

CALS 기반 조선정보 통합방안

이 규 열<서울대학교 조선해양공학과>

1. 서언 - 21세기 정보화 시대의 대비 -

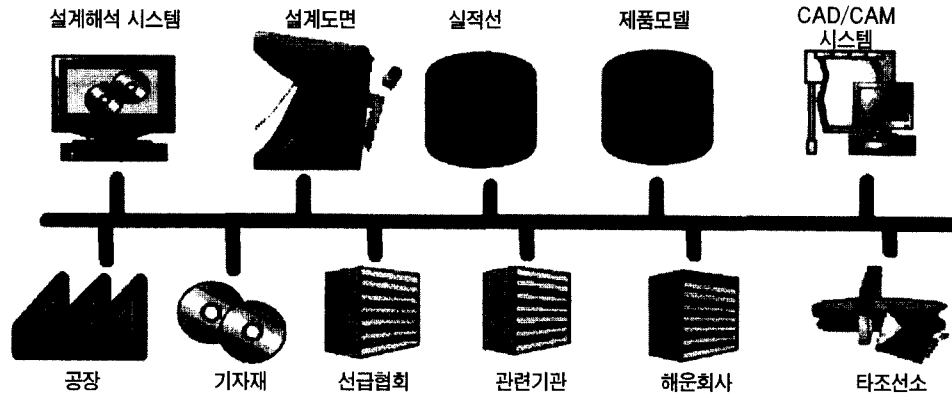
최근의 정보통신기술의 발달과 글로벌 네트워크화가 급속히 진전되면서 각광받고 있는 생산, 거래, 운영 정보 통합화 CALS(Commerce At Light Speed)개념을 산업 및 서비스업 분야에 적용하고자 하는 열풍이 불고 있다. CALS란 제품의 기획, 설계/해석, 자재조달, 제조, 거래 등 제품의 수명주기(Product life - cycle)전반에서 생성, 전송, 저장, 관리, 유통, 활용되는 거래서, 계획서, 설계규격서, 도면, 관리문서, 사용자침서 등 각종 자료들의 흐름을 컴퓨터 시스템과 정보통신망을 활용하여 디지털 정보의 흐름으로 변환하고자 하는 정보체계라고 할 수 있다. 또한 앞으로는 전자상거래(Electronic Commerce)를 통하여 전세계가 하나의 시장으로 형성될 전망이다.

본 글에서는 CALS개념을 기반으로 한 조선 분야의 엔지니어링 정보 통합방안을 모색해 보았다. 첫째, 동시공학적 조선 정보 통합 환경을 고찰하였고, 둘째 CALS 구축을 위한 중요한 표준 중의 하나인 제품정보의 국제표준인 STEP의 개발 현황을 살펴보고, 세계 분산환경에서의 객체간의 정보통합 도구인 CORBA의 활용방안을 검토하였고, 마지막으로 동시공학설계에서 최근에 도입되기 시작하는 AGENT(에이젠트)에 대해서 소개하였다.

2. 동시공학적 조선 정보통합 환경

조선의 특징은 매번 새로이 설계, 건조를 해야 하기 때문에 설계, 생산계획, 생산 업무를 주어진 기간 내에, 그리고 동시병행적으로 처리하여야 한다. 그러나 현재의 설계업무는 건적설계, 기본설계, 상세설계 등 단계별로 진행되고 있으며, 분야별로도 기본계획, 선형설계, 선체설계, 의장설계, 공정설계, 생산 등으로 구분하여 수행되고 있다. 그 원인은 여러 가지 있겠지만, 정보화 측면에서 그 이유를 살펴보면 조선의 단계별, 분야별로 CAD/CAM시스템을 활용하여 업무를 처리하고 있지만 아직도 서로간의 정보교환 수단이 미비하고, 데이터와 전문가의 경험적 지식을 공유할 수 있는 수단과 방법이 미흡하기 때문에 동시적으로 업무를 처리하는데 어려움이 있다. 이 문제를 해결하기 위해서는 분야별, 단계별로 철차적으로, 그리고 분산되어 처리되고 있는 정보를 효율적으로 교환하고 공유하여 동시적으로 업무를 처리할 수 있는 정보통합시스템이 필요하다.

