

분산객체를 응용한 RP Preprocessor 의 기능 구현

지해성*, 이승원**

RP Preprocessor Based on Distributed Objects

Hae Seong Jee*, Seung Won Lee**

ABSTRACT

When considering the use of rapid prototyping (RP), there are many issues a designer has to address for handling an STL model, the *de facto* standard for RP. Today designers can skip all these issues by visiting web-based service bureaus that readily supply needed information for the RP services. Since orders are taken for RP parts through the web page of service providers designers are now asked to upload their STL files to the company server either by direct upload, ftp file transfer, or as an e-mail attachment. If the service bureau, however, fixes or edits an STL file to optimize the RP process but neglects to tell its customer about the rework in detail, it may cause problems down the line in processing of the original CAD data for other applications. In this paper, we propose a framework for a collaborative virtual environment between CAD designers and RP processes on the Internet which directly provides designers with an advanced preprocessor functionality, design visualization, as well as model display, repair, and slicing over the network. This can help smooth data transfer from CAD to RP process with minimum inconsistency in CAD.

Key Words : CAD(컴퓨터원용설계), RP(쾌속조형), Internet(인터넷), Preprocessor(전처리기)

1. 서론

RP 조형장비는 학교나 연구소/일반 중소기업 체에서 직접 구입하여 운영하기에는 가격이 매우 고가이다. 일반적으로 RP 조형장비를 소유하지 않은 설계자가 조형작업을 수행하려면 통상 RP 조형관련 용역서비스를 제공하는 제작자에게 CAD 파일을(대부분의 경우 STL 파일로 주고 받음) 직접 보내거나 메일/FTP 형식을 빌려 발송하고 제작자는 이 파일을 이용하여 조형작업을 수행한 후 생성된 조형물을 클라이언트에게 직접 전달하거나 우편으로 발송하게 된다. 기존의 이러한 방식의

문제점으로는 첫째, 설계자가 갖고있는 STL 파일에 문제가 있는 경우로서 STL 모델에 기하학적인 구멍이 존재하거나 1 개의 모서리가 3 개 이상의 면을 공유하는 비다양체 파트가 포함되어 조형작업이 불가능한 경우이다. 둘째, 근본적으로 설계자-제작자간의 의견교환작업으로 인한 반복작업을 피할 수 없는 경우인데 비록 STL 모델에 아무런 문제가 없더라도 최적의 조형작업을 위해서는 과일을 받은 업체와 설계자간에 여러가지 조형 관련 정보(예를 들면 단면화 방향이나 두께처럼 조형변수이면서 동시에 설계자의 의도가 반영되어야 할)들에 관한 사전의견교환이 반복적으로 행해져야

2002년 7월 24일 접수

* 홍익대학교 기계.시스템 디자인공학과

** 주: 정소프트

한다. 셋째, 그럼에도 불구하고 설계자가 조형작업에 직접 관여할 수 있는 여지가 여전히 좁고 그에 따라 조형된 파트에 대한 설계자의 만족도가 충분히 높지 못하므로 때로는 설계자의 원래 의도대로 조형될 때까지 작업완료가 지연되는 경우가 많다.

본 논문은 전문화된 설계자-제작자 환경에서의 RP 조형 작업이 갖고있는 시간적/공간적 제한을 인터넷을 이용한 클라이언트/서버 모델 구축 환경을 통하여 극복하고자 함을 연구목표로 한다. 이를 위한 방법으로 우선 분산객체를 이용한 RP 전처리기, 즉 다수의 설계자들에게 동시에 그리고 실시간으로 전처리기의 기능을 제공하기 위한 시스템 프레임워크를 구현하고 또한 네트워크를 통하여 설계가시화기능을 제공함으로써 의사결정지원이 가능하도록 전처리기에서 구현하고자 한다. 이 기술을 이용하면 CAD 파일을 가진 설계자, 즉 클라이언트는 갖고 있는 STL 파일을 조형작업자의 서버로 전송하거나 혹은 이전에 설계자가 서버 내에 등록해 놓은 STL 파일들을 검색한 후 그 중에서 적절한 파일을 선택하여 가시화할 수 있음은 물론 CAD 파일의 오류를 검색/복구하거나 단면화 작업을 통한 2 차원 조형 중 데이터의 생성과 가시화, 그리고 가상조형영상생성작업까지도 처리하는 것이 가능하다. 이는 본 시스템에서의 클라이언트가 조형작업에 수반되는 제반 조형 파일 처리기능을 네트워크를 통하여 서버로부터 제공 받을 수 있기 때문이다. 피상적인 단견으로 보면 의뢰자와 RP 작업자가 각각 일반적인 preprocessor를 보유하고 파일 또는 DB를 공유할 수 있다면 이 문제가 해결 될 수 있다고 생각할 수 있으나 본 연구의 필요성은 바로 그와 같은 파일 또는 DB를 분산환경에서 공유해야 한다는 필요성에서 출발하고 있는 것이다.

지금까지 네트워크를 RP 조형작업에 적용한 국내외 연구사례는 인터넷을 이용한 RP 장비의 원격 제어에 관한 연구 등 몇 가지가 발표되었으나^[1,2,3] 이러한 연구들은 STL의 단순 가시화와 파일수정, 견적서 의뢰, 그리고 RP 장비 및 공정의 원격제어와 같은 전문분야를 대상으로 하는 것이 목적인데 반하여 본 연구는 일반 설계자와 조형자의 쌍방의견수렴을 통한 설계 및 RP 조형환경의 통합을 연구논문의 주제로 삼고자 한다.

