

엔지니어링 CALS를 위한 컴퓨터 지원 시스템*

김 현**, 이재열**, 김형선**, 한성배**, 박상봉**

Computer Supports for Engineering CALS

Hyun Kim, Jae Yeol Lee, Hyong-Sun Kim,
Sung-Bae Han and Sang-Bong Park

Abstract

As today's manufacturers and business organizations are struggling to compete in the global marketplace, they are concentrating on the efficient use of numerous information on design, development, production, testing, distribution, and maintenance of products throughout their whole life cycles. To meet these organizations' demands on information integration, the CALS has been recently focused as a dominant philosophy. In this paper, we introduce the computer supports for engineering CALS in which the engineering process at an initial phase of product development is simultaneously and collaboratively performed. The proposed system supports the following functions: a virtual prototyping, a distributed collaborative design, and engineering information management. We have conformed to CORBA (Common Object Request Broker Architecture) standard to support interoperability between distributed objects and have used JAVA to support cross-platform and distributed user access to the system on the Web. Under this system, multidisciplinary design teams in engineering CALS environment can collaboratively perform their tasks, share design information and communicate with each other on the Web.

Key Words : CALS, Engineering CALS, Distributed Collaborative Design, Virtual Prototype, Engineering Information Management

* 본 논문은 정보통신부가 지원하는 "CALS 요소기술 개발" 과제의 연구결과임.

** 한국전자통신연구원 전자상거래부

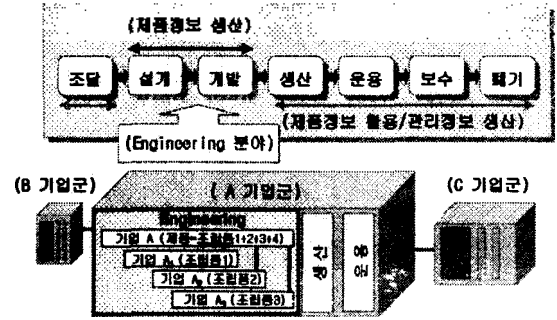
1. 서론

CALS의 어원적인 변천은 최초 Computer Aided Logistic Support로부터 Computer Aided Acquisition and Logistic Support, Continuous Acquisition and Lifecycle Support를 거쳐 현재 Commerce At Light Speed의 개념으로 진전되어 왔으며 최근에는 전자상거래(EC: Electronics Commerce)와의 통합화가 추진되고 있다.

이와 같이 CALS의 개념이 지속적으로 확대되는 과정에서 정보의 표준화, 디지털화, 공유화, Paperless화 등과 같은 CALS의 핵심 내용이 정보기술과 함께 발전되어왔다. 제조기업의 관점에서 CALS는 제품 및 시스템의 조달, 설계, 개발, 생산, 운용, 보수, 폐기라는 전 라이프사이클에 관련된 정보를 표준에 따라 디지털화하며, 이들 정보를 공유 가능한 환경을 구축하여, Paperless를 실현하는 것이며, 궁극적으로는 제품의 원가 절감, 제품의 개발 및 제조 리드타임 단축, 품질 향상을 통해 종합 생산성 향상을 추구하는 정보 통합화 개념이라고 할 수 있다.

제품의 수명주기 중에서 생성되는 정보의 관점에서 볼 때 전 프로세스 중에서 설계에 해당되는 엔지니어링 프로세스는 제품을 구성하는 각종 부품 및 조립품들의 3차원 형상정보, 모델의 특성정보, 제품의 구성정보, 형상의 변경정보, 공학해석 정보, 기술문서정보 등과 같은 각종 엔지니어링 정보가 생성된다. 조달, 생산, 운용, 보수, 폐기 등의 프로세스는 엔지니어링 분야에서 생성한 제품정보를 개개의 프로세스 특성에 맞게 활용하는 사용처로서의 특성을 가지며

또한 관리에 필요한 여러가지의 관리정보를 생산하는 곳이기도 하다 [그림 1-1].



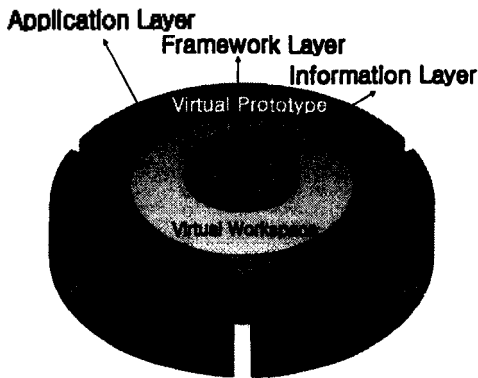
[그림 1-1] 엔지니어링 CALS의 범위

본 논문에서 제시하는 엔지니어링 CALS란, 공학설계 프로세스 상에서 제품정보를 생성함에 있어서, 관련된 정보를 디지털화하며, 이들 정보를 공유함으로써 협동작업을 가능하도록 지원하는 환경이며, 궁극적인 목표는 가상공학(Virtual Engineering)을 실현하는 것이라고 할 수 있다.

본 논문에서는 이러한 엔지니어링 CALS의 많은 이슈들 중에 특히 다음의 사항들을 고려하고, 이를 지원하는 컴퓨터 지원 시스템을 개발하고자 한다.

- (1) 지역적으로 분산된 제품개발 팀이 어떻게 효율적으로 가상모형을 통해 설계를 수행할 수 있도록 지원할 것인가?
- (2) 분산된 환경에서 어떻게 복잡한 엔지니어링 프로세스를 통합적으로 수행할 수 있도록 지원할 것인가?
- (3) 이러한 과정 중에 발생하는 엔지니어링 정보를 어떻게 통합 운용할 것인가?

시스템은 위의 사항을 고려하여 [그림 1-2]와 같은 개념적 구조 하에서 개발되었다. Application Layer는 가상모형을 만들고 이를 이용한 시뮬레이션(가상실험 및 가상생산)을 지원하기 위한 응용 시스템 레벨이며, 본 논문에서는 가상모형에 관련한 내용만을 논의한다. Framework Layer는 엔지니어링 CALS에서의 분산협동설계를 지원하기 위한 프레임워크 레벨로써, 본 논문에서는 프로세스 기반의 협동설계의 개념을 제시한다. Information Layer는 엔지니어링 프로세스 상에서 발생하는 정보를 통합 관리하기 위한 정보관리 레벨로써, 본 논문에서는 기술 정보 관리 시스템의 개발을 논의한다.



[그림 1-2] Conceptual architecture of the system

이들 시스템은 웹(World Wide Web)의 강건한 통신환경 위에서 인터넷 컴퓨팅 기술과 분산객체기술을 이용하여 구축되었으며, 데이터의 교환을 위해 CALS 표준인 STEP (Standard for Exchange of Product Model Data) 과 SGML/XML(Standard Generalized Markup Language/eXtensible Markup Language) 등의

국제표준을 활용하였다.

본 논문의 구성은 다음과 같다. 제 2 절에서는 분산환경 하에서의 특징형상 모델링 방법을 제시하고 이를 바탕으로 한 가상모형 기술 개발에 관해 기술한다. 제 3 절에서는 프로세스 기반의 분산협동설계 기술 개발에 관해 논의한다. 제 4 절은 엔지니어링 CALS 환경에서 제품개발 과정 중에 발생하는 엔지니어링 정보의 통합 관리를 위한 시스템 개발에 관련하여 논의한다. 제 5 절은 본 시스템의 적용예를 제시하며, 제 6 절에서 결론을 맺는다.