정보를 통합하기 위한 필수적인 조건은 크게 두 가지 사항으로 나누어 볼 수 있다. 첫째, 일반적으로 분산시스템을 구성하는 하위의 시스템들은 서로 원거리에 위치한 상이한 컴퓨팅 환경에서 작동한다. 따라서 네트워크 환경을 통한 시스템들간의 통합 및 원활한 정보 교환을 위한 기반 기술(infrastructure)의 구축이 선행되어야 한다. 둘째, 개개의 시스템들은 입출력을 위한 각기 고유한 규



〈그림 1〉 조선 CALS

약(protocol)을 정의하여 사용하고 있으므로, 시스템간의 유연하고 일관된 정보교환을 위한 방안이 강구되어야 한다.

예를 들어 그림 1에서 보는 바와 같이 네트워크상에 분산되어 존재하는 기기종 H/W 상의 상이한 S/W간의 정보 교환이 관건되고 있으며 이를 위하여 네트워크 상으로 전송하기 위한 데이터 표준 규격의 제정과 그 인터페이스 구축이 필요하다. CAD 데이터 교환규격으로는 STEP, 객체간의 통신을 통한 정보교환 규격으로는 Object Broker, 객체간의 통신을 통한 지식공유 방법으로는 Agent가 도입되고 있다. 이에 따라 프로그래밍 환경도 네트워크를 전제로 하여 변하고 있다.

3. STEP

3.1 개요

제품을 설계할 때 여러 단계를 거치면서 다음 단계로 또는 이전 단계로 feed-back되어 데이터가 전달되어야 할 필요가 발생한다. 예를 들어 초기 설계에서 생성된 형상이 구조 해석을 위한 시스템으로 전달되어 필요한 정보를 제공해야 할 필요도 있을 것이다. 이러한 전달에 있어서 각 단계에서 용도에 맞게 선택되어 사용되고 있는 CAD

system은 서로 다른 데이터 구조를 가지고 있고 그 데이터도 호환이 되지 않는 것이 보편적인 상황이다. 이러한 이 기종간의 모델 데이터를 교환할 목적으로 ISO(International Standard Organization)에서는 표준을 제정하고 있는데, 이를 STEP (STandard for the Exchange of Product model data)이라 한다. STEP에서는 해당 산업 분야에서 요구되는 정보들을 분석하여 정보모델을 만들게 되는데 이것을 AP(Application Protocol)이라고 한다. 이러한 정보모델을 기술하는데 있어서 STEP에서는 EXPRESS라는 정보모델링 언어를 사용한다. STEP을 이용한 응용분야에서의 연구 및 개발은 유럽과 미국 등지에서 여러 프로젝트를 통해 활발히 이루어지고 있으며[1],[2], 현재 조선분야 STEP 분과위원회에서는 일반배치(AP215), 선형(AP216), 배관(AP217), 선체(AP218) 등의 조선 AP들을 개발하기 위해 Building Block이라는 공통단위를 만들어서 시험하고 있는 상황으로서 조선 AP가 국제표준으로 제정되는데는 2-3년 정도 걸릴 것으로 예상된다.

3.2 STEP을 이용한 선박의 구획배치 정보의 CAD 시스템으로의 전달 사례

다음에는 선박계산프로그램 "SIKOB"의 선형 및 구획배치 정보를 범용 CAD 시스템인 "Pro/Engineer"에 전달하는 방법을 STEP을 기반으로 일련의 과정을 처리하는 시험 시스템에 대한 사례를 간략히 소개한다[3].

본 사례에서 구현한 시스템의 구조는 그림 2에 표시한 바와 같이 크게 세 단계로 나뉘어 진다.

