

네트워크 기반 특징형상 모델링

이재열*, 김 현**, 한성배***

Network-based Feature Modeling in Distributed Design Environment

Lee, J. Y.*, Kim, H.** and Han, S.-B.***

ABSTRACT

Network and Internet technology opens up another domain for building future CAD/CAM environment. The environment will be global, network-centric, and spatially distributed. In this paper, we present an approach for network-centric feature-based modeling in a distributed design environment. The presented approach combines the current feature-based modeling technique with distributed computing and communication technology for supporting product modeling and collaborative design activities over the network. The approach is implemented in a client/server architecture, in which Web-enabled feature modeling clients, neutral feature model server, and other applications communicate with one another via a standard communication protocol. The paper discusses how the neutral feature model supports multiple views and maintains naming consistency between geometric entities of the server and clients. Moreover, it explains how to minimize the network delay between the server and client according to incremental feature modeling operations.

Key words : Feature-based Modeling, Network-based CAD, Distributed Modeling, CORBA

1. 서 론

컴퓨터 네트워크 및 인터넷의 발전은 현재의 CAD/CAM 방법론을 급속히 변화시키고 있다. 뿐만 아니라, 제품설계 및 가공 등과 관련된 전문적인 환경 또한 변화시키고 있다. 그 환경은 글로벌화되고, 네트워크 기반의 분산된 환경을 의미한다. 이러한 변화는 제품 설계 과정에서 지리적인 한계 및 시간적인 한계를 극복하여 보다 효과적인 커뮤니케이션을 가능케하고, 보다 폭넓은 제품 정보를 획득하고 교환할 수 있도록 도와주고 있다. 특히, WWW 기술의 발전과 보편화로 인해 향후에는 네트워크 및 분산환경 하에서 제품설계 및 이와 관련된 많은 작업들이 이루어질 것이다^{1,5,6,13,14}.

한편, 엔지니어링 관습도 변화하고 있는데, 제품생산과 관련된 많은 일들이 초기 설계과정으로 옮겨오

고 있다^{9,12,16}. 그 이유는 비록 초기설계과정의 비용은 적으나 그 결과에 따른 생산비용은 엄청나게 차이가 나기 때문이다. 따라서, 초기설계와 추후생산과정과의 효율적인 통합방법론 개발에 많은 초점이 맞추어졌다. 특히, 특징형상기술은 설계와 그와 관련된 엔지니어링 활동을 실질적으로 통합시킬 수 있는 방법론으로 인식되었다^{11,13}. 따라서, 특징형상모델링, 특징형상인식 및 추출, 특징형상변환 등에 관해서 많은 연구가 진행되고 있다^{3,4,15,17,18,21,22}.

한편으로, 솔리드 모델러 및 CAD/CAM 시스템들은 분산환경 하에서 운용되고 있으며, 분산환경 하에서 일련의 작업들이 진행되도록 지원하고 있다^{14,19}. 만약 한 제품을 설계 및 가공하는데 모든 참여자들이 똑같은 소프트웨어와 하드웨어 장비를 사용해야 한다면 이는 바람직하지 못할 뿐 아니라 비효율적인 환경이 된다. 따라서, 효과적이고 올바른 공동작업을 위해서는 각 소프트웨어/하드웨어가 모듈화를 구현해야 하며 네트워크 상에서 서로의 정보를 교환할 수 있어야 한다⁷.

이에 따라 가상 공동작업 환경하에서 설계, 가공

*정회원, 한국전자통신연구원 김소연 동시공학연구팀
**중신회원, 한국전자통신연구원 김소연 동시공학연구팀
***한국전자통신연구원 김소연 동시공학 연구팀

및 생산과 관련된 일들을 동시에 처리할 수 있는 방법론 개발에 많은 연구가 이루어져 왔다. 여기에는 형상모델링과 응용시스템과의 커뮤니케이션 표준화^{4,17}, 분산환경 하에서 가공 서비스¹⁸, 가상기업간의 공동작업을 위한 인프라스트럭처⁵, 분산환경 하에서의 설계 및 생산결합¹², CORBA를 이용한 STEP 가시화²⁰, 분산환경 하에서 설계 및 프로세스 통합¹¹ 등의 많은 연구 결과들이 있었다.

