

MDO 기반 협력설계 시스템

최영#, 박진표*

MDO-Based Design Collaboration

Young Choi[#] and Jin Pyo Park^{*}

ABSTRACT

MDO is one of the efficient methods for huge and multi-functional system design. This paper describes a design collaboration framework with MDO in networked design environment. A prototype of web-based integrated design system was implemented to show sharing and exchange of models and analysis information between MDO modules and collaborative design stations. Server System consists of MDO modules for optimization and modeling module for 3D modeling operation. Client system provide user with graphic interface for shape modeling and system operation. We believe that the proposed approach can be extended to solve real complex multidisciplinary design problems.

Key Words : MDO(다분야 최적설계), Distributed collaborative design(분산협력설계), Parasolid(파라솔리드), Java(자바), CORBA(코바)

1. 서론

현재 많은 산업 분야에서 제품의 설계 및 개발이 단일 부서나 기업 안에서 끝나는 것이 아니라 다른 부서나 기업간의 협력 또는 분업을 통하여 이루어지고 있다. 또한 항공기나 선박과 같은 대규모 시스템의 설계를 위해서는 이러한 협력/분업은 필수적이라 하겠다. 분업화된 환경하에서 각각의 부서는 각각의 특화된 작업을 동시에 진행함으로써 개발 비용과 시간을 절약할 수 있다. 그러나 이러한 분업 체계에서는 부서나 기업간의 CAD 또는 PDM 시스템의 이질성이나 제품 데이터 표현 방법의 차이로 인해 이를 통합하기 위한

부가적인 비용과 개발 기간이 발생하기도 한다. 따라서 생산 과정에서의 협업을 성공적으로 수행하기 위해서는 제품 개발 초기 단계에서부터 분산된 CAD 와 관련 시스템들을 유기적으로 통합 관리할 수 있어야 하며 각 사이트간 의사 교환이 원활하게 이루어져야 한다. 이러한 환경이 조성되어야만 분산된 다수의 제품 개발자들이 이질적이고 지역적으로 분산된 컴퓨팅 환경에서 상이한 업무를 독립적으로 수행하면서 협력을 통한 제품의 공동 개발이 가능해 진다. 이를 토대로 Regli¹, Potter² 와 같이 생산과정에서 이를 활용하기 위한 방안이나 Martino³ 와 같이 분산 환경에서 설계와 해석을 통합하려는 연구 등이 활발히 진행되고 있다.

· 접수일: 2002 년 12 월 4 일; 게재승인일: 2003 년 7 월 11 일
교신저자: 중앙대학교 기계공학부
E-mail: yychoi@cau.ac.kr, Tel: (02) 820-5312
* 중앙대학교 대학원 기계공학과

본 연구에서는 개별적인 부서나 기업의 자율성을 최대한 살리고, 동시에 전체 설계 성능의 향상에 기여하기 위해 다분야 통합 최적설계(MDO)를 웹기반 분산환경 시스템으로 확대 적용하여 지역적 한계를 극복하고 부서간 또는 기업간 협력을 증대하기 위한 방법론을 제시하고자 한다. 또한 설계 및 생산과정에서의 효율적인 분산 엔지니어링 환경 구축을 위해 MDO 와 설계/모델링, 해석을 위한 정보를 상호 공유, 교환 할 수 있는 웹기반 통합 설계시스템을 구현하였다.

2. 분산 MDO 시스템 모듈

2.1 분산 MDO 시스템 개요

MDO 는 시스템이 대규모이고 다기능이 필요한 설계제품을 효율적으로 설계하기 위한 방안의 하나로 미국의 항공분야 중심으로 개발된 설계기법이며, 최근 그 유용성이 입증되고 있다 (Balling⁴). 그러므로 개별적인 부서나 기업의 자율성을 최대한 살리고, 동시에 전체 설계 성능의 향상에 기여하기 위해 다분야 통합 최적설계(MDO)를 웹기반 분산환경 시스템으로의 확대 적용이 필요하다.

MDO 시스템을 웹기반 분산환경으로 적용하므로써 기대되는 효과는 먼저 지리적으로 분산된 여러 전문 시스템의 웹을 통한 통합을 들 수 있다. 일반적으로 설계/개발 프로세스가 분업화 되어있는 현실에서 각 분야별 전문 시스템은 지역적으로 떨어져 있을 수 밖에 없기 때문에 이러한 환경하에서 전체 시스템을 통합/조율하여 관리하기 위한 방법이 필요하게 되고 이를 해결하기 위해서는 웹을 응용하는 것이 필수적이라 하겠다. 각 분야별 전문가 또한 각기 전문 프로세스 관리에 시간을 집중 투자함으로써 제품 개발에 대한 능률 향상을 기대할 수 있다.