첫번째 단계는 STEP의 정보모델링 언어인 EXPRESS를 사용하여 정보를 모델링하는 단계로서, AP203(config control design, STEP의 AP들 중 형상 및 제어에 관련된 Part이며, 응용분야에서의 형상을 표현하기 위하여 사용한다)를 이용하여 선형에 관한 선형스키마를 생성하고, 다시 AP203과 선형 스키마를 이용하여 구획배치에 관한 구획스키마를 생성한다. 여기서 생성된 선형 및 구획스키마는 선박계산프로그램 "SIKOB"의 데이터를 표현할 수 있도록 구성되었다. 이렇게 생성된 스키마들은 ST-Developer를 이용하여 컴파일하여 C++클래스로 변환된다.

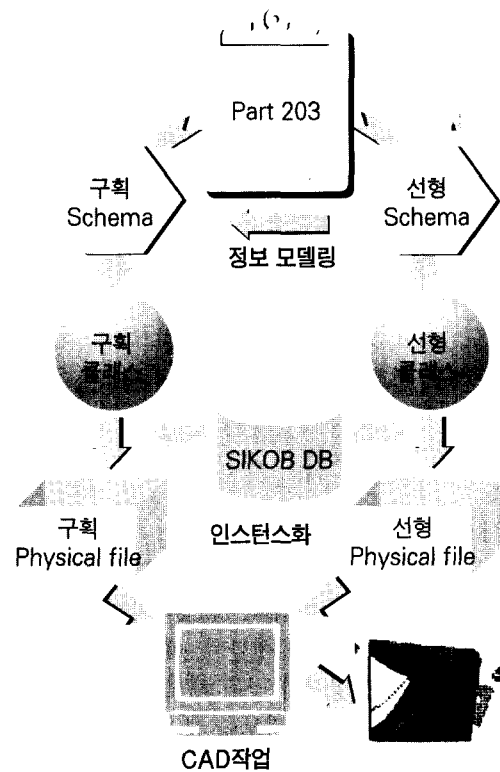
두번째 단계는, 첫 단계에서 생성된 선형 및 구획클래스를 인스턴스화하는 단계이다. SIKOB의 선형 및 구획 계산 결과를 읽어들이, C++프로그래밍을 통해 선형과 구획의 STEP 물리적 파일을 생성한다.

세번째 단계에서는 각각의 STEP 물리적 파일을 범용 CAD시스템 "Pro/Engineer"의 전달하고 CAD 작업을 하는 단계로서 STEP 물리적 파일 형태의 선형 및 구획 데이터가 CAD 데이터로 변환되어 작동 되는 것을 나타낸다.

본 사례에서는 선박의 기본설계시 사용하는 선형 및 구획정보를 STEP 데이터화 하는 과정을 처리하는 시험시스템의 기능 및 작동 모습을 보였다.

향후 연구 방향으로서는 현재 진행중인 선형, 구획, 선체구조, 배관에 관련된 조선

Building Block과 AP215(Ship Arrangement), AP216(Ship Moulded Form), AP218(Ship Structure) 등의 표준화 제정 작업의 진척도에 따라 선박제품데이터를 정보 모델링하고 데이터 전달 및 교환 방법을 표준화/자동화하여 현재 국내 조선소에서 사용하고 있는 전산시스템들 예로서, TRIBON- Initial Design 시스템, SIKOB, CADRA, AUTOCAD, 선체설계의 AUTODEF, TRIBON, Intergraph VDS, 의장설계 분야의 CADDS, Pro/Engineer 등의 이기종 CAD 시스템들간의 정보교환을 원활하게 하여 기획, 설계, 생산계획, 생산 등 조선 분야의 정보 측면에서 생산성을 향상시켜야 한다.