2. 가상 모형

2.1 가상모형의 개요

엔지니어링 CALS 환경에서 제품 개발은 순수하게 컴퓨터 모델, 즉 가상모형(virtual prototype)을 이용하여 이루어진다. 기존 2차원 도면 생성을 위한 CAD시스템이 3차원 솔리드 모델러로 대체되며, 나아가 글로벌 분산환경 하에서 작업 수행을 지원하도록 연구 개발이 진행되고 있다[1,2]. 제품 개발과정에서 모든 참여자들이 똑같은 제품과 하드웨어 장비를 사용하는 것은 바람직하지 못할 뿐 아니라 비효율적인 환경이 될 수 있다. 따라서, 효과적이고 올바른 가상모형을 위한 작업을 위해서는 각 소프트웨어들이 모듈화를 이룩해야 하며 네트워크상에서 서로의 형상정보를 교환할 수 있어야 한다.

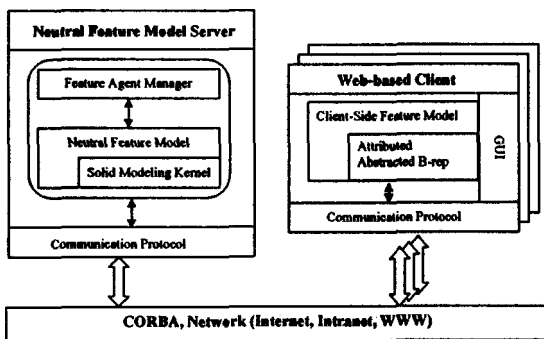
이와 같이 엔지니어링 CALS의 개념 하에서 설계, 가공 및 생산과 관련된 일들을

동시에 처리할 수 있는 방법론 개발에 많은 연구가 진행중이다. 현재까지는 이와 관련하여 CORBA를 이용한 STEP 데이터의 가시화[3,4], 웹 기반 가공 서비스[5], 표준 커뮤니케이션 인터페이스[1,2], 분산환경 하에서 설계 및 프로세스 통합[6] 등의 많은 연구결과들이 있었다. 하지만, 이러한 환경변화에 부합되고, 설계과정상의 여러 문제점을 동시에 해결하기 위한 구체적인 방법론에 관해서는 여전히 많은 연구가 필요하다.

본 장에서는 이러한 해결 방법중의 하나로 분산환경 하에서 특징형상 모델링 방법론을 제시하고 이를 바탕으로 한 가상모형(Virtual Prototyping)기술 개발에 관해 설명한다.

2.2 시스템 구조

웹 기반 가상모형 시스템의 구조는 [그림 2-1]에서와 같이 다음 3가지로 구성된다.



[그림 2-1] 가상모형 시스템의 구조

- 1) 중립특징형상 모델링 서버 (Neutral Feature Model Server)
- 2) 웹기반 모델링 클라이언트(Web-based Client)
- 3) CORBA 기반 표준 인터페이스

중립특징형상 모델링 서버는 분산환경하에서 여러 어플리케이션들에게 공통된 모델링 서비스를 제공하는 역할을 한다. 이는 다음과 같은 3가지의 구성으로 이루어진다.

- 1) 특징형상 에이전트 관리자(Feature Agent Manager)
- 2) 중립 특징형상 모델(Neutral Feature Model)
- 3) 솔리드 모델링 커널(ACIS)

에이전트 관리자는 분산 메타 오브젝트로서 서버에 접속한 모든 클라이언트를 관리한다. 각각의 에이전트는 접속된 개개의 클라이언트의 모든 요청에 따른 서비스를 제공한다. 중립특징형상 모델은 통합된 특징형상 모델링 기능을 제공하는 데이터베이스로서, 솔리드 모델링 커널을 기반으로 구현되어 있으며, 형상 요소의 고유이름을 관리하는 기능을 보유하고 있다. 고유이름 관리기법은 특징형상 간섭 문제 해결, 동적인 특징형상 모델 갱신(Update), 속성(Attribute) 기능 등을 제공한다. 따라서, 각 클라이언트는 중립특징형상 모델로부터 필요한 정보를 필터링하여 특정 영역에 적합한 모델을 구축할 수 있다.

중립 특징형상 모델을 기반으로 한 특징형상 모델링, 가공특징형상 인식 및 변환과

같은 어플리케이션은 네트워크를 통해서 분산 환경에서 이루어진다. 따라서, 이러한 어플리케이션들을 제대로 구현하기 위해서는 각 클라이언트는 1) 명료하고 정확한 형상 표현, 2) 신속한 디스플레이 및 네비게이션, 3) 사용자 인터페이스 등을 제공하여야 한다. 뿐만 아니라, 서버의 모델과 클라이언트의 모델의 일치성을 유지시킬 수 있어야 한다. 따라서, 클라이언트는 크게 두 모델에 기반을 두고 있다.

- 1) 클라이언트 기반 모델(Client-Side Feature Model)
- 2) *Attributed Abstracted B-rep (AAB)*.

앞에서 언급한 것처럼 중립 특징형상 모델은 여러 어플리케이션을 지원할 수 있는 공통된 모델인 반면, 클라이언트 기반 모델은 각 응용분야에 적합한 모델이며, 본 연구에서는 특징형상 모델링에 관련된 모델을 의미한다. 특히, AAB는 기존의 B-rep 정보를 단순화시킨 모델이며 다음의 목적을 위해서 이용된다

- 1) 서버와 클라이언트 사이에 고유이름 유지 (Naming Consistency Maintenance),
- 2) 서버와 커뮤니케이션 최소화
- 3) 실시간 특징형상 모델링을 구현하기 위한 네트워크 지연(Delay) 최소화.

따라서, 클라이언트 상에서 많은 프로세스들이 독립적으로 행해질 수 있다.

CORBA 기반 표준 인터페이스는 분산 환경하에서 작업이 이루어 질 수 있도록 서버와 클라이언트의 교량역할을 하는 미들웨어다. 제시된 인터페이스는 CAM-I AIS 방법에

기반을 두고 있다. 따라서, 여러가지 솔리드 모델링이 가능하며, 특히 특징형상 모델링에서 필수적인 속성(Attribute) 정보를 저장할 수 있도록 확장되었다.

이러한 시스템 구조하에서 이루어지는 특징형상 모델링 과정은 다음과 같다. 우선 사용자는 클라이언트 GUI를 통하여 특징형상 라이브러리로부터 특징형상을 생성한다. 그런 후, 특징형상에 관련된 여러가지 파라미터 값을 표준 인터페이스를 통하여 서버에 보낸다. 만약 보내진 값들이 유효하면, 서버에서는 그에 해당되는 솔리드를 생성한다. 특히, 생성된 솔리드의 각 형상 요소들을 속성 형태로 고유이름(Generic Name)을 부여한다. 또한, 생성된 모델은 AAB 형태로 클라이언트에 전송된다. 클라이언트에서는 전송된 AAB를 바탕으로 여러가지 프로세스를 할 수 있다. 사용자는 위의 과정을 반복하여 원하는 모델을 분산환경에서 만들 수 있다.