2. RP 전처리기와 분산환경

2.1 RP 전처리기의 개념정의

현재 대부분의 RP 조형장비는 'STL' 파일이라는 매우 단순한 형상 정보 교환 체계에 기반을 두고 운용이 되는 까닭에 RP 조형 작업을 수행하기 위해서는 설계된 제품형상정보의 3 차원 CAD 모델을 상용화된 RP 조형장비들에 장착된 운용소프트웨어 프로그램이 인식할 수 있도록 STL 파일로 변환해주어야 한다. 3 차원 CAD 모델이 STL 파일로 변환되는 과정에서 파일의 정확성(accuracy), 완전성(integrity), 중복성(redundancy)으로 대별되는 문제점들이 보고되어 왔는데 이를 부연하면 STL은 일반적인 CAD 모델의 외부경계를 삼각형들의 중복된 집합으로 근사화하여 재구성한 것이라는 점(정확성), 그 자료구조는 하나의 삼각형에 대한 세 개의 꼭지점 정보와 삼각형 면의 법선 벡터들을 나열한 형태의 단순한 반복으로 이루어져 있다는 점(중복성), 그리고 서로 다른 CAD 시스템내의 자체 변환기로 기하학적인 정보를 STL 파일로 변환하기 때문에 꼭지점 좌표의 중복, 임의의 삼각형 면의 누락이나 중복, 혹은 뒤바뀌어진 법선벡터 등의 뒤바뀐과 같은 여러 가지 오류가 발생할 수 있다는 점(완전성)이 바로 그것이다. 그와 같은 기하학적인 오류이외에도 STL 파일에는 위상정보가 전혀 포함되어 있지 않으므로 파일을 적절하게 처리하여 RP 조형작업에 필요한 2 차원 중 단면 정보들을 올바르게 생성하기 위해서는 STL 파일을 읽어들인 후 이를 화면에 가시화하고, 오류를 검색/수정하며, 파트들을 편집한 후, 최종적으로 단면화 작업을 행하기 위한 RP 조형 전처리기 소프트웨어가 필요하다. 이와 같은 전처리기 소프트웨어는 대개 간단한 모듈형태로 상용 RP 장비들에 내장되어 판매되고 있는데 대부분 적절한 파일처리를 위해서 STL 자료표현을 보다 완전한 B-Rep. 표현으로 바꾸어준 후에 기타 작업을 수행하게 된다. Fig. 1 은 이와 같이 CAD 시스템에서 생성된 기하학적 CAD 형상 정보를 STL 파일로 변환한 후 RP 전처리기에서의 여러가지 작업을 거쳐 RP 조형 작업에 필요한 2 차원 중 단면 정보를 생성하기까지의 일련의 과정을 도시하고 있다.

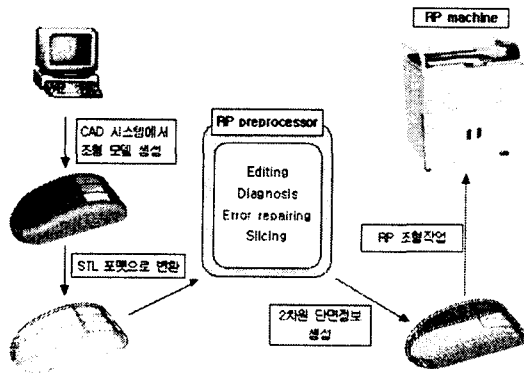


Fig. 1 Environment for RP preprocessor

2.2 분산환경에서의 RP 전처리

본 논문의 연구주제는 RP 전처리를 분산객체 기반의 클라이언트/서버시스템으로 구현함과 동시에 기존의 전처리에서는 네트워크상에서의 구현이 시도되지 않았던 의사결정지원기능기능을 추가하여 조형 작업의 공간적/시간적 제한을 극복하고, 동시에 여러 클라이언트들에게 간편하게 전처리 기능을 제공하고자 하는 것이다. RP 전처리를 클라이언트/서버환경으로 구축하는 이유는 통상 전처리에서 수행되는 작업이 컴퓨터 성능에 크게 좌우된다는 점을 고려할 때 클라이언트보다 상대적으로 성능이 뛰어난 서버에서 작업을 수행하고 그 작업수행 결과만 클라이언트에게 전송함으로써 전체 작업 효율을 높이고자 하는 목적과 또한 향후 네트워크 환경에 대비한 웹 브라우저를 통한 RP 전처리 기능활용환경의 구축이라는 목적에서 출발한다. 또한, RP 전처리를 클라이언트/서버시스템으로 구축함에 있어서 분산 객체 기술을 기반으로 한 소프트웨어의 재사용성, 이식성, 상호연동성의 향상을 도모하고, 부가적으로 분산 객체 기술에서 일반적으로 지원하는 고급언어 바인딩, 지역/원격 투명성, 기존 시스템과의 공존, 내장된 보안과 자료처리 기술구축의 효과를 얻고자 한다.