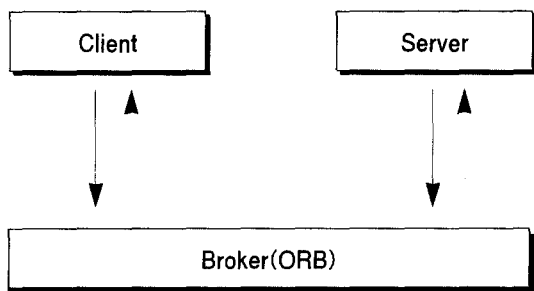


〈그림 2〉 STEP을 이용한 선박 구획정보 전달 시스템 흐름도

4. CORBA(Common Object Broker Request Architecture)

4.1 CORBA란?

CORBA(Common Object Request Broker Architecture)는 OMG(Object Management Group; 1989년 4월에 미국에서 설립된 비영리 기관으로, 객체지향 표준의 제정 및 배포와 기존의 어플리케이션 통합을 위한 표준화 방안 제시 등의 일을 담당하고 있다)에 의해 제안된 객체지향 기술을 기반으로 이종의 분산된 환경하에서 응용 프로그램들이 서로 통합될 수 있는 표준 기술로서 오늘날의 빠르게 변화하는 하드웨어와 소프트웨어 사이의 상호 연동성의 필요에 의해 출현하게 되었다. 1991년에 CORBA 1.1 표준이 제정된 이후 현재 CORBA 2.1이 발표된 상태이다. CORBA는 기본적으로 클라이언트와 서버 사이에 브로커(Broker - ORB(Object Request Broker)에 해당)라고 하는 하나의 새로운 층을 두게 되는 데(그림 3), 이 브로커를 통하여 클라이언트와 서버가 서로 통신을 하게 되고 클라이언트와 서버의 분리가 가능하며 이들은 서로에 대한 구현의 상세한 사항을 알고 있을 필요가 없고, 서로의 인터페이스 정보만 알고 있다면 각각의 구현은 독립적으로 가능하게 된다.



〈그림 3〉 클라이언트/서버 환경에서의 Broker의 삽입

4.2 분산 환경하에서의 선박 초기설계 정보 시스템 구현 사례

본 사례에서는 물리적으로 분산되어 있는 설계 정보자원을 보다 쉽고, 안정되고, 일관된 방법으로 관리하기 위한 방법을 연구하였고, 선박의 정보들을 추출, 교환, 공유하기 위해 그림 4에 나타나 있는 바와 같이 CORBA와 ODBC를 근간으로 한 분산환경하에서 선박 개념설계 시스템, 선박계산 프로그램, 그리고 DBMS를 통합한 선박 초기 설계정보시스템을 구현하였다.[4]

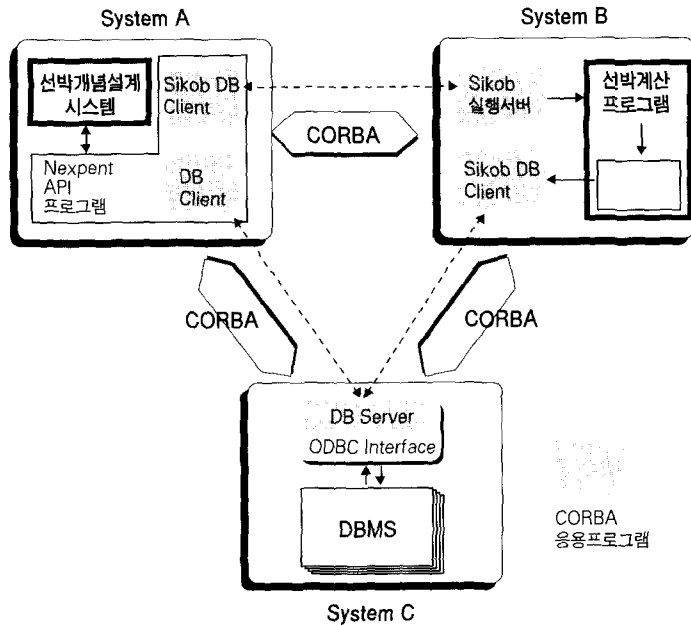
분산환경하에서 설계정보시스템을 구축하기 위해, 각 시스템 사이의 통신과 클라이언트/서버의 역할을 담당하는 CORBA 응용프로그램을 작성하였고, 현재 많은 DBMS에서 지원하는 ODBC 인터페이스를 이용하여 여러 DBMS에 저장되어 있는 설계정보에 접근하는 방법도 보였다. 선박 개념설계 시스템에서 CORBA 클라이언트 프로그램을 통해, 물리적으로 분산되어 있는 선박계산 프로그램 및 DBMS의 CORBA 서버 프로그램을 호출함으로써 DBMS의 정보에 접근하여 필요한 정보를 추출할 수 있도록 하였고, 또한 선박계산 프로그램의 실행결과를 공유하므로써 각 시스템 사이의 정보 교환이 이루어질 수 있도록 하였다.