둘째로 제품개발비용의 절약을 기대할 수 있다. 실제 프로세스에 사용되는 시스템은 분산되어 있기 때문에 각각의 개별 시스템은 외부 요청에 대해 동시 다발적으로 고유의 역할을 최대한으로 발휘하게 되고 결국, 각 시스템의 유휴 시간을 줄이는 효과를 가져와 제품 개발비용을 단축하는 효과를 얻게 된다.

다음의 분산 MDO 시스템은 실제 MDO 기법을 어떻게 웹기반의 통합 시스템으로 구현할 수 있는가에 대한 실례를 보여준다.

2.2 분산 MDO 시스템 구조

본 연구에서는 Yang⁵ 등에 의해 연구된 MDO 기법을 분산 환경에 적용하였다. 분산 MDO 모듈은 모두 5 개의 독립 서버로 구성하였다. 각각의 서버는 자체적인 운용을 위한 인터페이스와 분산된 환경에서 클라이언트의 요구에 반응하기 위한 네트워크 인터페이스를 가지고 있다. Fig. 1 하단의 4 개의 하위 서버(DOE, Kriging, Optimization, Discipline Server)는 관리시스템 서버(Management System Server)의 관리에 따라 자신이 맡은 분야의 역할을 수행하게 되고, 관리서버는 각 서버들의 결과를 종합, 조율하여 프로세스를 관리한다.

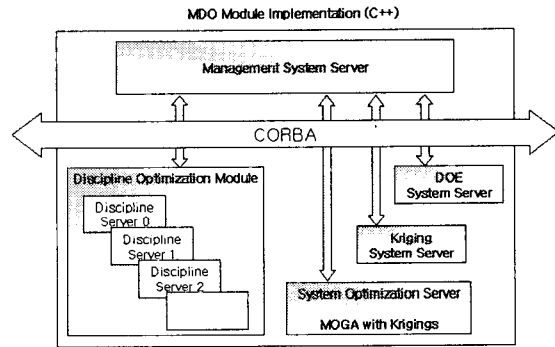


Fig. 1 MDO module framework for distributed network environment

분산 환경에서 MDO 를 이용해 최적화 작업을 수행하고자 할 경우 클라이언트는 먼저 관리시스템 서버에 접속한다. 클라이언트가 관리시스템에 필요한 작업을 요청하면 클라이언트의 요구에 따라 최적화 작업을 수행하기 위해 필요한 다른 서버와의 접속을 시도한다. 활성화된 서버는 관리시스템의 요청을 수행하기 위해 대기상태가 되고, 요청이 접수되면 내부적인 프로세스에 의해 각자의 역할을 수행하고 수행 결과를 다시 관리시스템에 되돌려 주게 된다. 이러한 과정을 반복함으로써 생성된 최종 결과를 클라이언트로 전송한다.

관리시스템은 전체적 흐름을 제어하는 시스템이다. 먼저 DOE 시스템에 변수 수와 샘플 수를 전달하여 샘플데이터 최적화 수행을 요청한다. DOE 시스템으로부터 넘겨받은 결과를 이용해 샘플데이터 파일을 만들고 이를 Kriging 시스템에 전달, 근사모델인 Kriging 정보를 넘겨 받는다. 시스템 최적화 서버는 Kriging 정보를 이용해 근사

적인 Pareto Set 을 생성하여 관리 시스템에 전달하고 관리 시스템은 Pareto Set 을 이용해 최적화에 대한 수렴 정도를 체크하고 만족스럽지 못할 경우 위 과정을 반복적으로 수행한다. 최종적으로 수행된 결과는 클라이언트에 전송되고 클라이언트는 전송된 Pareto Set 중 원하는 데이터를 선택해 관리시스템에 최종 설계변수값을 요구한다. 이때 관리시스템은 Discipline 모듈을 이용해 최종 설계변수를 클라이언트로 전달함으로써 모든 최적화 작업을 마치게 된다.

하위 서버 중에서 Discipline 모듈의 각 서버들은 관리시스템의 요청에 의해 동시에 최적화 작업을 수행하게 되어 프로세스 시간을 절약할 수 있다. 다른 나머지 3 개의 서버는 관리시스템의 프로세스 순서에 의해 작동된다. 이 과정에 쓰레드(Thread)를 적용, 클라이언트의 요구를 동시에 수용하여 모든 서버가 대기시간 없이 운용되도록 하여 분산 작업의 효율을 증대시켰다.

MDO 모듈의 모든 서버는 C++를 이용하여 구현되었으며 타 서버와의 인터페이스를 위하여 CORBA 의 ORB 클래스를 포함하고 있다.

3. 원격공동설계 시스템

3.1 연구 개요

공동 협업 시스템을 위한 연구는 다양한 분야에서 시도되었다. Debras⁶ 는 표준화된 제품 데이터를 확장된 기업 환경에서 공유하려는 시도를 했으며, Charles⁷ 가 구축이 된 스탠포드 대학의 SHARE 프로젝트는 웹 환경에서의 실시간 공동 협업 환경을 목적으로 수행되었다. 이외에도 Dong⁸, Cutkosky⁹, Choi^{10,11}, Kim¹² 등에 의해 유사한 연구가 진행되고 있다.