3. CORBA 기술과 RP 전처리 구현

3.1 CORBA 기술과 분산환경 소개

최근의 CAD 작업 환경에서는 네트워크 기술의 급속한 발전으로 인하여 필요한 정보를 서로 공유

할 수 있게 되었으며, 특히 분산객체 기술의 도입으로 말미암아 클라이언트/서버방식을 통하여 네트워크 상에 존재하는 여러 응용 프로그램들의 모듈들이 공유될 수 있고 하드웨어 플랫폼에 관계없이 시스템이 유연성 있게 유지될 수 있게 되었다. 이러한 분산객체 기술을 통하여 저 수준의 통신 인터페이스를 따로 구성할 필요가 없게 되었으며, 객체지향형 프로그래밍의 개념에 충실하게 따를 수 있게 되어 모듈의 재사용이 수월하다. 또한 기존 클라이언트/서버 시스템에서는 하나의 서버에 동시에 여러 클라이언트가 접속됨에 따라 높은 부하가 걸리고 전체 네트워크 성능이 저하되는 문제점이 있었으나 분산객체 기반 클라이언트/서버 시스템에서는 여러 개의 서버를 설치하여 각 서버마다 필요한 객체모듈만 설치하고 이들을 잘 정의된 프로토콜로 연결하면 시스템에 접속하는 클라이언트의 연결이 분산되어져 전체 네트워크의 성능이 훨씬 높아지게 됨은 물론, 서버의 부담이 크게 줄기 때문에 상대적으로 저렴한 비용으로 시스템을 구축할 수 있는 장점이 있다.

CORBA(Common Object Request Broker Architecture)는 바로 이와 같은 이종의 분산된 환경하에서 응용 프로그램들이 통합되어 운용되는 소위 분산 컴퓨팅 시스템의 개발을 위하여 제정된 객체 지향 표준으로서 1989년 800 개 이상의 관련 단체가 모여서 결성한 컨소시엄인 OMG(Object Management Group)가 제안한 OMA (Object Management Architecture) 중 일부이다. CORBA는 하드웨어와 운영체제에 구애됨이 없이 사용이 가능하고, 다른 언어로 작성된 프로그램 사이에서 서로에게 객체를 전달할 수 있으며, 객체의 메소드 호출도 가능하다. CORBA 클라이언트와 서버 객체간의 통신에 있어서 외부에 노출된 서버의 인터페이스는 CORBA 표준 인터페이스 언어인 IDL (Interface Definition Language)로 표현된다. 또한 IDL로 표현한 인터페이스의 내용은 해당 컴파일러를 이용하여 JAVA, C++ 등의 언어로 변환해 바인딩 됨으로써 CORBA가 서로 다른 언어의 한계를 뛰어 넘을 수 있게 된다^[4,5]. 이러한 장점 때문에 CORBA는 분산환경에서의 일반적인 제품설계와 정보관리 및 평가에 적용하거나^[6-12], 영상자료 뷰어개발과 모니터링을 위한 이미지 프로세싱^[13-15], 그리고 제어^[16]에 이르기까지 다양한 응용분야에서의 구현기술로 적용되고 있다.

3.2 CORBA 기술을 적용한 RP 전처리기 구현

전술한 분산객체를 전제로 한 RP 전처리기의 구현을 위한 CORBA 클라이언트/서버간의 데이터 교환관계를 Fig. 2 에서 도식화하였다.

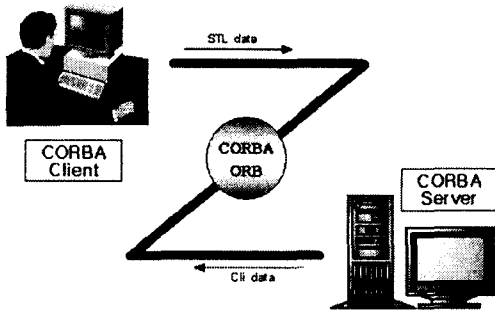


Fig. 2 Data exchange in CORBA server/client system

RP 전처리기를 분산객체 기술인 CORBA 를 이용하여 클라이언트/서버환경으로 구성하여 얻는 장점을 정리하면 일반적인 분산객체기술의 장점과 매우 흡사하나 여기에서 다시 서술하면 다음과 같다.

첫째, 기존 RP 전처리기의 기능을 각각의 모듈 별로 CORBA 객체로 wrapping 하여 기존 시스템을 쉽게 재 사용할 수 있기 때문에 개발시간과 비용을 크게 단축시킬 수 있다.

둘째, 클라이언트/서버환경의 구축으로 인하여 간편하게 동시에 여러 클라이언트에게 RP 전처리기 서비스를 제공할 수 있다.

셋째, CORBA ORB 를 이용하여 네트워크를 구현하기 때문에 개발자는 복잡한 통신 모듈을 개발하지 않아도 되며, CORBA ORB 에서 제공하는 자료흐름의 원활함과 안전성덕분에 전체 시스템이 아주 견고해지며 신뢰성이 높아지게 된다.

넷째, 차후에 새로운 모듈의 추가/확장이 간편하게 된다. 이는 CORBA 기술이 내세우는 큰 장점중의 하나로써 운영환경과 개발언어의 종류에 상관없이 확장 모듈을 개발하고 이를 CORBA 객체로 wrapping 하여 전체 시스템에 추가 하다는 것을 의미하며 따라서 새로운 모듈의 추가로 인하여 기존 시스템을 고칠 필요가 없으므로 개발 생산성이 크게 향상된다.

다섯째, 클라이언트/서버구조에서 서버의 구현 객체를 여러 곳에 분산시킴으로써 서버의 부담을

줄이고 전체 네트워크의 효율을 증대시키며, 또한 서버 구축비용을 낮출 수 있다.

여섯째, 다른 여러 분산객체 시스템과 서로 연동이 가능하므로 시스템의 확장이 무한하다.