선박설계를 수행함에 있어 위의 세 시스템들이 하나의 시스템안에 위치하는 것도 가능하나, 본 사례에서 구성한 시스템의 특징은 설계를 위한 각각의 시스템들이 분산되어 있을 경우에도 CORBA라는 매개를 이용하여 통합할 수 있다는 것과 선박 데이터의 관리를 하나의 시스템에서 관리하여 데이터의 저장, 추출, 또는 변경된 데이터의 갱신등이 효율적으로 이루어질 수 있다는 점이다.

5. 에이전트

5.1 에이전트의 개념

에이전트라는 용어는 최근 분산시스템, 동시공학, 협동적 업무처리 분야 등에서 사용되기 시작



〈그림 4〉 CORBA를 이용한 선박초기설계 정보통합

하였고, 각 연구 분야마다 제각기 조금씩 다른 정의와 개념을 사용하지만 여기서는 Stanford 대학교 Computer Science Department의 Logic Group에서 연구하고 있는 Agent-based software Engineering(Heterogeneous 에이전트 시스템)의 관점에 따라 에이전트를 살펴보기로 한다.

현재 소프트웨어는 다양한 분야에서 다양한 정보나 서비스를 제공하고 있다. 하나의 소프트웨어가 독립적으로 사용될 때, 이들 소프트웨어는 사용자에게 유용한 가치를 제공하고 있지만 점차 여러 소프트웨어 간의 협동을 요구하는 일들이 늘어나고 있다. 즉, 프로그램 사이의 정보 및 서비스의 교환이 필요한 것이다. 현재 개발되고 있는 네트워크를 통한 통신기술 및 분산객체기술(CORBA), 다양한 프로그래밍 환경(OOP, ODBC, STEP, GUI 등)의 등장으로 협동하여 일을 수행할 수 있는 시스템을 이전보다 손쉽게 개발할 수

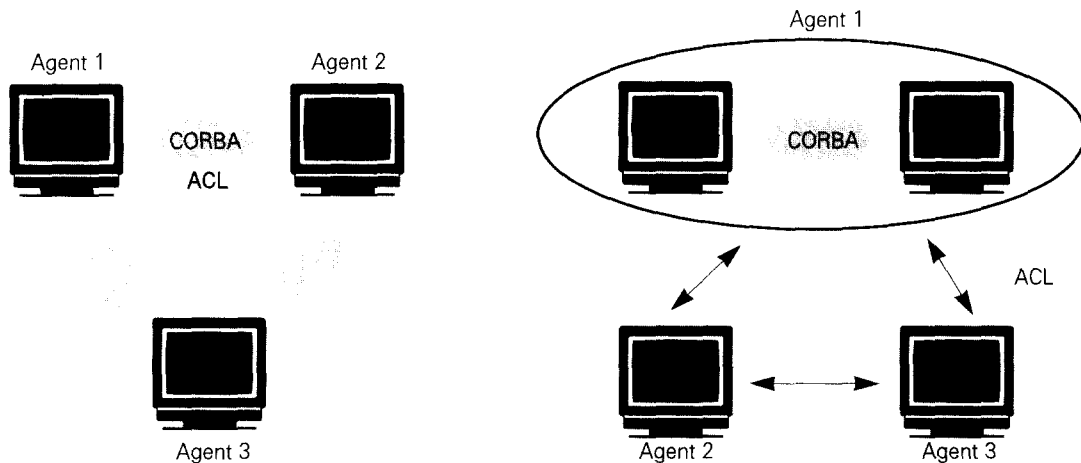
있다. 하지만 이러한 프로그래밍 환경하에서도 다른 사람이 다른 언어로 다른 시점에 개발한 수많은 이질적인 프로그램을 효과적으로 통합할 방법은 없다.