본 절에서는 이러한 연구를 바탕으로 제품의 상세형상 모델링 단계에서 여러 사용자가 실시간으로 동시에 참여할 수 있는 공동모델링시스템의 구현방법에 대해서 논의한다.

분산된 작업자간의 협력 모델링은 문제점 발견 단계와 이를 해결하는 단계로 나눌 수 있다. (Choi¹¹) 문제점의 발견 단계에서 여러 작업자간의 의견을 개진하고 이를 조율하는 일이 필요한데 이를 위해서 뷰 데이터와 마크업 데이터를 공유할 수 있어야 하고 측정기능 등과 함께 의견 교환을 하기위한 채팅이나 화상회의 기능이 필요하다. 또

한 서버는 데이터베이스나 기타 기능성 모듈 또는 객체들과 연결되어 사용자들의 요구에 응답하고, 때로는 능동적으로 사용자에게 데이터와 메시지를 전달하는 기능 또한 필요하게 된다. 발견된 문제를 해결하기 위해 블렌딩, 테이핑 등 솔리드 모델러가 제공해야 하는 기능이 필요하며, 모델 데이터에 대한 BOM 정보 등도 함께 다룰 수 있어야 한다.

구현된 시스템은 사용자에게 기본적인 솔리드 모델링 기능과 사용자간 의견교환을 할 수 있는 뷰 공유, 챗(chat) 등의 컨퍼런스 기능을 제공한다. 시스템을 이용한 모델링 작업은 실시간으로 진행되며, 한 사용자가 변경한 내용은 즉시 다른 사용자에게 알려진다. 이는 사용자들이 완성된 제품 모델이나 모델링 오퍼레이션을 공유하는 것 뿐 아니라 모델링 과정에 있는 데이터를 공유하고 있기 때문에 가능하다. 시스템은 클라이언트/서버의 구조로, 클라이언트는 웹 상에서 작동하며, 모델링과 컨퍼런스를 위한 사용자 인터페이스를 제공하고 사용자가 서버의 웹 페이지에 접근하는 순간 다운로드되어 실행된다.

서버는 상용 솔리드 모델링 커널인 파라솔리드를 이용하여 구현되었으며, CORBA 를 통해 클라이언트의 요구를 수용하고 결과를 넘겨주게 된다.

3.2 시스템의 구조

Fig. 2 는 서버와 클라이언트를 구성하는 요소들을 보여주고 있다. 전체적인 구조는 Orfali¹³ 의 Object Web 클라이언트-서버의 구조를 가지며 서버와 클라이언트는 CORBA 를 통해 상호 통신하고 있다.

서버는 클라이언트를 관리하고 솔리드 모델링 커널(Parasolid¹⁴)과 커널을 둘러싼 세션 객체, 모델링 과정에서 사용되는 여러 객체들로 이루어져 있으며 설계 데이터와 부가적인 정보를 표현하기 위해 데이터베이스를 사용한다. 또한 서버는 컨퍼런스를 위해서 세션에 참가한 클라이언트의 참조 객체를 리스트로 가지고, 마크업이나 채팅 텍스트 등 사용자가 발생시키는 데이터를 관리하여 클라이언트들에 분배하기 위한 관리자를 가진다.

클라이언트는 Java3D 와 swing 패키지를 이용한 사용자 인터페이스 부분과 서버의 질의에 응답하기 위한 콜백 객체, ORB 커넥션을 담당하는 코어 부분 등으로 이루어져 있다.

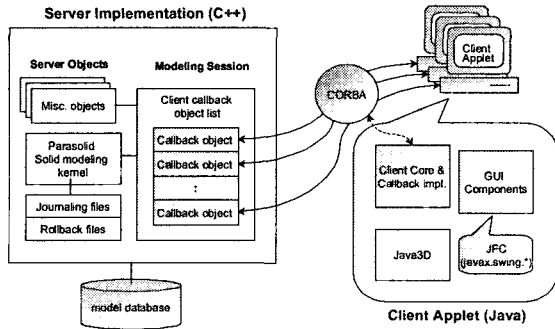


Fig. 2 Modeling system for distribute collaborative design

클라이언트 코어는 IDL로부터 생성된 스텝 코드를 확장하여 서버의 서비스를 사용할 수 있도록 하는 핵심 부이고, 클라이언트 콜백은 콜백 객체의 구현부이며, 서버가 호출하고 클라이언트가 반응할 수 있게 하는 부분이다. 예를 들어 서버가 한 사용자에게 뷰 변경 메시지를 받으면 다른 사용자들의 클라이언트측의 콜백 메소드를 호출하게 되고, 그 구현부에서는 렌더러에 뷰 정보를 전달함으로써 화면이 갱신되게 한다. 이로써 모든 사용자가 같은 뷰를 공유할 수 있게 된다.

