지할 수도 있다. 이 때, 문제는 상이한 표현형식에 따라 정의된 export schema 들을 함축되어 있는 의미들의 손실이 없이 잘 결합할 수 있는 논리를 찾아내는 일이다. DB 스키마들을 통합하기 위한 schema integration 논리는 관계형 DB 내에서는 어느 정도 정의된 바 있으나 이종의 DB 시스템들이 대상이 되는 경우는 아직 충분한 연구가 없는 상황이다.

Schema integration 은 일반적으로 (1) 스키마 사전 통합 (preintegration) (2) 통합할 export schema 들에 대한 상호 비교 (3) federated schema 의 구성

(conformation) (4) 스키마 조합 및 재구성(merge & restructuring) 단계들을 거친다.[Batini 등, 1986] 1 단계에서는 component schema 의 표현형식을 표준 데이터 모형 (CDM: Canonical data model 또는 Common Data Model) 형식으로 변환하고 필요 시, 의미의 보충등을 하게 된다. 2 단계는 export schema 상호간의 의미적 모순이나 상충(conflicts)을 보완, 조정하는 단계이며 3 단계는 federated schema 를 구성하고 이의 형식적, 의미적 일관성을 검증하는 단계이다.

스키마 통합 과정에서 발생할 수 있는 의미적 상충의 예로는 (1) 하나의 개념을 두 스키마에서 다른 객체들로 정의한 경우 (identity conflicts) (2) 동

일한 타입에 대해 두 개의 스키마에서 서로 다른 특성을 갖는 것으로 정의한 경우 (schema conflicts) (3) 하나의 개념이 두 개의 스키마에서 다른 의미로 해석되는 경우 (semantic conflicts) (4) 하나의 개념에 대한 기본값을 두 개의 스키마에서 달리 정의한 경우 (data conflicts) 등이 있다. 스키마의 정의나 DB 시스템의 실행 시에 이러한 의미적 상충을 자동적으로 조정하기 위해서는 CDM 자체가 다양한 지식 및 메타지식들을 포괄할 수 있는 것이어야 한다.

표준 데이터 모형의 요건

이중의 DB 스키마들을 효과적으로 통합하기 위해서는 충분한 표현력을 가지면서도 구현의 편의성이 보장될 수 있는 단순함을 가진 CDM의 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 급적 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.[Pitoura 등, 1995]

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.

현 단계에서 통합 DB를 위한 CDM의 요건에 근접하고 있는 것으로는 ANSI의 X3H2 및 ISO의 DB Language Committee가 1997년까지 정의할 예정인 SQL3와, ODMG (Object Data Management Group)가 정의한 ODMG-93 등이 있다. SQL3는 추상자료형 (ADT: Abstract Data Type)에 기반한 확장된 타입 시스템을 정의하고 있으나 자료의 영속성 (persistence)이 table 타입에 대해서만 가능하도록 정의하고 있다. ODMG-93은 기존의 프로그래밍 언어의 기반 위에 영속성과 DB로서의 다른 기능들을 구현해 넣음으로써 다양한 객체지향 DBMS들에 대한 표준적 인터페이스가 될 것을 목표로 하고 있다.

MDBS 연구/구현 사례

지금까지 MDBS에 대한 여러가지 연구와 시험구현이 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] Pegasus 는 functional data model 인 Iris 를 기반으로 해서 HOSQL 이라는 multi-DB 언어를 구현한 시스템이고, ViewSystem 은 VODAK 이라는 데이터 모형과 Smalltalk 언어에 기반한 DB 개발/운영환경으로서 스키마 통합을 위한 다양한 연산자들을 포함하고 있다. UniSQL/M 은 SQL 을 확장한 SQL/M 을 통해 다종의 DBMS 에 접근할 수 있게 한 시스템이다. DOMS 는 FROOM (Functional, Relational, Object-Oriented data Model)이라는 CDM 을 바탕으로 DB 는 물론 하이퍼미디어 시스템과 다양한 application 들을 통합할 수 있게 한 시스템이다. Camot 은 규칙에 기반한 스키마 통합방식을 구현하고 있으며 DB, 전문가 시스템, 지식베이스, 워크플로우 등을 논리적으로 통합할 수 있는 시스템이다.