예를 들어, 사용자가 STL 파일의 진단기능을 사용하고자 할 때, CORBA 클라이언트는 내부적으로 해당 STL 모델의 자료를 CORBA 서버로 보내어 오류 검색을 수행 하도록 하며 그 결과 값을 돌려 받게 된다. 또한 단면화 작업을 수행할 경우에도 같은 원리가 적용되어 실제 작업은 CORBA 서버가 담당하고 단면화 작업 결과만 CLI 자료구조로 클라이언트에게 넘겨준다

3.3 분산환경에서의 의사결정지원 기능의 구현

본 논문에서 제시된 전처리기의 독창적인 기능중의 하나가 바로 의사결정지원(decision support) 기능을 네트워크를 통해서 구현한다는 개념이다. 이는 STL 파일로 주어진 설계형상이 실제 물리적인 조형으로 구현되었을 때 발견될 수 있는 기하학적인 결함들, 즉 표면거칠기와 조형가능성에 대한 가시화기능을 네트워크를 통해 지원하여 사용자 하여금 조형결정 및 설계변경을 보다 용이하게 하도록 지원하고자 하는 개념이다. 이는 5 장의 구현사례에서 제시된 바와 같이 주어진 CAD 형상에 대한 2 차원 층단면화 결과에 따른 최소/최대 조형층 방향을 자동으로 결정해주는 것을 포함하며 특히 본 논문에서는 계단형상오차를 정량화한 후에 이를 가시화해주는 기능을 구현한 점이 다른 의사결정지원 기능과 대별된다.

주지하다시피 모든 RP 조형파트의 표면에는 Fig. 3a 에서 보듯이 소위 계단형상에 의한 표면정도의 결함이 발견된다. 이와 같은 결함들은 조형파트의 형상특징과 조형방향 및 조형층 두께와 밀접한 관계가 있으나 이를 설계자가 조형 전에 미리 파악하는 것은 용이하지 않다. 이러한 점을 극복하기 위하여 조형형상 이미지를 사전에 예측하여 이를 가시화하는 가상조형기능은 이전에 제시되어 왔다^[7]. 본 논문에서는 이와 같은 가상조형의 알고리즘을 보다 구체화된 가시화 알고리즘으로 발전시키는 동시에 이를 본 논문에서 네트워크를 통하여 구현, 사용자에게 제공하고자 하는 것이다. 이를 위해 조형파트에 발견될 계단형상을

미리 2 차원 단면 층으로부터 예측하여 이를 근거로 각 조형 층간의 계단형상오차를 측정한다. 우선 Fig. 3b 에서 보여지듯이 주어진 CAD 형상을 구성하는 임의의 조형층들을 조형층 형상의 평면도를 기준으로 서로 겹치도록 가시화한 후에 단면 무게중심을 기준점으로 정한다. 이 점으로부터 방사선 방향으로 일정한 각도의 방사선들이 작도되면 임의의 방사선에 대하여 모든 방향의 단면형상 경계간의 계단형상오차가 구해질 수 있다.

이를 기준으로 하여 모든 방사선에 대하여 평균계단형상오차를 다음과 같이 계산하고 이 오차들의 상대적인 크기를 이용하여 Fig. 4 와 같은 조형단면층의 계단형상오차를 전체형상에 대하여 서로 다른 색채로 가시화가 가능하다. 이러한 기능은 전술한 바와 같이 설계자가 직접 판단하기 곤란한 설계변수를 일단 정량화 한 후에, 이를 설계자의 시각에서 보다 효과적인 가시화라는 수단으로 그리고 무엇보다 네트워크를 통해서 제시한다는 점에서 하나의 독창적인 의사결정지원기능으로 본 논문에서 제시하고자 하는 바이다.

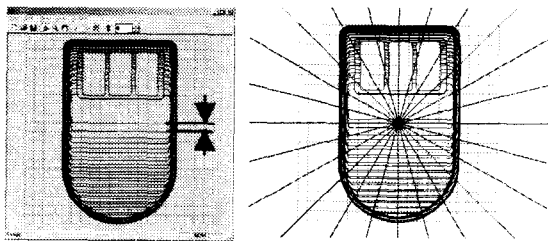


Fig. 3 An illustration of estimating staircase cusp size between two adjacent layers

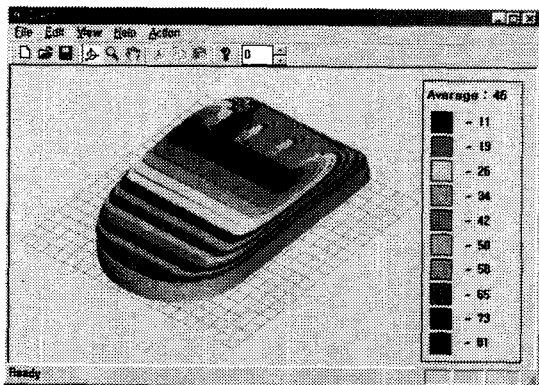


Fig. 4 Virtual prototyping of 3D layers with color for decision support

4. 시스템 구성

4.1 구현환경 및 도구

본 논문에서 제안한 시스템의 서버모듈은 윈도우 2000 서버 운영체제에서 그리고 클라이언트 모듈은 윈도우 98 운영체제에서 동작되도록 구현되었다. 개발 도구로는 MFC/ Visual C++ 6.0 을, 그리고 3 차원 그래픽 라이브러리는 OpenGL 을 이용하였으며^[18,19] CORBA 사용을 위한 ORB 로는 IONA 사의 Orbix 3.0.1 을 적용하였다^[20].