뷰어는 와이어 프레임 모델이나 셰이딩 모델로 보여주는 기능을 한다. 또한 마우스 메시지에 반응하여 화면을 확대, 축소하거나 회전, 이동이 가능하며 이러한 뷰 변경 메시지를 컨퍼런스 서버에 전달하고, 전달 받는 기능을 가지는데, 뷰 변경 시 화면의 크기나 중형비가 다른 사용자들 간에도 일관된 뷰를 공유할 수 있어야 하므로 정규화된 뷰포트 데이터와 회전 변환 행렬을 전송하거나 받게 된다. 또한 마크업 데이터나 엔터티 브라우저에 의한 선택 부분에 해당하는 형상을 별도로 표시하는 기능 등을 가진다.

4. 연구결과

지금까지 분산 MDO 시스템 및 원격설계 시스템 등에 대하여 살펴 보았다. Fig.3은 이러한 분산 MDO 기법과 원격설계 시스템을 묶은 통합 네트워크 기반 설계 시스템의 전체적인 구조를 보여준다. 시스템은 위에서 언급한 분산 MDO 모듈, 원격협력설계 모듈 외에 이들을 운용하기 위한 클라이언트 모듈, 세가지로 구성되어 있다.

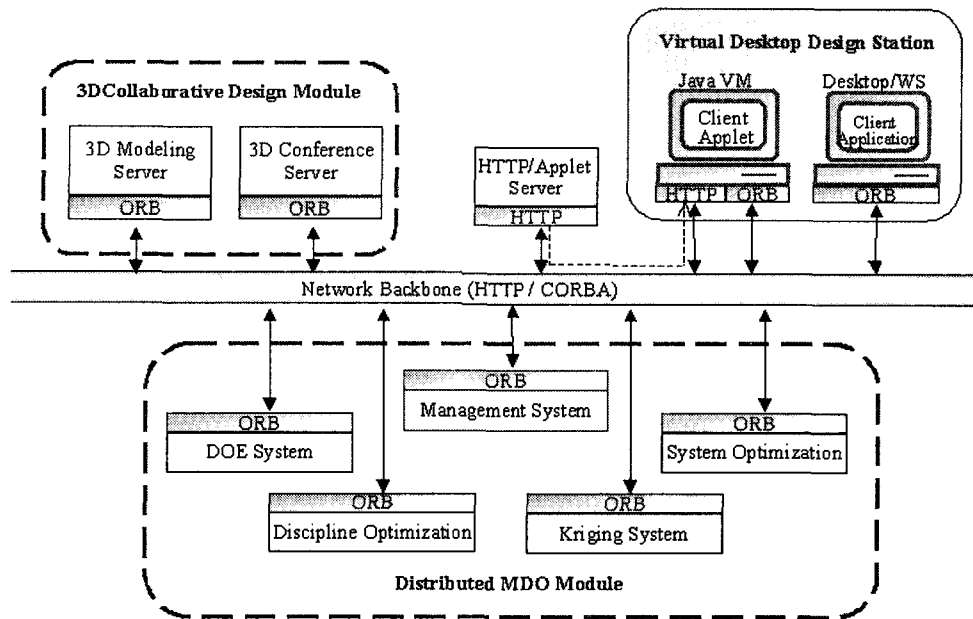


Fig. 3 Structure of networked design system with MDO modules

4.1 Client Applet

클라이언트 애플릿은 Fig. 4 와 같이 여러 서버들의 질의에 응답하기 위한 콜백 객체, ORB 커넥션을 담당하는 코어 부분과 Java3D 와 Swing 패키지를 이용한 사용자 인터페이스에 MDO 모듈의 서버들의 프로세싱 상황을 보여주기 위한 MDO 브라우저로 구성되어 있다.

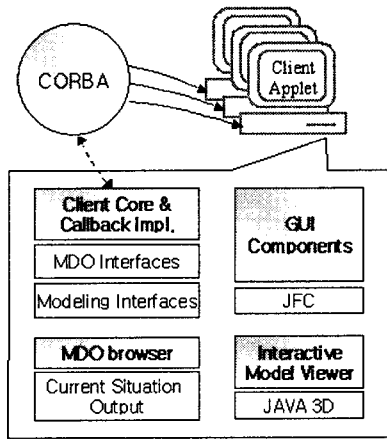


Fig. 4 Structure of client applet

콜백은 일반적인 CORBA 객체들과는 반대로 클라이언트에 콜백 객체의 구현부가 있고, 서버측에서 호출을 할 수 있게 되는 메커니즘을 가진다. 이를 이용하여 서버에서 능동적으로 클라이언트나 또는 다른 서버에 데이터나 메시지를 전달할 수 있다(Siegel¹⁵).