가상모형을 제작하기 위해서는 앞에서 언급한 것처럼 설계된 부품들을 조립해야 한다. 여기에는 두가지 방법이 있을 수 있는데 하나는 기존의 CAD 시스템에서 설계한 후 조립체 모델을 Import하여 제작할 수 있는 방법이 있겠고, 또 하나는 앞에서 설명한 웹 기반 가상모형 시스템에서 형상을 제작하고 이들을 조립하는 방법이 있다. 본 연구에서는 현재 기존의 CAD시스템에서 Import받은 후 가상모형을 제작하는 방법을 개발하였으며, 후자도 또한 동시에 개발하고 있다. CAD시스템에서 데이터를 Import할 때 여러가지 표준이 있을 수 있으나 본 연구에서는 CALS 표준으로 채택된 STEP

AP203을 표준 데이터로 설정하고 있다.

시스템 아키텍처는 [그림 2-1]과 유사하며 단 이들간의 관계 및 어셈블리 구조를 또한 포함하고 있다. 예를 들면 [그림 2-1]에서 설계된 부품은 어셈블리 구조 하에서 각 Leaf 노드의 형상정보에 해당되며, 각 Leaf 노드는 형상정보 이외에 상위 노드와의 관계정보 역시 포함하고 있다.

가상모형제작 시스템에서는 여러가지 기능을 분산환경 하에서 수행할수 있다. 예를 들면, STEP AP203데이터의 가시화 및 네비게이션, Interference Checking, Cross Sectioning, Assemblability Check 등을 제공하고 있다.

2.3 웹 기반 가상모형 시스템 구현

제시된 방법은 클라이언트/서버 구조를 지니고 있으며, 서버는 C++로 구현되었으며, Solid Modeling Kernel은 ACIS를 기반으로 하고 있다. 클라이언트는 Java로 구현되었으며 Rendering부분은 Java3D로 구현되었다. 클라이언트와 서버의 CORBA 기반 표준 인터페이스로 Visibroker가 사용되었다.

[그림 2-2]는 시스템 구현을 도시하고 있다. 모델링 과정은 우선 사용자가 웹 서버에 접속하여 Java Applet을 다운로드 받는다. 이와 동시에 Applet상의 클라이언트는 NetVP 서버와 연결된다. 연결이 이루어지면 사용자는 가상모형을 만들 수 있다. 우선 데이터베이스로부터 STEP 데이터(STEP physical file)를 입력받으면 NetVP 서버의 STEP Adaptor는 어셈블리 구조 및 형상정보를 추출한다. 이 정보는 AAB형태로 NetVP

클라이언트로 전송된다. 이 과정이 진행되면 사용자는 웹 상에서 여러가지 분석을 할 수 있다.

3. 분산협동설계

3.1 분산협동설계의 개요

엔지니어링 CALS 환경에서 기존 도면작업이 가상모형에 의해 대체됨을 논의하였으며, 본 장에서는 엔지니어링 CALS 환경에서의 협동작업에 관하여 논의한다.

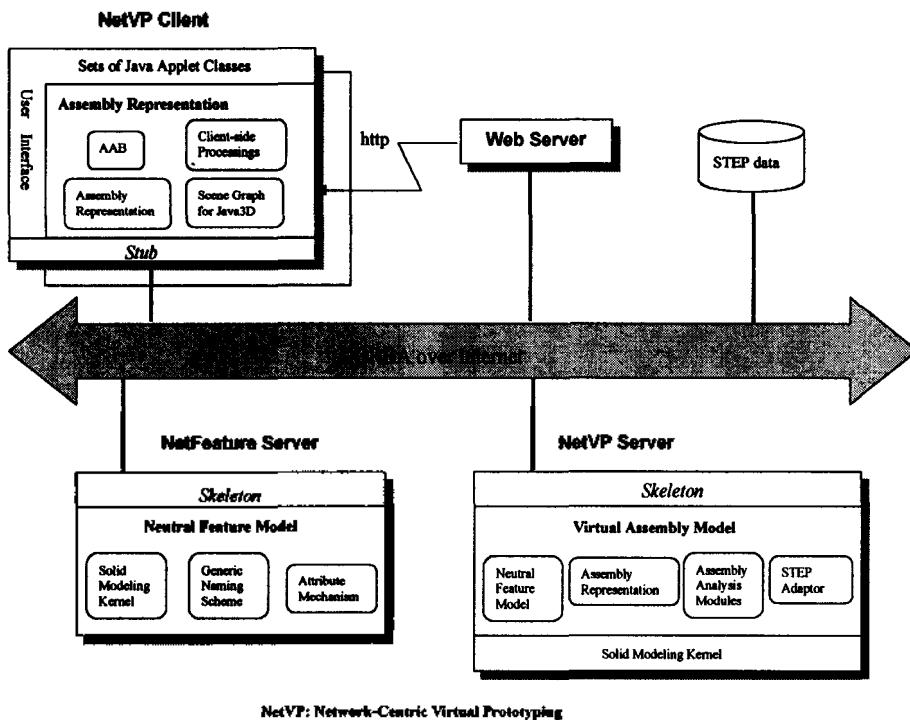
가장 단순한 형태의 협동작업은 정보를 원하는 측에서 상대방에게 Email 등을 이용해 정보를 요청하고 이 결과를 Ftp를 통해 전송받는 방법이다. 그러나 보다 진보된 형태를 고려한다면 공통의 데이터베이스를 구축하고 엔지니어링 프로세스에 참여하는 모든 관련팀이 STEP 등의 표준을 통해 상호 정보를 교환하는 방법을 생각할 수 있다 [7,8]. 이러한 환경은 협동설계를 위해 매우 중요하지만 엔지니어링 프로세스 자체의 정보가 없기 때문에 서로의 공학적 의도를 전달하기가 어려운 단점이 있다. 가장 이상적인 협동작업의 방법은 각 관련팀 또는 제작사가 공통의 데이터베이스를 통해 엔지니어링 정보를 공유할 뿐만 아니라 서로의 엔지니어링 프로세스를 공유하고 자신의 의도를 전달하면서 컴퓨터 지원시스템을 통해 자동으로 프로세스를 진행해 가는 것이다. 각 작업팀은 설계목적을 달성하기 위해 수행해야 할 자신의 작업에 대한 프로세스를 명시적으로 기술하고, 이를 다른 관련팀과 공유

하면서 협동적으로 프로세스를 수행해 간다. 이것이 본 논문에서 제시하는 분산협동설계의 기본 개념이다. 즉 설계 프로세스가 중심이 되어 설계가 진행되며, 이 때 컴퓨터 지원 설계 시스템들은 프로세스 상의 작업을 수행하기 위한 수단이고 이들 시스템으로부터 얻어지는 데이터는 설계목적을 달성하는 설계정보이다. 이러한 개념은 설계 프로세스에 관련된 모든 설계팀이 전체 목적을 공유하면서 이 목적을 달성할 수 있는 컴퓨터 지원 시스템과 설계정보들 간의 연관관계 및 설계팀 간의 분산된 전문지식을 효율적으로 공유할 수 있도록 한다.

3.2 분산협동설계 지원 시스템의 구조

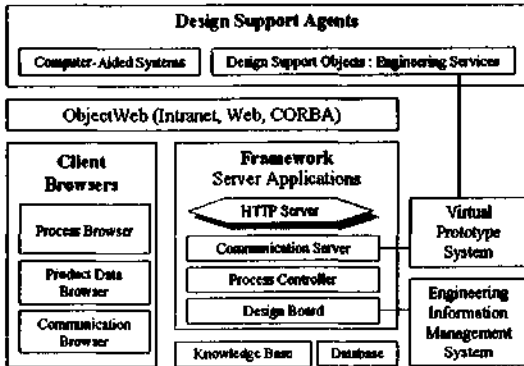
[그림 3-1]은 앞서 언급한 개념을 구현하기 위한 분산협동설계 시스템의 개념적인 구조를 보여준다.