하지만, 많은 방법론들이 개념적일 뿐, 구체적이고 상세한 방법론을 제시하지는 않았다. 예를들면, 효율적인 커뮤니케이션과 공동작업을 위하여 컴포넌트들 간에 필요한 프로세싱을 어떻게 분산시킬 것인가, 어떻게 하면 네트워크의 부하를 최소화시키면서 분산 모델링을 유효하게 수행할 수 있는가 등의 문제점이다. 따라서 이러한 환경변화에 부합하고, 설계과정상의 여러 문제점을 동시에 해결하기 위한 구체적인 방법론에 관해서는 여전히 많은 연구가 필요하다.

본 논문은 이러한 분산모델링과 관련된 해결 방법론 중의 하나로 네트워크 기반 특징형상 모델링 방법론을 제시한다. 즉, 기존의 특징형상 모델링을 분산환경 하에서 실현시키고자 한다. 제시된 방법론은 클라이언트/서버 구조를 지니고 있으며, 솔리드 모델러에 기반을 둔 중립 특징형상 모델링 서버와 Web 상에서 실행 가능한 특징형상 모델링 클라이언트들이 CORBA를 기반으로한 표준 인터페이스를 통해서 특징형상 모델링이 가능하도록 구현되었다.

본 논문은, 1) 중립 특징형상 모델링 서버, 2) 클라이언트 프로세싱, 그리고 3) 분산 특징형상 모델링에 따른 네트워크 지연을 최소화시키기 위한 방법론에 중점을 둔다. 서비스 공급자의 역할을 하는 중립 특징형상 모델링 서버는 특징형상 모델링 및 다른 어플리케이션에 필요한 여러 기능들을 제공하며, 다수의 클라이언트들이 공유한다. 특히, 특징형상 간섭 문제를 해결하고 서버와 클라이언트 사이의 일관성을 유지하기 위해서 고유이름 관리기법(Generic Naming Scheme)을 도입한다. 또한 가벼운(Thin) 클라이언트를 실현시키고 클라이언트 프로세싱을 효과적으로 돕기 위하여 Attributed Abstracted B-rep (AAB) 개념을 제시한다. AAB는 함축적이며 단순화된 B-rep으로서 형상모델 데이터 양을 줄여 가벼운 클라이언트 프로세싱을 가능케 한다. 그리고, 특징형상 모델링 과정에 따른 서버상의 중립 특징형상 모델 관리 및 서버와 클라이언트간의 네트워크 지연을 최소화시키는 방법론에 대해서도 언급한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 시

스템 구조를 설명하고, 제 3장에서는 분산특징형상 모델링이 가능한 각 요소기술에 관해서 설명하고, 제 4장에서는 시스템 구현결과를 보여주며, 5장에서는 결론 및 추후 연구과제에 대해서 언급한다.

2. 시스템 구조

네트워크 기반 특징형상 모델링 시스템(NetFEATURE)의 구조는 Fig. 1에서 볼 수 있듯이 1) 중립 특징형상 모델 서버 (NetFEATURE Server), 2) 웹 기반 모델링 클라이언트(Web-based NetFEATURE Client), 3) CORBA 기반 표준 인터페이스로 구성되어 있다.

중립특징형상 모델링 서버는 분산환경 하에서 특징형상 모델링 뿐만 아니라 여러 어플리케이션에 필요한 공통된 모델링 서비스를 제공하는 역할을 한다. 이는 1) 특징형상 에이전트 관리자(Feature Agent Manager), 2) 중립 특징형상 모델 (Neutral Feature Model), 3) 솔리드 모델링 커널(Solid Modeling Kernel: ACIS)로 구성되어 있다. 에이전트 관리자는 분산 메타 오브젝트로서 서버에 접속한 모든 클라이언트를 관리한다. 각 에이전트는 접속된 각각의 클라이언트의 모든 요청에 따른 서비스를 제공한다. 중립특징형상 모델은 통합된 특징형상 모델링 기능을 제공하는 표현기법으로서, 솔리드 모델링 커널을 기반으로 구현되었으며, 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 여과하여 특정 영역에 적합한 모델을 구축할 수 있다.