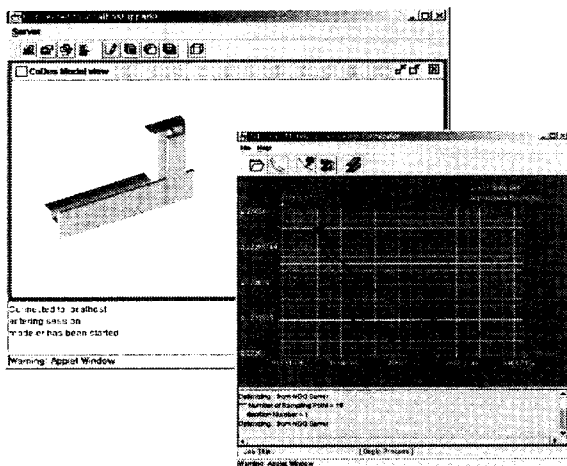


Fig. 5 Client applet

이러한 구조하의 시스템에서는 웹브라우저만 있으면 플랫폼에 관계없이 Fig. 5 와 같이 클라이언트 어플리케이션이 일관성 있게 실행되며 CORBA 기술의 적용으로 분산된 최적화 모듈간의 인터랙션이 실시간으로 가능하게 된다. 또한 원격지의 해석모듈이나 CAD 모듈과의 능률적인 정보 공유가 이루어짐으로써 MDO 기법을 적용한 효과적인 설계 시스템으로서의 발전을 기대할 수 있다.

4.2 시스템 운용 시나리오 및 실행 예

4.2.1 통합 설계 시스템 운용

Fig. 6 은 통합 설계 시스템의 운용 절차를 나타낸다. 사용자가 웹브라우저로 클라이언트 애플릿을 포함하는 서버에 접속하면 클라이언트가 다운로드 되어 실행되고, 협력 모델링 서버에 연결할 수 있다. 최초로 서버에 연결한 클라이언트의 요구에 의해 ORB 는 서버 프로그램을 실행시켜서 세션 객체 등의 구현 객체들을 활성화 하고, 세션에 참여하게 된다. 세션이 시작되면 MDO 모듈에 해당하는 서버들과 모델러가 초기화되며, 이 때부터 시스템이 제공하는 모델링 기능과 MDO 를 이용한 최적화 기능을 사용할 수 있게 된다.

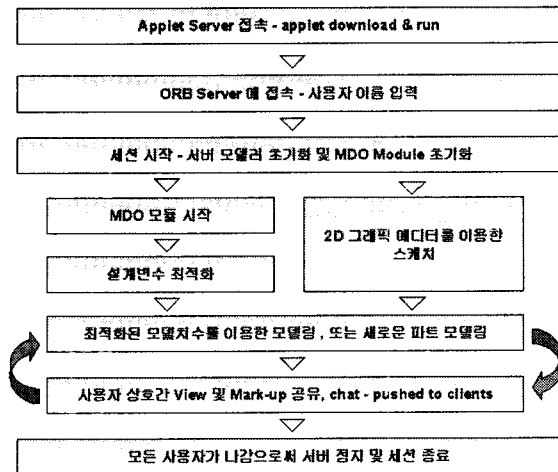


Fig. 6 System application scenario

세션을 열수 있는 권한이 있는 첫 사용자가 세션에 참여한 이후 다른 사용자가 세션에 참여하려고 하면, 세션의 개설자는 그 사용자를 참여시킬 것인지를 결정할 수 있고, 개설자의 허가 하에 다른 사용자가 참여할 수 있다.

세션이 시작되고 서버들이 초기화되면 사용자는 크게 두 가지 프로세스로 구성된 작업을 할 수 있다. 하나는 MDO 서버 모듈을 이용한 설계변수에 대한 최적화 작업이고, 다른 하나는 최적화된 설계변수를 협력 모델링 모듈로 전달하여 3D 모델링을 수행하고 이를 토대로 모델에 대한 수정/검증을 하는 단계이다.

최적화 작업을 진행하기 위해서는 사용자가 초기화된 관리시스템을 호출하여 필요한 작업내용을 전달한다. 이때 관리시스템은 다른 하위 서버들을 초기화하고 사용자의 요청을 받아 자신의 통제 하에 있는 하위 서버들을 필요에 따라 호출하여 최적화에 필요한 작업을 순차적 혹은 동시다발적으로 진행시킨다. 하위 서버들로부터 생성된 결과를 토대로 관리시스템은 분석/조율하는 역할을 수행하고 최종결과를 사용자에게 넘겨주게 된다.

MDO 모듈로부터 최종 결과를 넘겨받은 사용자는 협력 모델링 시스템을 이용해 모델링 작업을 수행하게 된다. 설계변수를 모델링 서버에 전달하면 서버는 모델링 커널을 이용, 모델링을 수행하고 그 결과를 클라이언트의 화면에 디스플레이하게 된다. 사용자는 애플릿 상의 모델링 도구를 이용해 데이터를 수정, 보완할 수 있다. 이때 실제 데이터는 서버의 데이터베이스나 모델링 커널에 존재하고 있으며 한 사용자가 변경한 파트는 모든 사용자의 클라이언트 뷰에 반영되고 이때의 뷰와 마크업 등은 세션에 참가한 여러 사용자에게 공유되어 있으므로 동시에 같은 모델을 관찰하면서 채팅이나 그 외의 통신수단을 통한 협력 모델링을 할 수 있다.