4.2 시스템 구조

본 논문에서 제안한 분산객체 기반 RP 전처리기의 시스템구조에 따른 수행 작업 과정을 서술하면 다음과 같다. 사용자는 우선 CORBA 클라이언트 모듈을 사용하여 RP 전처리기 서비스 수행 대상인 STL 모델을 입력받아 클라이언트의 가시화기능으로 해당 모델을 검사하거나 진단모듈을 이용하여 STL 파일의 오류포함여부를 검색한다. 만약 오류가 검색되면 해당 오류를 사용자에게 알려서 자동으로 수정/복구하지 아니면 수동으로 처리할 지를 묻는다. 오류가 모두 복구되면 2 차원층 단면정보를 얻기 위해 단면화 작업을 수행할 층 두께를 입력할 것을 요구하거나 의사결정지원기능을 이용하여 각 조형모델의 형상특성에 따른 최적의 단면화 작업의 방향을 결정할 수 있도록 정보를 제공한다. 이밖에도 전술한 여러가지 형상편집 기능들도 각각 모듈화 된 CORBA 객체로써 근거리컴퓨터가 아닌 원격지 컴퓨터에 네트워크로 연결되어 작업을 수행한다. 전체 시스템을 살펴보면 사용자에게 인터페이스를 제공하는 CORBA 클라이언트, 클라이언트로부터 명령을 받아서 실제 작업을 수행하는 CORBA 서버, 그리고 이들을 분산객체 시스템으로 구성하여 주는 미들웨어인 ORB, 이렇게 세 가지 파트로 나눌 수 있고, 이들은 모두 IDL 에 의해서 정의된 CORBA 객체의 정의에 따른다.

4.3 RP 전처리기 구현용 IDL 파일구조

IDL 은 CORBA 에서 객체를 기술하게 되는 인터페이스 정의 언어로서 어떠한 특정 언어뿐만 아니라, 다양한 프로그래밍 언어로 작성된 객체들간의 서비스 교환이 가능하도록 한다. 본 논문에서 구현한 분산객체 기반 RP 전처리기 시스템의 명

칭은 CyberRP_NET 로 하였고 IDL 파일의 내용은 Fig. 5 와 같다.

```
#ifndef __CYBERRP_IDL__
#define __CYBERRP_IDL__

interface CyberRP_NET
{
    typedef sequence<char> charSeq;
    long corba_set_client_index();
    Boolean corba_send_stl_file(in long client_index, in
string name, in charSeq data);
    string corba_checkererror_and_autorepair(in long
client_index);
    void corba_decision_support(in long client_index);
    long corba_slicing(in long client_index, in float layer,
out charSeq data);
    void corba_disconnect(in long client_index);
};

interface CyberRP_NET_Factory
{
    CyberRP_NET newCyberRP_NET();
    Void freeCyberRP_NET(in CyberRP_NET
aCyberRP_NET);
};

#endif
```

Fig. 5 IDL file for preprocessor

4.4 RP 전처리기 구현용 CORBA 서버

분산 객체 시스템의 장점중의 하나는 기존 시스템을 CORBA 인터페이스로 wrapping 하여 객체화 할 수 있다는 것이다. 이에 따라 본 논문의 CORBA 서버는 Fig. 6 에서 보듯, 기존의 RP 전처리기의 기능들을 각각 독립적인 모듈로 구성하여 IDL 에서 정의된 인터페이스대로 wrapping 하여 구현하였으며 이는 Fig. 7 에 나타나 있는 것처럼, RP 전처리기 시스템의 원래 기능 중에서 diagnosis/repair/decision support/slicing 모듈 등을 분리하여 IDL 로 정의하고, 독립적인 구현객체로 개발한 것이다.

이와 같은 구현객체를 통해서 클라이언트들의 서비스 요청을 수행할 목적으로 제공되는 기능들은 전술한 바와 같이 클라이언트로부터 STL 파일을 전달받아 자료구조로 저장하고 가시화하는 기능을 비롯, 다수의 전처리기가 갖추어야 할 일반적인 기능들이 포함되며 최종적인 단면화 작업 결과는 CLI 자료구조의 형태로 다시 해당 클라이언트에게 되돌려 준다. 이처럼 기존 RP 전처리기 시스템 기능들을 CORBA 서버에서 구현하기 위해 wrapping 한 부분은 서로 완전히 독립적으로 구현

되어 모듈들의 독립성을 최대한으로 유지시켰는데 예를 들어 기존 RP 전처리기 부분과 CORBA 연결/실행 부분은 서로 독립적인 트래드로 구성하여 나중에 다른 모듈을 추가하거나 기존 모듈을 수정하기가 용이하도록 구성하였다.