Agent-based software Engineering은 이런 이질적인 환경하에서 보다 쉽게 협동적인 프로그램을 개발하기 위해 연구되었으며 그 핵심적인 사항이 에이전트이다. Agent-based software Engineering의 관점에서 에이전트는 에이전트 통신 언어(Agent Communication Language: ACL)로 다른 에이전트와 통신하는 소프트웨어 컴포넌트이다. 즉, 에이전트는 상황에 따라 적당한 ACL을 주고 받으며 다른 에이전트와 정보 및 서비스를 교환하고 결국 협동적으로 일을 수

행한다. 이때 ACL의 표현력은 에이전트가 수행할 수 있는 일의 범위를 결정짓는 중요한 요소로, Knowledge Query Manipulation Language (KQML)과 Knowledge Interchange Format(KIF)를 각각 외부언어와 내부언어로 사용한다(5, 6, 7). Agent-based software Engineering은 message-driven 방식의 프로그래밍 기법을 분산된 에이전트들(응용 프로그램)에게 확대한 것으로 ACL이 갖고 있는 넓은 지식 표현 능력 덕분에 보다 유연한 능력을 나타낼 수 있다.

5.2 퍼실리테이터

에이전트시스템에서 다수의 에이전트가 서로 통신하기 위해서는 각각의 에이전트가 다른 에이전트들의 주소를 관리해야 하고 또, 각 에이전트에 대한 인터페이스를 알아야 하므로 에이전트개발을 매우 어렵고, 복잡하게 만들 수 있다. 따라서 각 에이전트 간의 통신 문제에 대한 부담을 줄이



〈그림 5〉 Agent와 CORBA

고 인터페이스에 대한 중앙 집중식 관리를 위해서 연방체제(Federation system) 개념을 채용한다. Federation system에서는 에이전트들의 연결을 퍼실리테이터(Facilitator)가 담당하며 각 에이전트들은 초기화 단계에서 자신의 주소와 인터페이스를 퍼실리테이터에 등록함으로써 퍼실리테이터가 에이전트간의 통신을 중계하고 업무를 "monitoring" 하고 "coordination" 하는 역할을 한다. 즉, 에이전트가 ACL을 퍼실리테이터에게 보내기만 하면 퍼실리테이터는 자신에게 전달된 ACL의 내용과 등록된 인터페이스를 보고 적당한 에이전트에게 그 메시지를 보내주고 return되는 결과를 다시 발신자에게 중계해 준다. 이상의 개념에 따라 선박설계 에이전트시스템의 기본 아키텍처를 설계한 사례가 있다(8,9).

5.3 에이전트와 CORBA

일반적으로 전산 시스템의 구축을 위해서는 대상 시스템 모델링, 이에 따른 전산 시스템의 설계, 구현의 세 과정을 거친다.

에이전트 시스템의 개념은 이 세 단계 중 주로 전산 시스템의 설계시에 고려되어야 하는 개념이다. 대상 시스템 모델링에 따라 전산 시스템의 컴

포넌트를 나누고 각 컴포넌트간 인터페이스를 정의할 때 에이전트 개념이 들어오게 된다. 반면 CORBA는 전산 시스템의 설계 단계보다는 구현 단계에 관련된 기술로 볼 수 있다. CORBA는 네트워크 프로그램 구현시의 여러가지 어려운 점들을 개발자가 신경쓰지 않도록 도와준다. 이런 관점에서 에이전트와 CORBA는 상호 배타적인 경쟁관계에 있는 것이 아니라 상호 보완적 관계에 있는 것이다.