시스템은 크게 설계 지원 에이전트들(Design Support Agents)과 이들이 상호 협동적으로 설계를 진행하고 설계 프로세스를 통합할 수 있는 하부구조(Framework) 및 이때 발생하는 정보를 관리하기 위한 통합 데이터베이스 및 지식베이스로 구성되며, 사용자는 웹 기반의 브라우저(Browser)를 통해 분산협동설계 지원 시스템의 클라이언트 응용 프로그램에 접근한다. 설계 지원 에이전트라는 것은 협동설계 프로세스에 참여하여



[그림 2-2] 가상모형 시스템의 구현

특정 문제를 해결하는 실질적인 프로그램 또는 설계를 행하는 설계자로 정의한다. 하부구조는 실제 협동설계를 진행시키는 엔진으로써 설계판, 프로세스 제어기 및 커뮤니케이션 서버 등으로 구성된다. 클라이언트 브라우저는 설계자가 자신의 설계문제를 명시적으로 기술하거나 기술된 프로세스의 수행을 모니터링하면서 설계를 진행해가기 위한 사용자 인터페이스를 갖는 응용 프로그램이다.



[그림 3-1] 분산협동설계 시스템의 구조

분산협동설계 지원 시스템의 미들웨어는 가상모형 시스템과 마찬가지로 CORBA 표준을 이용하였다. 그러나 실제의 공학분야의 설계 지원 에이전트를 모두 CORBA 분산객체로 만들 수는 없다. 따라서 본 논문에서 설계 지원 에이전트는 CORBA 분산객체로써 분산협동설계 지원 시스템에 의해 자동으로 바인딩되어 수행되는 프로그램과 CORBA 분산객체로 구성할 수 없고 설계자에 의해 호출되어 독립적으로 수행되는 프로그램으로 구분하였다. 앞장에서 설명한 가상모형 시스템은 본 시스템에 바인딩되어

네트워크 상에서 서비스를 제공하는 CORBA 객체로 존재한다.

한편, 분산환경의 이질적인 컴퓨터 환경에서 Platform에 독립적인 시스템을 구현하기 위해 Java언어를 사용하였다.

3.3 협동설계 프로세스의 표현 및 실행

3.3.1 설계 프로세스의 표현

본 논문에서 제시한 분산협동설계에서는 설계에 참여하는 각설계팀이 자신의 설계 프로세스를 전체 설계 목적에 부합하도록 명시적으로 정의하고 이를 통해 설계를 진행한다. 이 때 서로 다른 분야의 설계팀이 어떻게 자신의 설계 프로세스를 컴퓨터 상에 표현할 것인가가 매우 중요하다. 본 논문에서는 설계 프로세스를 설계목적만을 만족시키기 위해 수행되는 작업과 이로부터 산출되는 데이터의 관점에서 고려하였다. 즉, 설계 프로세스는 어떤 상위 작업으로부터의 입력을 받아 임의의 작업을 거쳐 출력으로 변환되는 연속된 과정으로 볼 수 있으며 이를 프로세스 흐름 그래프로 표현하였다[9]. 또한 복잡한 엔지니어링 프로세스를 정형적으로 표현한다는 것은 목적분해(Goal Decomposition)를 통해 프로세스의 작업들을 지정함으로써 이루어진다. 설계초기 단계에서는 개략적이고 추상적인 작업이 정의되며 설계가 진행됨에 따라 보다 구체적인 작업으로 분해된다. 이와 같은 프로세스의 분해는 최종적으로 컴퓨터 지원 시스템에 의해 수행될 수 있는 구체적인 작업이 정의될 때까지 진행된다. 따라서 이러한 목적분해를 지원할 수 있도록 작업노드를 추상작업(Abstract task)과 수행작업(Atomic task)으로 분류하였다. 수행작업이라는 것은 더 이상

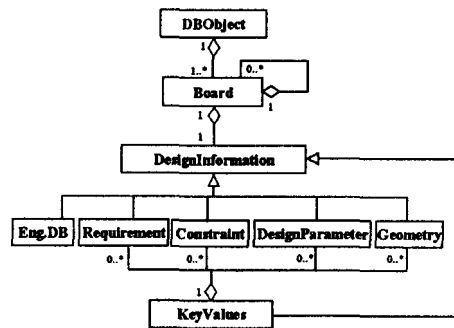
분해될 수 없는 최종작업으로써 그래프 상에서 Terminal Node로 표기된다. 수행작업은 실제 응용 프로그램 등이 호출되어 작업이 수행되는 구체적인 작업이다. 이에 반해 추상작업이라는 것은 다시 하위의 구체화된 작업을 갖는 Non-terminal Node이다. 즉 추상작업은 수행되어야 하는 방법론만을 표현하는 비구체적인 상태로써 또 다른 추상작업이나 수행작업들로 다시 분해된다. 따라서 본 논문에서는 이러한 분해작업을 위해서 임의의 그래프를 상세 수준의 그래프로 변환하기 위한 수단을 제공하였다[7]. 이러한 규칙을 이용하여 다양한 설계대안 (Alternatives)들을 함께 정의할 수 있다.

3.3.2 설계 프로세스의 수행

앞서 설명한 표현방법에 의해 정의된 프로세스는 시스템의 서버 프로그램인 프로세스 제어기를 통해 수행된다. 프로세스 제어기는 설계자가 초기 프로세스를 시작하거나 또는 기존에 진행 중인 프로세스에 참가할 때 프로세스 인스턴스를 생성하여 넘겨준다. 프로세스 제어기는 프로세스의 진행을 지속적으로 추적하여 협동설계에 참여한 설계자가 설계 진행을 모니터링할 수 있게 한다. 프로세스 제어기는 추상작업의 경우에는 최하위 수행작업에 이르기까지 새로운 하위 프로세스를 생성해 간다. 최종 수행작업의 경우 이 작업에 연결된 설계 지원 에이전트를 호출한다. 작업이 완료된 경우 그 결과를 설계판이라고 하는 데이터 저장소에 저장한다. 설계판은 협동설계에 참여하여 실제 문제를 해결하는 컴퓨터 프로그램들이 공유해야 하는 정보를 저장하고 있는 저장소이며, 모든 설계 참여자들이 공통으로 이용하는 핵심정보를 저장한다. 설계판은 다

음 절에서 논의할 기술정보관리 시스템과 연계되어 운용된다. [그림 3-3]은 이러한 요건을 만족하는 설계판의 구조를 보여준다.

프로세스 제어기는 설계판에서의 변화에 의해 상충성을 감지하거나 일관성을 체크하고 작업이 실패하거나 설계자에 의해 재설계를 요청받는 경우 Backtracking을 수행한다.