마지막으로 모든 사용자가 시스템을 종료하면 서버가 정지되고 세션이 종료 된다.

4.2.2 MDO 모듈을 통한 최적화 작업 수행 과정

이 절에서는 L 자 아이빔 2 개에 외력이 가해졌을 때 외력에 견딜 수 있는 아이빔의 최적 치수를 결정하는 최적화 작업이 시스템에서 실행되는 예를 통하여 구현된 MDO 시스템의 기능과 동작 방식에 대하여 설명한다.

최적화 작업을 위해 사용자가 MDO 서버 시스템에 접속하면 Fig. 7 과 같이 최적화 작업 상황을 보여주기에 위한 창이 나타난다.

관리시스템은 하위 시스템들을 초기화 하고

각 서버에 필요한 작업을 지시한다. 이 때부터 모든 시스템들은 각자의 맡은 역할을 수행하고 그 결과를 상위 시스템에 전달하는 과정을 반복함으로써 최적화 결과를 산출해낸다. 메인 클라이언트 주위의 창은 각 서버시스템의 현재 운용 상황을 보여준다.

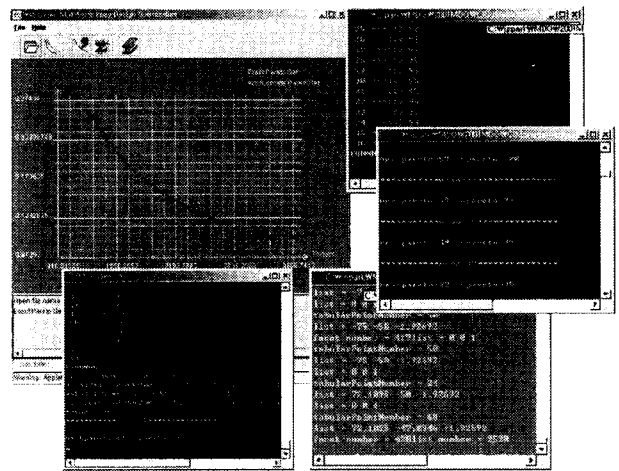


Fig. 7 Client and server systems of MDO

Fig. 8 은 시스템을 통하여 최적 Pareto Set 을 얻어내기 위해 반복되는 작업 과정을 보여준다. 시스템의 최적화 작업의 정확성을 판단하기 위해 예제에 대한 정확한 값과 계산된 값을 비교하였다. 회색의 그래프는 실제 정확한 Pareto Set 를 나타내고 검은색의 그래프는 MDO 모듈을 통해 계산된 Pareto Set 변화의 추이를 보여준다.

시스템을 통해 Pareto Set 를 얻어낸 후, 클라이언트가 MDO 창을 통해 원하는 데이터 포인트를

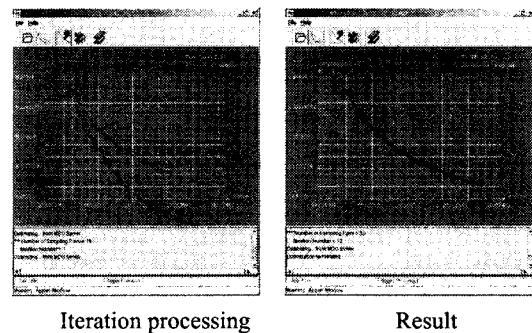


Fig. 8 Opmization process using MDO module

선택하면 관리서버는 Discipline Optimization 모듈을 통해 최종 아이빔의 치수를 결정한다.

이렇게 결정된 치수는 삼차원 모델링을 위한 초기 설계 치수로 사용되기 위해 모델링 서버로 전달되게 된다.

4.2.3 협력설계 시스템을 이용한 모델링 과정

MDO 모듈을 통해 생성된 데이터를 이용해 두 명의 사용자가 협업하여 모델링을 하는 작업을 예로 들어 설명한다. 단, 두 명의 사용자가 동시에 모델링 시스템에 접속해 있는 것을 전제로 한다. 또한 다음에서 설명되는 작업은 두 사용자 중 한 사람이 혼자 작업을 수행하였을 때에도 마찬가지로의 결과를 가지며 모델링의 전 과정은 모든 사용자에게 공유된다.