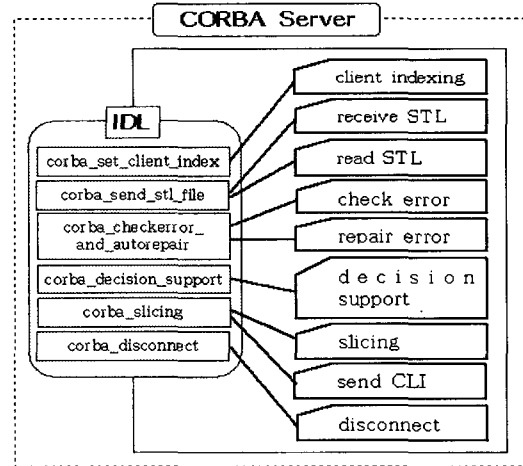


Fig. 6 CORBA server site window for RP preprocessor

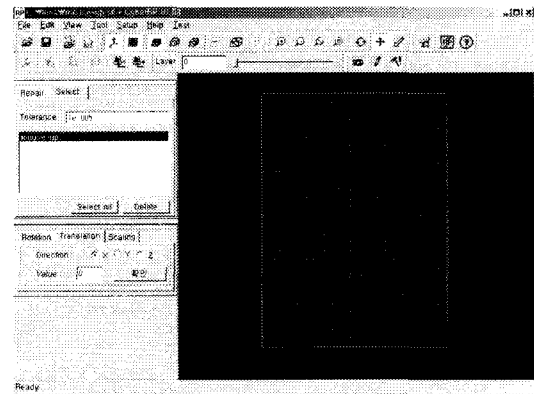


Fig. 7 CORBA Server wrapping

한편, CORBA 서버는 여러 클라이언트들에게 동시에 서비스를 제공해야 하므로 각 클라이언트마다 전달 트래드를 두어서 트래드별로 작업을 진행하는 다중트래드 시스템이 되어야 하지만, 클라이언트/서버 시스템에서 둘 이상의 클라이언트가 서버상의 같은 공통자원을 동시에 접근하여 입력, 출력, 수정 등의 작업을 할 때 충돌이 일어나게 되는 이른바 동기화 문제가 발생할 수 있다. 이런 동기화 문제를 해결하기 위해서 MFC(Microsoft

Foundation Class)의 다중 트레드 동기화 객체인 Mutex 를 사용하였는데 이로써 임의의 클라이언트가 서버의 특정 공통자원에 접근을 시도할 때에는 동기화 객체를 이용하여 해당 공통자원을 폐쇄하며 작업이 끝난 후에야만 다시 개방하여 다른 클라이언트로 하여금 해당 공통자원에 대한 접근을 허용하게 된다.

4.5 RP 전처리기 구현용 CORBA 클라이언트

CORBA 클라이언트의 기본적인 역할을 서술하면 우선 STL 형상모델의 가시화기능과 서버로부터 오는 CLI 포맷의 데이터의 가시화 기능들은 자체적으로 구현되도록 설정되어있고, 그 밖의 다른 기능들(display/diagnosis/repair/decision support/slicing)에 대해서는 사용자로부터 명령을 받아서 IDL 에 정의되어 있는 CORBA 구현객체를 이용하여 CORBA 서버에 위치하는 해당 모듈을 호출하여 처리하도록 구현되어 있다.

5. 구현 사례

이 장에서는 본 논문에서 제안한 분산객체 기반 RP 전처리기의 실제 구현 사례를 서술하고자 한다. 시스템의 동작은 사용자가 클라이언트 프로그램을 실행하면서 시작되며 이때 CORBA ORB 는 CORBA 클라이언트가 CORBA 서버의 위치를 찾을 수 있게 해준다. Fig. 8 은 서버에서 실행되는 CORBA daemon 에서 클라이언트가 접속하는 상황과, 역시 서버의 CORBA 서버운영자에 클라이언트의 정보가 등록되어져 있는 과정을 보여준다.

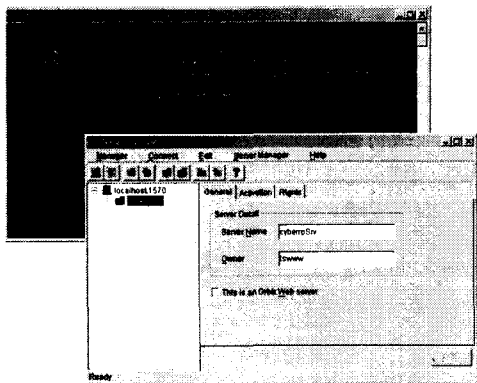


Fig. 8 Orbix server daemon & server manager

클라이언트가 서버의 위치를 확인하게 되면 서버로부터 구현객체의 참조를 요구하게 된다. 클라이언트는 이때 받은 구현객체의 참조를 이용하여 IDL 에 명시되고 서버측에 구현되어져 있는 메소드와 속성에 접근할 수 있게 된다. Fig. 9 는 사용자가 서버로부터 구현객체의 참조를 얻은 후 자신의 STL file 을 가시화하는 것을 보여주며 여러 부분에서 기하학적 오류(작은 흑색 삼각형과 반점)가 검출될 수 있다. 검출된 기하학적 오류들은 수정/복구되어 다시 클라이언트에게 전달된다.

Fig. 10 은 오류가 수정/복구된 STL 파일을 사용자가 서버로 보내어 층 단면화 작업을 수행한 후에 그 결과를 다시 서버로부터 CLI 자료구조의 형태로 받아서 와이어프레임으로 2 차원 층단면별로 모니터에 가시화하는 장면이며, Fig. 11 은 클라이언트의 가상조형기능을 이용하여 RP 조형작업의 수행결과로 얻어질 파트의 모습을 음영처리하여 가시화 한 이미지이다.