[그림 3-3] 설계판의 객체 모델

설계가 진행되는 중에 작업간 상충성이 도출되어 관련 설계자들 간에 실시간 협동작업이 요구될 때 해당 설계자는 커뮤니케이션 세션을 생성하고 관련 설계자는 세션의 구성원으로 참여하게 된다. 세션을 생성한 설계자는 현재 설계 모델을 Whiteboard에 올리고 관련자들은 이를 공유면서 마크업 기능을 이용하여 필요한 의사교환을 한다. 이러한 과정은 커뮤니케이션 서버와 앞에서 설명한 가상모형 시스템에 의해 이루어진다. 커뮤니케이션 서버는 다수의 클라이언트 애플릿을 통하여 관련 작업을 수행 중인 설계자들이 설계모델을 공유하고 대화를 할 수 있도록 해주는 통로로서 CORBA 서비스인 이벤트 서비스(Event Service)를 이용하여 구현되었다.

본 시스템은 엔지니어링 CALS의 구현을

위한 가상제품개발 환경의 프레임워크로서의 역할을 하며 제품정보 및 프로세스 정보를 공유함으로써 보다 고도화된 분산협동작업을 지원한다.

4. 기술정보 관리

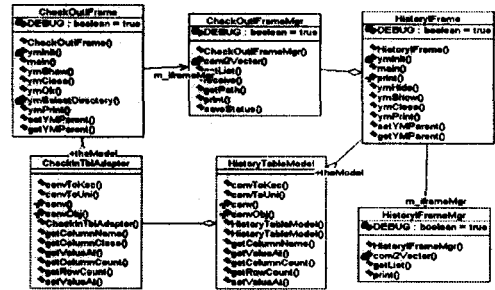
4.1 기술정보 관리의 개요

기술정보 관리는 설계 과정 중에 발생하는 기술문서정보, 형상정보 및 멀티미디어 정보 등의 모든 자료를 종합적이고 체계적으로 데이터 리파지토리 내에서 관리하는 것이다. 설계 엔지니어들은 설계 과정에서 기존의 정보나 문서, 도면을 보다 쉽게 찾을 수 있어야 하며, 설계변경으로 인한 변경정보를 반영시키기 위하여 여러 관련팀들의 작업과 연관될 수 있도록 해야 한다. 이에 기술정보 관리는 엔지니어링 CALS 환경에서 정보의 공유를 통해 업무의 효율을 높이고, 제품의 수명주기를 단축시키는데 있어 매우 중요한 역할을 한다.

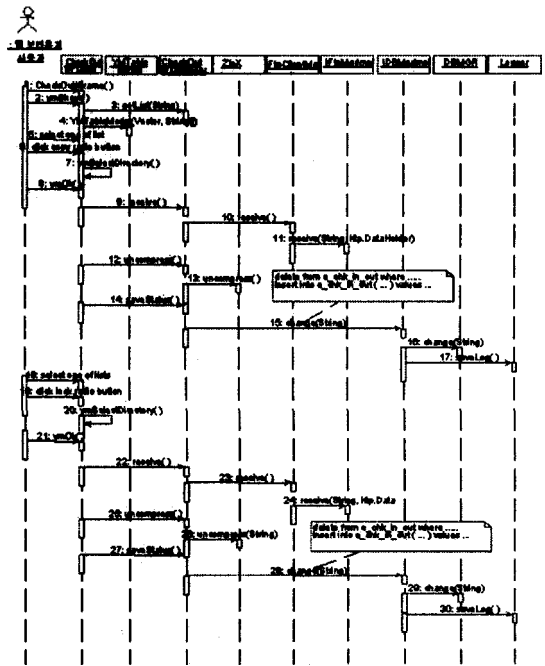
기술정보관리 시스템의 개발을 위한 설계 및 분석은 OMG (Object Management Group)의 표준인 UML (Unified Modeling Language)을 이용하였으며 다음과 같은 여러 개의 하위 모듈로 구성하였다.

- 기술정보의 프로파일을 관리하는 프로파일 관리 모듈
- 다양한 방법으로 기술 정보를 검색할 수 있는 검색 모듈
- 버전관리 및 체크인 체크아웃 모듈
- 기술정보유통 모듈

기술정보 관리 4개의 메인 모듈은 다시 세부적으로 18개의 서브 모듈로 구성하였다. [그림 4-1]은 그 모듈 중의 하나인 체크아웃 모듈에 대한 클래스 다이어그램의 예를 보여주며 [그림 4-2]는 기술된 객체를 순차 다이어그램에서 표현한 예를 보여준다.



[그림 4-1] 체크아웃 클래스 다이어그램 예



[그림 4-2] 체크아웃 순차 다이어그램 예

4.3 기술정보 관리 시스템 구현

서론에서 언급한 바와 같이 기술정보를 검색하고 이를 공유, 교환하는 것은 웹을 통해 이루어진다. 이를 위해 웹과 데이터베이스 관리 시스템의 연동은 매우 중요하다. 다수의 클라이언트에게 광범위하고 지역적으로 분산된 복수의 데이터베이스에 대해 일관성 있고 투명성 있는 액세스를 제공할 수 있도록 하기 위하여 웹과 데이터베이스와의 연동 모듈과 데이터베이스 관리기(Database Manager)를 개발하고, 분산된 클라이언트/서버 환경에 적합한 n-tier 아키텍처를 갖는 환경을 바탕으로 하여 앞의 시스템과 마찬가지로 CORBA 미들웨어에서 JAVA를 이용하여 기술정보 관리시스템을 구현하였다.

4.3.1 웹과 데이터베이스 관리 시스템 연동

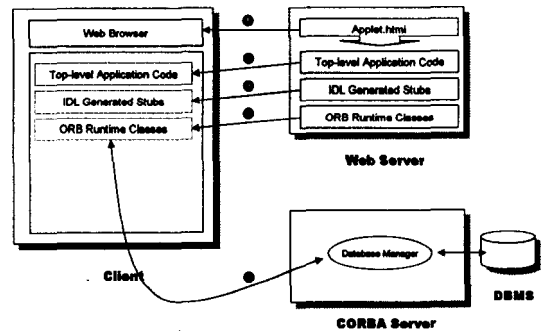
웹 클라이언트에서 데이터베이스 서버에 연동 하는 방법은 CGI (Common Gateway Interface)를 통해 처리하는 방법과 벤더 전용 프로토콜 기반 JDBC(Java Database Connectivity)를 이용하는 등 여러 가지 방법이 있으나, 본 연구에서는 객체지향 미들웨어 CORBA를 이용하여 웹 클라이언트와 객체 (Object) 서버, CORBA 서버, 데이터베이스 서버에 연결하였다.

CGI를 통해 처리하는 방법은 웹 서버에 대한 부하의 집중으로 인해 클라이언트 수가 많을수록 성능이 급격하게 저하하는 현상이 발생한다. 이러한 부하를 줄이기 위해 JDBC Thin 드라이버를 이용하여 웹 브라우저에서 데이터베이스 서버에 직접적으로 연결을 하면 웹서버에 대한 부하를 효과적으로 분산시킬 수 있으나 클라이언트마다

JDBC 드라이버를 설치해야 하는 번거로움이 있고 이 방법 또한 성능 평가 결과 뛰어난 결과를 얻을 수 없다.

본 논문에서 제안한 웹 클라이언트에서 데이터베이스 서버에 연동 하는 기법은 CORBA IDL를 사용하여 클라이언트와 구현 객체간의 인터페이스인 데이터베이스 관리기를 정의하여, 개발자와 사용자 모두에게 투명한 방법으로 분산 이 기종 플랫폼 환경 하에서 웹 클라이언트와 객체 서버, 데이터베이스 서버간의 상호 연동을 동적으로 처리할 수 있는 인터페이스 기법이다.