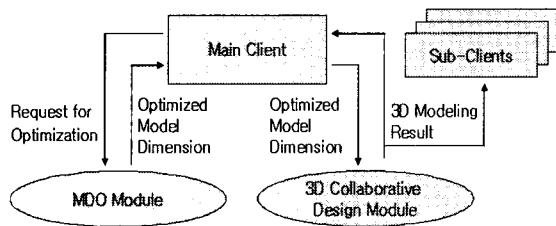


Fig. 9 Process flow between MDO module, design module and clients

Fig. 9 와 같이 한 사용자가 MDO 모듈을 이용해 계산된 결과를 모델링 서버에 전달하여 삼차원 모델링을 요청하면 서버는 모델링 작업을 수행한 후, 양쪽 사용자 모두에게 모델링 결과를 넘겨주게 된다. 실제의 데이터는 서버의 모델링 커널에 있으며 한 사용자가 변경한 파트는 클라이언트 폴백 메커니즘에 의해 즉시 모든 사용자의 클라이언트 뷰에 반영된다.

Fig. 10 의 오른쪽과 왼쪽 타이틀 바에 나타나 있는 정보와 같이 각 창은 다른 사용자가 로그인하여 작업하는 것이며, 그림에는 같이 나타나 있지만 각각의 창은 분산된 다른 시스템에서 실행되는 별도의 애플릿 인스턴스이다. 한명은 셰이딩된 모델로, 다른 한명은 와이어 프레임으로 보고 있지만 모델의 회전된 각도와 위치를 보면 현재 뷰가 공유되고 있음을 알 수 있다.

이렇게 생성된 모델 데이터와 시스템이 제공하는 채팅이나 다양한 커뮤니케이션 도구를 이용,

사용자간 협력을 통해 모델을 수정 보완할 수 있다.

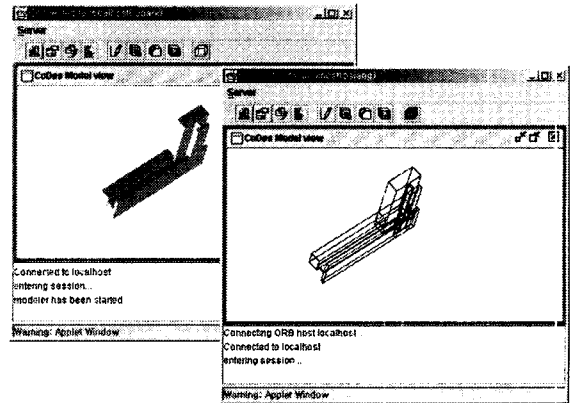


Fig. 10 Collaborative modeling with optimization result

Fig. 11 에서는 와이어 프레임 뷰 상태에서 오른쪽의 한 사용자가 에지들을 선택하여 필렛 작업을 하고 있다. 그 결과로 왼쪽의 사용자 또한 수정된 모델을 보게 되며, 이러한 작업을 통해 상호보완적이고 합리적인 설계작업을 수행하게 된다.

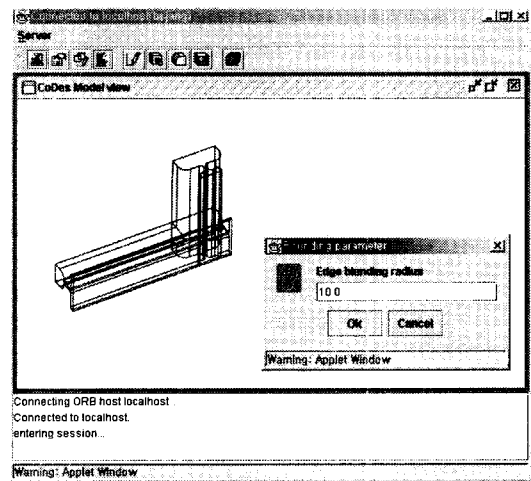


Fig. 11 Edge fillet operation

모델링 시스템은 MDO 모듈의 결과를 이용한 모델링 작업과는 별도로 2D 스케치 작업부터 시작하여 최종 삼차원 모델을 생성할 수 있다. 이 경우에도 마찬가지로 모든 작업은 같은 세션의 다중 사용자에게 공유된다.

사용자는 모델링하는 도중, 언제든지 모델을 저장할 수 있다. 모델의 저장은 기능에 의해 이루어지며 부가적인 정보는 별도로 데이터베이스화되어 서버측에 저장된다.

현재까지 구현된 모델링 기능은 위에서 예시한 몇 가지 기능뿐이지만, 그 외의 모델링 기능들은 구현된 시스템의 프레임워크의 변경 없이 쉽게 추가될 수 있으며 그럼으로써 실용적인 협력 솔리드 모델링 시스템으로 확장 될 수 있다.

5. 결론

본 논문에서는 급변하는 현재 및 미래의 생산 환경과 시장의 요구에 빠르게 부응하고 기업 경쟁력 향상을 꾀하기 위한 방법으로 인터넷/인트라넷에서 적용 가능한 여러 기술들과 다분야 통합 최적설계(MDO) 기법을 웹기반 분산환경 시스템으로 적용하는 방법론과 프로토타입 구현에 대해 논의하였다.