에이전트 시스템은 CORBA를 이용하여 프로그램으로 구현될 수 있다. 예로써 에이전트 간의 ACL을 실어나르기 위한 수단으로 CORBA의 기술을 활용할 수도 있고 CORBA로 구현된 분산 프로그램을 에이전트화하여 시스템의 컴포넌트로 활용할 수도 있다. 아래에 두가지 경우를 그림 5에 나타내었다.

물론 같은 기능을 하는 시스템을 CORBA만을 이용하여 구현할 수 있지만, 이것은 에이전트 시스템으로부터 유도되는 유연한 인터페이스, 메시지의 자율적인 전달기능, 메시지 번역기능, 메시지 합성 및 분석기능 등의 장점을 잃는 결과를 낳게 된다.

6. 결론

본 글에서는 CALS개념을 기반으로 한 조선 분야의 엔지니어링 정보 통합방안을 모색해 보았다. 정보를 통합하기 위해서는 네트워크에 분산되어 존재하는 이기종 H/W 상의 상이한 S/W간의 정보교환이 관건되고 있으며 이를 위하여 네트워크상으로 전송하기 위한 데이터 표준 규격의 제정과 그 인터페이스 구축이 필요하다. CAD 데이터 교환규격으로는 STEP, 객체간의 통신을 통한 정보 교환 규격으로는 Object Broker, 객체간의 통신을 통한 지식공유 방법으로는 Agent를 도입하여 조선분야의 통합정보시스템을 구축할 것을 제안한다.

참 고 문 헌

[1] Uwe Rabilen and Uwe Langbecker, "Practical Use Of STEP Data Models In Ship Design And Anaylsis", ICCAS 1997, Yokohama, Japan, Oct. 1997

[2] John Kendall and Knut Hasund , "Product Data Integration And Exchange For Ship Design And Operation", ICCAS 1997, Yokohama, Japan, Oct. 1997

[3] Kyu-Yeul Lee, Jung-Woo Seo and Kwang-Phil Park, "Exchange of Ship Compartment Information Model Using STEP", ICCAS 1997, Yokohama, Japan, Oct. 1997

[4] 김경수, 이규열, "CORBA를 이용한 분산 환경에서의 설계 정보시스템 구축에 관한 연구", 조선학회 1997년도 추계학술대회 논문집, 1997, 11.13

[5] T. Finin, and G. Wiederhold, "An Overview of KQML: A Knowledge

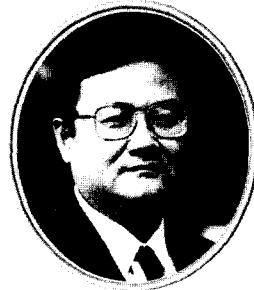
Query and Manipulation Language", Technical Reports, Computer Science Dep., Stanford Unisversity, 1991

[6] M.R. Genesereth, and E.F. Richard, "Knowledge Interchange Format Version 3.0 Reference Manual", Dep. of Computer Science, Stanford University, 1994

[7] <http://cdr.stanford.edu/ABE/>, Agent Based Engineering Research Group Homepage

[8] 이규열, 연운석, 김수영, 윤덕영, "Standard Interchange Format(SIF)을 토대로 한 선박설계 에이전트 시스템 사양개발 및 그 구현 예에 관한 연구, 조선학회 1997년도 춘계학술대회 논문집, 1997, 4.25

[9] 이상욱, 이규열, "에이전트 기반의 시스템 통합을 위한 에이전트 기본 아키텍처에 관한 연구", 조선학회 1997년도 추계학술대회 논문집, 1997, 11.13



이 규 열

- 1947년 10월 12일생
- 1982년 공학박사(독일 하노버공대)
- 1994년 이후 서울대학교 조선해양공학과
- 관심 분야 : 전산선박설계, 형상모델링, CALS
- E-Mail : kyLee@plaza.snu.ac.kr