[그림 4-3]은 웹서버와 CORBA 서버, 데이터베이스 서버(DBMS Server)와의 동작과 정이며, 순서는 아래와 같다.

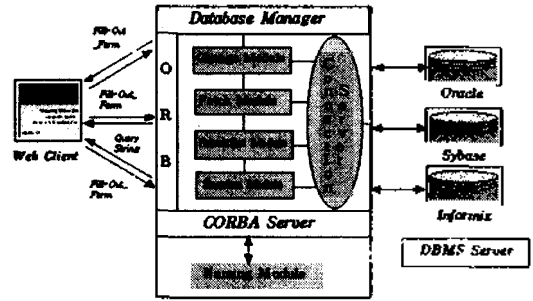


[그림 4-3] Web 과 DBMS 연동 모델

- ① 사용자가 브라우저를 통하여 해당 URL 을 호출한다.
- ② URL 호출에 의하여 최상위 응용 프로그램(Top-level Application Code)을 다운로드 받는다.
- ③ 최상위 응용 프로그램에서 사용하는 IDL 에 의해 생성된 스텝 클래스(Stub Class)를 다운로드 받는다.

- ④ 스텝 클래스에서 사용하는 ORB 실시간 클래스(ORB Runtime Class)를 다운로드 받는다.
- ⑤ ORB 실시간 클래스는 인터넷에서 자신의 서버를 찾아 바인딩 한 후, 데이터베이스 관리자에게 서비스 요청을 시행한다. 요청이 완료된 경우 CORBA 서버는 웹 브라우저에 그 결과를 되돌려 주며, 웹 브라우저는 이를 화면상에 디스플레이 한다.

터베이스 관리기의 동작구조는 [그림 4-4]와 같다.



4.3.2 데이터베이스 관리기

CORBA를 기반으로 한 웹/데이터베이스 시스템 간의 연동 모델에서 데이터베이스 관리기는 데이터베이스 내의 테이블을 검색하고 사용자 편의를 위한 유연한 질의 기능을 제공한다. 또한 질의 결과와 함께 SQL 코드를 웹 브라우저에 보여줌으로써 질의 결과를 쉽게 확인할 수 있다. 데이터베이스 관리기는 SQL 문법을 모르더라도 웹상에서 데이터베이스 시스템에 접근하여 원하는 조건에 따라 질의할 수 있다.

데이터베이스 관리기의 구조는 3-Tier 방식으로서 웹 클라이언트-CORBA 서버(데이터베이스 관리기), CORBA 서버-DBMS 서버로 이루어진다. 클라이언트와 데이터베이스 관리기 사이의 통신은 ORB간의 통신 매커니즘인 IOP(Internet Inter-ORB Protocol)를 사용한다.

데이터베이스 관리기는 Change Module (Insert, Update, Delete), Fetch Module(Query), ResultSet Module, Session Module, Naming Module의 5개의 모듈로 구성되어 있고, 이 5개의 모듈은 웹 클라이언트에서 요구한 내용을 해당모듈로 접근하여 실행되며, 데이

[그림 4-4] 데이터베이스 관리기의 동작구조

- Change Module은 사용자가 정보의 삽입, 수정, 삭제 등의 요청에 의해 웹 브라우저에서 동적 SQL문을 생성한다.
- Fetch Module은 사전에 있는 질의 대상 테이블의 Description과 칼럼명, 데이터 타입, 데이터 길이, 칼럼 등의 내용을 검색한다.
- ResultSet Module 질의한 결과를 CORBA IDL 데이터 타입으로 변환하고 다시 Java 데이터 타입으로 변환하는 역할을 한다.
- Session Module은 데이터베이스 서버와 연결된 Connection Server와 1:1 연결관계를 유지하고 사용자로부터 질의 문장을 받으면 Connection Server에게 처리를 요구하는 기능을 수행한다.
- Naming Module은 데이터베이스 관리기가 액세스할 수 있는 데이터베이스의 바인딩 정보(위치정보, JDBC 드라이버 정보)를 저장한다.
- Connection Server는 Session module의 요

청에 따라 연결 에이전트를 생성, 유지, 제거, 관리하는 것을 담당한다.

사용자의 질의 요청은 웹 브라우저상의 CORBA ORB를 통하여 데이터베이스 관리기에 접근하고, 데이터베이스 관리기의 해당 모듈은 데이터베이스 서버에 전달되어 관련된 테이블을 검색하고, 결과를 역 방향으로 웹 브라우저에 반환한다. 웹 브라우저에서는 자바 애플릿이 다운로드 되어, SQL 소스 코드와 질의 결과를 보여준다.

기술정보 관리는 앞의 분산협동설계 지원 시스템에서의 설계관과 가상모형 시스템의 형상 모델 가시화 기능을 연동하여 엔지니어링 정보를 통합 관리하고 이를 가시화할 수 있도록 개발되었다.

기술문서의 경우 CALS 표준인 SGML/XML을 기본적으로 수용할 수 있도록 하였다. SGML/XML 문서의 구조정보는 DTD (Document Type Definition)를 파싱함으로써 얻을 수 있는 요소(element)와 속성(attribute) 등의 정보에 의해 결정된다. 본 시스템은 DTD 트리 구조를 생성한 다음, 그 트리 구조에 맞게 기술문서를 Depth First Search를 이용하여 데이터베이스에 저장하였다. 저장된 SGML/XML 문서의 검색은 DTD 트리 구조에 따른 TreeView 형태로 보여주고, 구조기반 검색과 내용기반 검색이 가능하도록 구현 하였다.

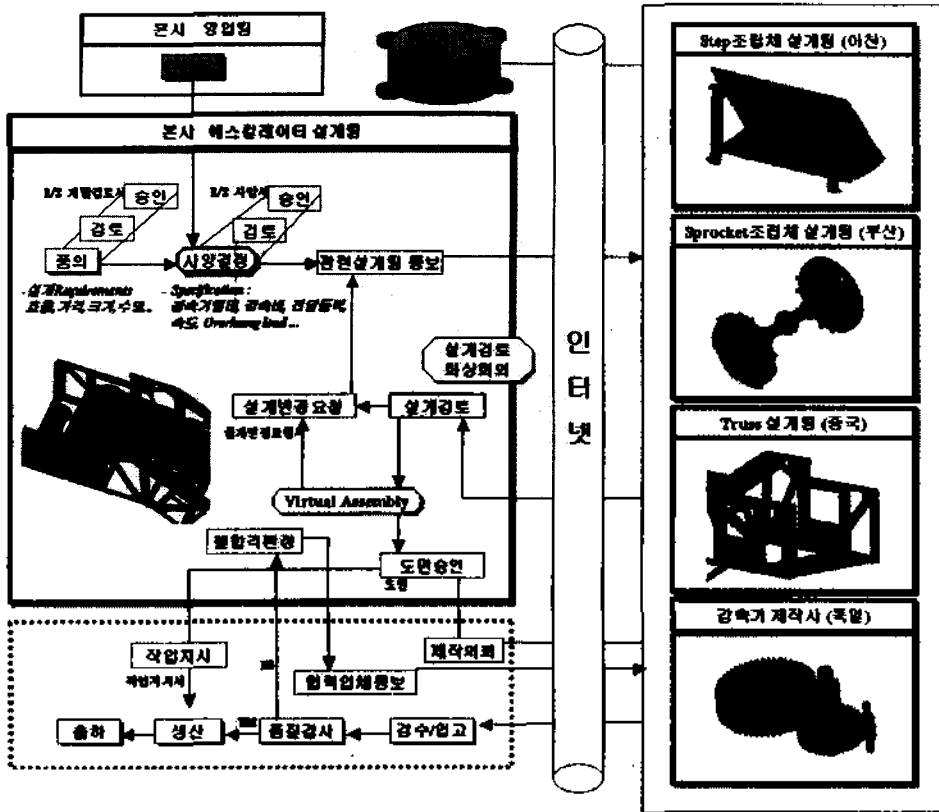
본 시스템은 이들 기능 이외에도 정보의 배포 기능과 같은 팀 작업에 적합한 기본적인 그룹웨어 제품의 기능도 포함하였다.