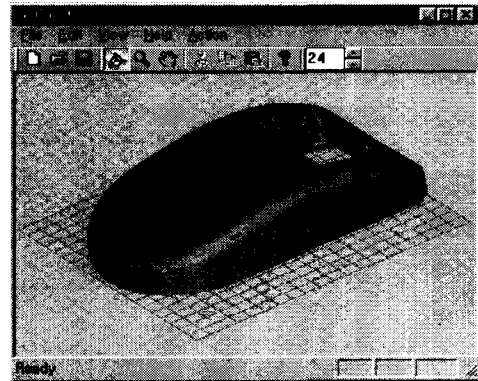


Fig. 9 STL model display (display & diagnosis)

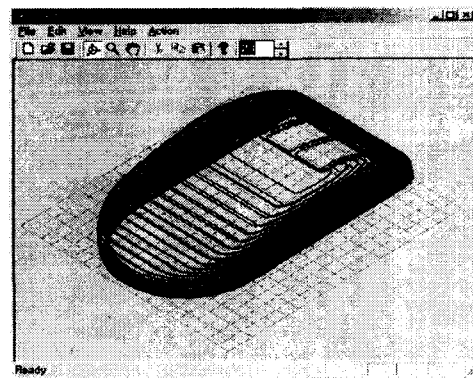


Fig. 10 2D layer display after slicing

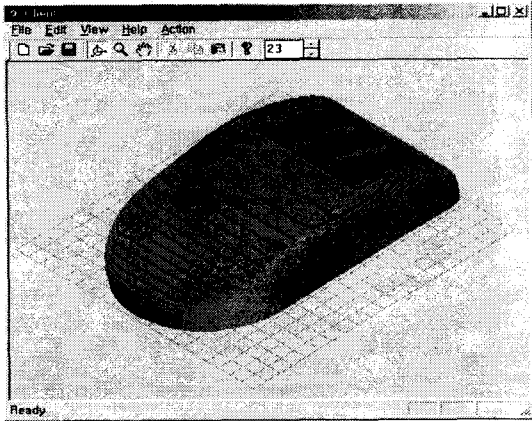


Fig. 11 Virtual prototyping of 3D layer after slicing

Fig. 11 에서 보듯이 가상조형작업 결과로 얻어진 이미지를 이용하여 RP 조형물의 표면계단부위에 의한 조형면 오차를 관찰하는 것이 가능한데 이를 기준으로 조형면 오차를 최소화 하기위한 최적의 조형 방향을 결정하기 위해 의사결정지원기능모듈을 실행할 수 있다. 즉 Fig. 12 와 Fig. 13 에서는 각각 의사결정지원기능모듈에서 계산되어진 최소/최대 조형층 개수를 갖는 방향에서의 층 단면화된 모델의 이미지를 보여준다. 한편 전술한 바와 같이(Fig. 4) 임의의 방향에 대한 계단형상오차를 그 크기별로 색깔을 할당하여 모델을 가시화하는 가상조형기능을 통하여 설계자로 하여금 인접한 조형층 별로 계산되어진 각각의 층별 조형면 오차 분포를 직관적으로 예측할 수 있게 해주며 그에 따라 직접적인 판단이 용이하지 않은 설계변수들을 자동적으로 설정해 주거나 혹은 이를 정량화 및 효과적인 가시화를 통하여 설계자에게 직접 제공하여 설계자의 결정을 돕는 것이 가능하다.

Fig. 14 는 이와 같은 가상조형 알고리즘을 다른 CAD 형상에 적용한 것으로 CAD 형상의 형상 특성이 복잡할수록 이와 같은 의사결정지원기능의 효용성이 상대적으로 보다 더 명확하게 대비됨을 느낄 수 있다.

6. 결론

본 논문은 기존 RP 전처리기의 기능을 각각 모듈별로 분리하고 이를 분산객체 기술인 CORBA 객체로 wrapping 하여 기존 시스템의 재사용/뛰어

난 확장성/플랫폼 독립성/개발언어의 독립성 등의 장점을 제공한다. 또한 사용자로 하여금 가장 효율적인 조형 작업을 할 수 있도록 도움을 주는 의사결정지원 시스템 모듈을 새롭게 개발하여 분산 객체화 시켜서 시스템에 추가하였다.

이처럼 구성된 분산객체 시스템을 서버에 구현하게 되면 사용자는 클라이언트 프로그램이 제공하는 사용자 인터페이스와 인터넷을 통하여 CORBA 서버에 접속, 기본적인 RP 전처리의 기능과 설계자에게 최적의 2 차원 단면 정보를 얻을 수 있도록 도움을 줄 수 있는 의사결정지원기능까지 함께 전문적인 서버로부터 네트워크를 통해 제공받게 되어 설계로부터 조형단계에 이르기까지의 일관화 작업이 보다 효율적으로 이루어 지며 될 것으로 사료된다. 단 네트워크의 속도와 보안 문제는 본 논문과는 별도로 고려되어야 할 것이다