본 연구에서는 객체간의 풍부한 상호 작용을 위한 CORBA 와 웹 환경의 표준 프로그래밍 언어인 자바 등과 같은 기술들을 제품 설계 환경에 어떻게 실질적인 방법으로 적용할 것인가에 대한 논의와 그 활용 가능성을 모색하는 것을 목적으로 하였다.

이러한 목적을 달성하기 위해서 먼저 MDO 기법의 분산환경으로의 적용과 웹기반 MDO 시스템으로부터의 결과를 이용해 분산 환경하에서 여러 사용자가 실시간 협력 작업이 가능한 솔리드 모델링 시스템을 구현하였다.

구현된 시스템에서는 웹브라우저만 있으면 플랫폼에 관계없이 클라이언트 어플리케이션이 일관성있게 실행되게 하였다. 자바기술을 도입함으로써 사용자의 입장에서 최적화를 위한 각 서버 시스템들이나 고가의 솔리드 모델러를 구입, 설치하지 않아도 웹브라우저 만으로 구현 시스템을 사용할 수 있게 하였고, CORBA 기술의 적용으로 물리적으로 분산된 독립적 시스템들의 통합/제어, 여러 사용자 간의 정보 공유를 실시간으로 가능하게 하였다.

시스템은 설계최적화를 수행하기 위한 MDO 모듈과 독립적인 모델러의 기능을 제공하는 모델링 모듈의 서버 시스템과 이들의 운용을 위한 클라이언트로 구성되어 있으며 자체적으로 설계한

인터페이스에 의해 상호간의 인터랙션을 수행하지만, 해석, 최적화 설계 등의 기능을 지원하는 다른 표준 CORBA 객체와도 함께 작동할 수 있을 것이다.

본 연구를 통하여 구현된 시스템 구성기술은 실제 다분야 통합 설계의 대상이 되는 대규모이고 다기능이 필요한 대형 설계제품에 효율적으로 적용될 수 있으리라 기대된다.

후 기

본 연구는 한국과학재단 목적기초연구(R01-1999-00268)와 첨단조선공학연구센터의 지원으로 수행되었다.

참고문헌

1. Regli, W. C., "Internet-Enabled Computer-Aided Design," IEEE Internet Computing, Vol. 1, No. 1, pp. 39-50, 1997.
2. Potter, C., "Web-Enabled Engineering: step-by-step," Computer Graphics World, pp. 64-69, Nov. 1997.
3. Martino, T. D., Falcidieno, B. and Haßinger, S., "Design and Engineering Process Integration through a Multiple View Intermediate Modeler in a Distributed Object-oriented System Environment," Computer-Aided Design, Vol. 30, No.6, pp. 437-452, 1998.
4. Balling, R. J. and Gale, D. L., "Collaborative Optimization of Systems Involving Discrete Design at the Discipline Level," ASME Journal of Mechanical Design, Vol. 120, No. 1, pp. 32-39, 1998.
5. Yang, Y. S., Lee, K. Y., Choi, Y. and Yeon, Y. S., Web-based Multidisciplinary Robust Ship Design, KOSEF Research Report, 2001.
6. Debras, P., Huguet, P., Amar, V. and Poyet, P., "RISESTEP (EP 20459) : a user driven project to develop a platform for product data sharing in a concurrent engineering context," Proceeding of the Product Data Technology Days 1997, French Rivera, France, pp. 23-30, 1997.
7. Charles, P., SHARE: A Methodology and Environment for Collaborative Product Development, <http://cdr.stanford.edu/SHAE/share.html>.

8. Dong, A. and Agogino, A., M., "Managing design information in enterprise-wide CAD using smart drawings," *Computer-Aided Design*, Vol. 30, No. 6, pp. 425-435, 1998.
9. Cutkosky, M. R., Tenenbaum, J. M. and Glicksman, J., "Madefast: Collaborative engineering over the internet," *Communications of the ACM*, Vol. 39, No. 9, pp. 78-87, 1996.
10. Choi, Y. and Shin, H. Y., "Sharing STEP data on the Internet," *Proceedings of the 8th International Conference on Production Engineering*, Sapporo, Japan, pp. 397-405, Aug. 18-20, 1997.
11. Choi, Y. and Yang, S. W., "Conference System for CAD Based on Product Data Standard," *Society of CAD/CAM Engineers*, Vol. 4, No. 3, pp. 200-209, 1999.
12. Kim, K. W., Jun, Y. T. and Chong, T. H., "Development of the CAD Conferencing System for Real-time Design Collaboration," *Proceedings of the KSPE*, Vol. 1, pp. 531-535, 2002.
13. Orfali, R., Harkey, D., and Edwards, J., CORBA, Java, and the Object Web, *Byte magazine*, <http://www.byte.com/art/9710/sec6/art3.htm>, Oct. 1997.
14. Parasolid Technical Support Group, *Parasolid On-line Documentation*, Apr. 1999.
15. Siegel, J., *CORBA fundamentals and programming*, John Wiley & Sons, Inc., 1996.