5. 적용예

[그림 5-1]은 국내 모기업의 에스컬레이터 구동장치 설계의 예로써 이러한 엔지니어링 CALS의 개념을 구체화하기 위한 시나리오를 보여준다.

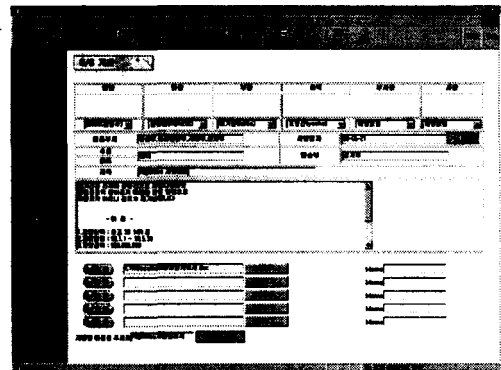
영업부에서 수주가 들어오거나 설계부 자체의 품의에 의해 에스컬레이터의 개발이 결정되면, 본사 설계부는 요구조건을 고려해 설계사양을 결정한다. 기존의 설계사양서는 종이 문서형태로써 결재과정을 거쳐 이루어지나 엔지니어링 CALS 환경에서는 전자문서로 작성되며 그룹웨어 기능을 통해 결재가 이루어질 수 있는 기술정보 관리 시스템을 이용한다. [그림 5-2]는 설계품의를 위한 시스템의 화면 예이며, 설계사양서 등의 사무업무 및 결재과정이 같은 방법으로 이루어진다.

에스컬레이터 사양으로부터 에스컬레이터의 구조가 결정되면, 본사에는 최상위에서의 설계 프로세스를 결정하고 ([그림 5-3]), 각각의 하위 시스템들의 설계를 추진한다. 에스컬레이터 구동부 설계의 경우, 구성품인 감속기는 독일의 감속기 제작사에 의뢰하여 이루어지고, Sprocket, Chain 등은 부산에서 설계 및 개발이 이루어진다. 또한 구동부가 안치될 Truss부는 중국 상해에서 설계된다. 따라서 본사에서는 설계사양을 지역적으로 분산된 설계팀 또는 제작사에 통보하게 되고 각 설계팀 또는 제작사는 자신의 설계를 행하게 된다. 이들은 각각 서로 다른 전문지식을 갖고 서로 다른 컴퓨터 지원 시스템을 이용해 자신의 설계를 수행하지만 상호 유기적인 연관성을 갖고 수행된다. 앞서 언급한 바와 같이 각 설계팀 또는 제작사는 본 시스템을 이용하여 공통의

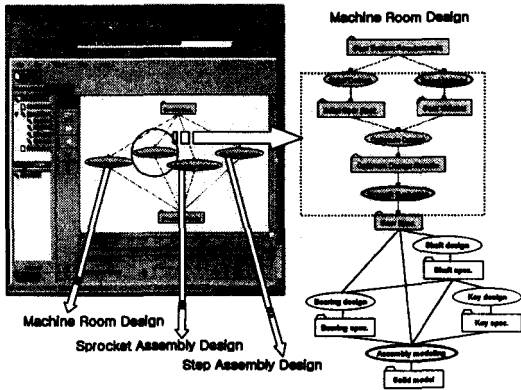


[그림 5-1] 엔지니어링 CALS 시나리오

데이터베이스를 통해 설계정보를 공유할 뿐만 아니라 서로의 설계 프로세스를 공유하고 설계의도를 전달하면서 설계를 진행해 간다. [그림 5-3]은 이러한 개념에 의한 프로세스 표현 예를 보여준다. 즉 에스컬레이터 구동부에 대한 전체 프로세스가 최상위 단계에서 [그림 5-3]의 좌측과 같이 정의된다. 각각의 작업 노드는 최종적인 수행작업이 만들어져 설계 지원 에이전트가 연결될 수 있을 때까지 하위의 세부 프로세스로 정의되고 전체 프로세스로서 등록된다.

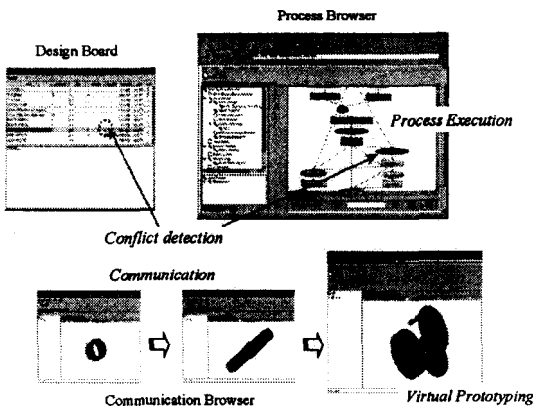


[그림 5-2] 기술정보관리 시스템을 통한 설계품의



[그림 5-3] 협동설계 프로세스의 표현

정의된 프로세스는 공동의 작업팀들에게 공유되면서 수행될 수 있다. [그림 5-4]는 감속기 설계의 실행과정에 대한 예를 보여준다.

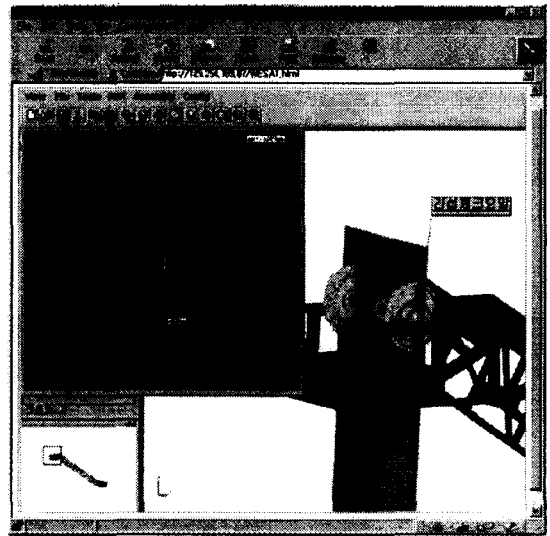


[그림 5-4] 프로세스 수행 예

그림의 좌측 상단에서는 설계판에서의 상충성 발생을 감지하는 화면을 볼 수 있으며, 우측 상단에서 이로 인한 작업의 실패 (Fail)가 프로세스 브라우저 상에 나타남을

볼 수 있다. 이 문제를 해결하기 위한 커뮤니케이션 세션이 구성되어 실시간 협동작업을 수행하는 화면을 좌측 하단에서 볼 수 있으며, 문제 해결 후 최종 형상 모델링된 결과를 웹 상에서 검토하는 화면을 우측 하단에서 볼 수 있다. 형상모델은 STEP AP203 데이터를 웹 상에서 가시화한 결과이다.