5. 결론

본 논문에서는 CALS 의 최종목표인 통합 데이터 환경 (IDE: Integrated Data Environment)의 기본 구조를 미국 국방부의 IDE 참조모형과 제조업 중심의 가상기업을 시험구현하기 위한 과제인 NIIP (National Industrial Information Infrastructure Protocol) 의 참조모형을 중심으로 소개하고, IDE 의 중추인 통합 DB 구현을 위한 핵심기술인 분산객체컴퓨팅

(distributed object computing) 기술과 다중 데이터베이스 시스템 (MDBS: Multi-Database System) 기술의 의의 및 연구동향을 분석, 제시하였다.

CALS 의 도입에 있어서 선진국보다 적어도 10년이 뒤진 우리나라 입장에서 선진국 수준을 따라 잡기 위해서는 IDE 의 구현을 위한 연구개발의 조속한 착수가 정책적, 기술적 측면에서 매우 중요한 문제라 본다. 이는 첫째, CALS 의 도입이 국제화를 위한 불가피한 선택이라고 할 때 IDE 는 CALS 의 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. 셋째, IDE 는 세계수준에 근접하고 있는 일부 국내 정보기술들의 전략적 육성을 통한 정보산업의 선진화를 도모해 볼 만한 과제이기 때문이다.

IDE 는 산업체의 실무자들은 물론 학계, 연구계의 다양한 전문가들의 광범위한 참여가 필수적인 과제인 점에서 IDE 의 시범구현을 위한 산, 학, 연의 컨소시엄 구성과 이의 활발한 활동을 지원할 수 있는 정부와 업계의 과감한 투자가 시급히 요구된다.

참고문헌

- [김덕현, 1996] 김덕현, "CALs 개념의 통합 DB 구현기술", *한국 CALS/EC 학회 제1 회 세미나 발표자료집*, 1996. 6. 21, pp. 61-97
- [라오, 1996] 스리나스 라오, "NIIP: National Industrial Information Infrastructure Protocols", *한국 CALS/EC Journal*, 한국 CALS/EC 협회/기술협회, 1996. 7., pp. 48-51
- [박재현, 1996] 박재현, "분산객체 기술을 꿈꾸며", *마이크로 소프트웨어*, 1996. 3. ~ 1996. 6.
- [이동만, 1995] 이동만, "Distributed Objects and Component Integration: CORBA", *KRNET '95 특강자료집*, 1995. 7.
- [Batini 등, 1986] 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
- [Kidwell, 1995] R. Kidwell, *Integrated Data Environment*, *CALS KOREA '95 튜토리얼 교재 및 발표자료집*, 1995. 10.
- [Kidwell & Brazy, 1996] R. Kidwell and J. Brazy, *CALS and EC/EDI (IDE 중심)*, 국방과학연구소 해외전문가 초빙 기술세미나 교재, 1996. 6.
- [ManTech, 1994] ManTech International Co., *Preliminary Integrated Weapon System Database (IWSDB) Implementation Strategy Paper, Version 2*, 1994. 7.
* <http://www.mantech-wva.com>
- [Nicol 등, 1993] 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
- [NIIP, 1995] NIIP Consortium, "NIIP" (Cycle 0, Revision 6), NIIP Consortium, 1995. 11.
* <http://www.niip.org>
- [OMG, 1995] OMG, *The Common Object Request Broker: Architecture and Specification, Revision 2.0*, 1995 * <http://www.omg.org>
- [Pitoura 등, 1995] E. Pitoura, O. Bukhres, and A. Elmagarmid, "Object-orientation in Multidatabase Systems", *ACM Computing Surveys*, v.27, n.2, 1995. 6., pp. 141-195
- [USDOD] U.S. DoD, *CALS Strategic Overview*, U.S. DoD * <http://www.acq.osd.mil/cals/>

저자소개

저자는 '76년부터 현재까지 국방과학연구소에서 MIS 개발, 소프트웨어 기반기술연구 등에 종사해 왔으며 현재는 통합 데이터베이스 중심의 CALS 기술연구를 담당하고 있다. '96년 4월부터 한국 CALS/EC 학회 통합 DB 전문연구위원장을 맡고 있다. 서울대 산업공학과를 졸업('76년)했으며 KAIST 경영과학과에서 공학석사('88년), 공학박사('93년) 학위를 취득하였다. 전공분야는 CASE, Decision Support System 등에 대한 객체지향 모델링이며, 현재 관심분야는 분산 객체관리, 워크플로우 및 지식처리, 인트라넷 등이다.