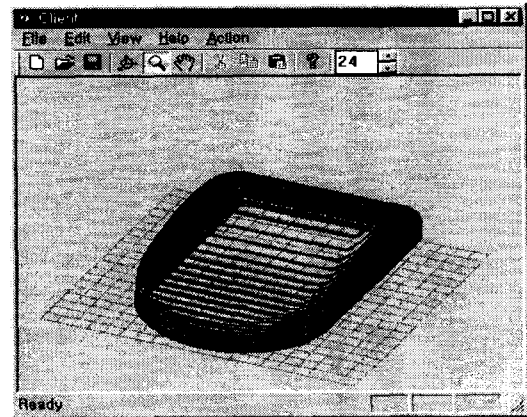


Fig. 12 2D layer display when minimum layer number

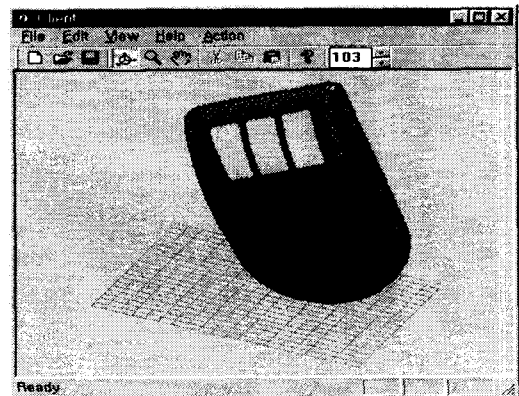


Fig. 13 2D layer display when maximum layer number

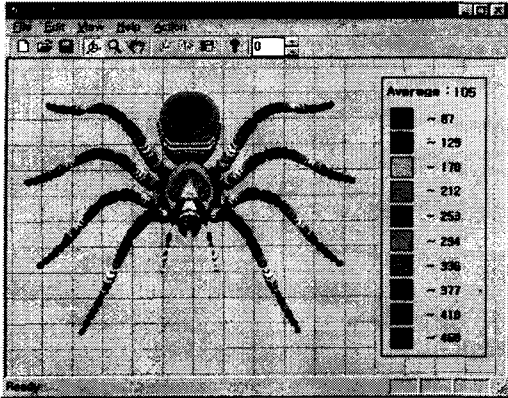


Fig. 15 Virtual prototyping of 3D layers with color for decision support

후 기

이 논문은 2001 년도 홍익대학교 학술연구 조성비에 의하여 연구되었음.

참고문헌

1. Bailey, J. M., "Tele-Manufacturing: Rapid Prototyping on the Internet," IEEE Computer Graphics & Applications, Vol. 15, No. 6, pp. 20-26, 1995.
2. 이석희, 이주호, 허성민, 최홍태, "인터넷을 이용한 RP 의 상호협동시스템 구현," 한국정밀공학회 춘계학술대회 논문집 pp. 909-912, 1998.
3. Luo, C. R. et. Al, "Tele-Control of Rapid Prototyping Machine Via Internet for Automated Tele-Manufacturing," IEEE International Conference on Robotics and Automation, 3, pp. 2203-2209. 1999.
4. Robert Orfali, Dan Harkey, Jeri Edwards, Instant CORBA, WILEY, 1997.
5. Jon Siegel, CORBA 3 : Fundamentals and Programming, John Wiley & Sons, 2000.
6. Pahng, F., Senin, N. and Wallace, D., "Distributed Modeling and Evaluation of Product Design Problems," Computer-Aided Design, Vol. 30, No. 6, pp. 411-423, 1998.
7. 김영호, 강석호, 이수홍, 유상봉, "분산, 개방, 지능형 제품정보관리시스템," 한국 CAD/CAM 학회 논문집, 제 4 권, 제 3 호, pp. 210-223, 1999.
8. 복기소, 명세현, 한순홍, "분산 지식베이스에 기반한 렌즈 경통 설계," 한국 CAD/CAM 학회 논문집, 제 5 권, 제 1 호, pp. 1-11, 2000.
9. 양상욱, 최영, "실시간 원격 협력 설계 시스템 - DoDes," 한국 CAD/CAM 학회 논문집, 제 5 권, 제 1 호, pp. 42-49, 2000.
10. 이동욱, 이수홍, "분산환경에서 CORBA 를 이용한 에이전트기반 초기설계 시스템," 한국 CAD/CAM 학회 논문집, 제 5 권, 제 4 호, pp. 373-379, 2000.
11. 방건동, "네트워크 기반 엔지니어링 환경에서의 제품 개발: 개념과 접근방식," 한국 CAD/CAM 학회 논문집, 제 5 권, 제 1 호, pp. 79-87, 2000.
12. 이경호, 이순섭, 이종갑, "인터넷 기반의 원격 협동 선박설계 시스템," 한국 CAD/CAM 학회 논문집, 제 6 권, 제 3 호, pp. 198-205, 2001.
13. 손정모, 유상봉, 김영호, 이수홍, "분산환경에서 표준제품모델(STEP)을 이용한 내용검색," 한국 CAD/CAM 학회 논문집, 제 4 권, 제 4 호, pp. 285-294, 1999.
14. W.L. Chan, A.T.P. So and L.L. Lai, "Internet Based Transmission Substation Monitoring." IEEE Transactions on Power Systems, Vol.14, No.1, pp. 293-298, February 1999.
15. Petrick, Nicholas; Chiao, Ping-Chun; Clinthorne, Neal H. "Distributed framework for delivery of medical image processing algorithms," Progress in biomedical Optics and Imaging Medical Imaging 2001 PACS and Integrated Medical Information systems : Design and Evaluation, pp. 222-228, 2001.
16. Cork, C. Timossi, C., "Distributed Control of Protein Crystallography Beamline 5.0 Using CORBA," International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, Italy, pp. 486-487, 1999.
17. Jee, H., "Virtual Models for 3D Printing," 한국 CAD/CAM 학회 논문집, 제 4 권, 제 1 호, pp. 1-11, 1999.
18. George Shepherd, Scot Wingo, MFC Internals, Addison-Wesley Developers Press, 1996.
19. Kruglinski, David J., Inside Visual C++ 5.0 and Internet Development, Microsoft Press, 1997.
20. IONA technologies, Orbix C++ Programming Guide, IONA technologies TLS, 1999.