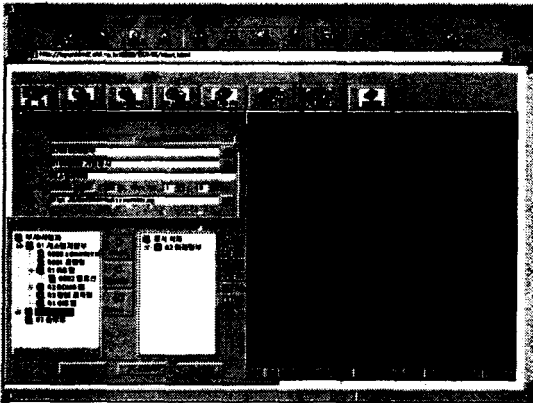
[그림 5-5]는 가상모형에서 수행할 수 있는 한 예로 가상조립을 수행하고 간섭을 확인을 위한 과정을 보여준다. 우선, 3차원 상에서 간섭을 확인한 후, 간섭에 대한 보다 정확한 정보를 얻기 위해 Cross Sectioning과정을 거친다.



[그림 5-5] 가상조립 및 간섭체크 (Cross Sectioning)

최종적으로 설계가 완료되면 그 결과물은 기술정보관리 시스템을 통해 데이터베이스에 등록된다([그림 5-6]). 왼쪽 상단은 엔지니어링 정보의 등록 화면이고, 하단은 해

당 정보에 대한 액세스 권한에 관련된 화면이다. 오른쪽은 해당 정보에 대한 가상모형이며 이 정보가 문서일 경우 SGML 브라우저가 기동된다.



[그림 5-6] 기술정보관리 등록화면

6. 결 론

본 논문에서는 엔지니어링 CALS 환경에서 공학설계를 지원하는 시스템을 개발하고자 하였다. 이 시스템은 분산된 환경에서 가상모형을 이용해 설계를 수행할 수 있도록 지원하고, 분산협동설계를 지원하며, 이로부터 발생하는 엔지니어링 정보를 통합 운용하기 위한 기능을 제공한다.

웹 기반 가상모형 시스템은 기존의 도면 작업을 3차원 형상에 의한 제품모델링 작업으로 대체함으로써 엔지니어링 CALS에서 목표로 하는 정보의 디지털화 및 Paperless와 한번 만든 정보의 일관된 활용이라는 목표에 부응한다.

분산환경에서 협동설계를 지원하기 위한 시스템은 프로세스 중심 협동설계의 개념을 도입하고, 이를 기반으로 여러 설계자 또는 설계팀이 설계 프로세스 및 설계 데이터를 공유하면서 협동설계를 수행함으로써 엔지니어링 CALS에서 목표로 하는 정보 공유와 협동작업의 목표에 부응한다.

또한 기술정보 관리를 통하여 기술정보를 체계적으로 저장, 검색, 수정할 수 있는 환경을 제공하며 이를 통해 정보의 최초 생성에서부터 폐기까지 일관성 있는 기술정보에 대한 라이프사이클 관리와 효율적 공유를 가능케함으로써 엔지니어링 CALS의 목표에 부응한다.

또한 CALS 표준인 STEP과 SGML을 활용하여 CALS가 지향하는 표준화의 목표에 부응한다.

궁극적으로는 본 시스템을 통해 엔지니어링 CALS의 최종 목표인 제품의 원가 절감, 제품 개발기간 단축, 품질향상을 통한 종합 생산성 향상에 기여할 수 있을 것으로 기대한다.

참고문헌

- [1] Han, J. and Requicha, A.A.G., "Modeler-independent feature recognition in a distributed environment", *Computer-Aided Design*, Vol. 30, No. 6, pp.453-463, 1998.
- [2] Shah, J. J., Dedhia, H., Pherwani, V., and Solkhan, S., "Dynamic interfacing of applications to geometric modeling services via modeler neutral protocol", *Computer-Aided Design*, Vol. 29, No. 12, pp.811-824, 1997.
- [3] 문왕식, 김덕수, 이동규, 장태범, 한성배, 김 현, "웹과 STEP 기반 3 차원 가상회의", CAD/CAM 학회 학술발표회 논문집, pp.11-16, 1998.
- [4] 최영, 양상욱, "STEPShare: 웹 환경에서의 CORBA 기반 3D STEP 브라우저", CAD/CAM 학회 학술발표회 논문집, pp.24-29, 1998.
- [5] Wang, F.-C. and Wright, P. K., "Web-based CAD tools for a networked manufacturing service", *Proc. ASME Design Engineering Technical Conference*, 1998.
- [6] Martino, T.D., Falcidieno, B., and Habinger, S., "Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment", *Computer-Aided Design*, Vol 30, No. 6, pp. 437-452, 1998.
- [7] M. Hardwick and D. Spooner, 1995, "An information infrastructure for a virtual manufacturing enterprise," *Proceedings of Concurrent Engineering: A Global Perspective*, McLean, VA, pp.417-429.
- [8] Jasnoch, U., Alok, S.K., and Haas, S., 1995, "A Collaborative Environment Based on Distributed Object-Oriented Databases," *Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE Computer Society.
- [9] Kim, Hyun and Chung, M. J., "A Web-based Framework for Engineering Design Process in a Concurrent Engineering Environment", *Proceedings of Advanced Concurrent Engineering, Tokyo, Japan*, pp.417-423, 1998.

저자 소개

김 현 (e-mail : hyunkim@etri.re.kr)

- 1984년 한양대학교 기계설계학과 학사
- 1987년 한양대학교 기계설계학과 석사
- 1997년 한양대학교 기계설계학과 박사
- 1990년 ~ 현재 한국전자통신연구원 동시공학연구팀 선임연구원

이재열 (e-mail : jaelee@etri.re.kr)

- 1992년 포항공과대학교 산업공학과 학사
- 1994년 포항공과대학교 산업공학과 석사
- 1998년 포항공과대학교 산업공학과 박사
- 1998년 ~ 현재 전자통신연구원 동시공학연구팀 선임연구원

김형선 (e-mail : kimhs@etri.re.kr)

- 1981년 상지대학교 경영학과 학사
- 1993년 광운대학교 전자계산학과 석사
- 1985년 - 현재 한국전자통신연구원 동시공학연구팀

한성배 (e-mail : sbhan@etri.re.kr)

- 1977년 고려대학교 산업공학과
- 1982년 고려대학교 대학원 산업공학과 공학석사
- 1996년 고려대학교 대학원 산업공학과 공학박사
- 1979년 - 1991년 한국기계연구원 근무
- 1992년 - 현재 한국전자통신연구원 전자상거래연구부 동시공학연구팀장

박상봉 (e-mail : sbpark@etri.re.kr)

- 1974년 고려대학교 산업공학과 학사
- 1981년 고려대학교 대학원 경영학 석사
- 1975 ~ 1981 한국과학기술연구소 전자계산부 선임연구원
- 1982 ~ 1988 한국과학기술원 시스템공학센터 6그룹리더
- 1989 ~ 1992 한국과학기술연구원 생산자동화연구단장
- 1993 ~ 1996 한국과학기술원 통합생산시스템연구부장
- 1996 ~ 1998 한국전자통신연구소 시스템통합연구부장
- 1998 ~ 현재 한국전자통신연구원 전자상거래연구부장