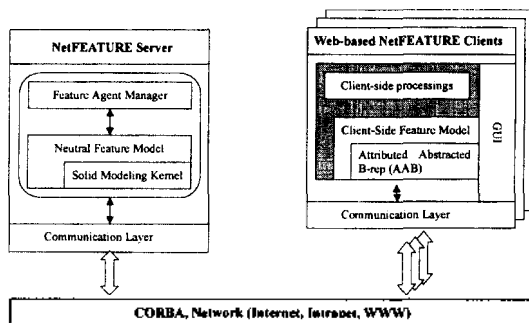


Fig. 1. System architecture for network-centric feature-based modeling

중립 특징형상 모델을 기반으로 한 특징형상 모델링, 가공특징형상 인식 및 변환과 같은 어플리케이션은 네트워크를 통해서 분산환경에서 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)과 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 모델링 시스템 개발에 관련된 모델을 의미한다. AAB는 기존의 B-rep정보를 단순화시킨 모델이며 다음의 목적을 위해서 이용된다: 1) 서버와 클라이언트 간의 고유이름 유지(Naming Consistency Maintenance), 2) 특징형상 모델링에 따른 서버와 커뮤니케이션 최소화, 3) 클라이언트 프로세싱을 위한 모델 제공. 따라서, 가벼운 클라이언트를 실현할 수 있으며, 궁극적으로 클라이언트 상에서 많은 프로세싱이 독립적으로 행해 질 수 있다.

CORBA 기반 표준 인터페이스는 분산환경 하에서 작업이 이루어 질 수 있도록 서버와 클라이언트의 교량역할을 하는 미들웨어다. 제시된 인터페이스는 CAM-I AIS 방법을 기반으로 두고 있다[1]. 따라서, 여러가지 솔리드 모델링이 가능하며, 특히 특징형상 모델링에서 필수적인 속성 정보를 저장할 수 있도록 확장되었다.

3. 네트워크 기반 특징형상 모델링

3.1 중립특징형상 모델

중립특징형상 모델은 형상모델링 커널을 기반으로 설계되었으며, 형상 요소들간의(특히, 클라이언트/서버 간) 일관성 있는 고유이름 부여기능을 포함하고 있다.

중립형상 모델이 제공하는 주요 기능은 1) 형상의 생성 및 삭제, 2) 위상 검색(Topological Queries), 3) 특성(Properties) 검색, 4) 수정 등이 있다. 생성 및 삭제 기능은 여러가지 형상요소(솔리드 프리미티브 및 형상 요소) 생성 및 삭제 할 수 있는 기능을 제공하며, 여기에는 한 가지 이상의 방법이 있을 수 있

다. 형상 요소가 생성되면 클라이언트와의 커뮤니케이션을 위하여 이를 서버의 매핑 테이블에 저장한다. 만약, 형상요소를 삭제하게 되면 형상요소는 자신뿐만 아니라 매핑 테이블에서 역시 함께 삭제된다. 위상 검색기능은 Parent 요소, 첫번째 Child 검색 등 위상에 관련된 여러가지 검색기능을 부여한다. 특성 검색 기능은 위상 및 형상(Geometry)의 검색을 가능케 한다. 예를 들면, 특정 면과 관련된 값의 검색(Query) 등의 기능이 이에 포함된다. 수정 기능은 여러가지 형상요소들과 관련된 프로세싱을 제공한다. 예를들면, 스위핑이나 블리언 연산 등이 여기에 해당된다. 이 모든 기능은 표준 인터페이스를 통해서 제공된다.

한편, 특징형상 모델링 및 그 응용과정에서 발생하는 특징형상 간의 간섭은 여러가지 문제를 야기할 수 있다. 특히, 특징형상이 추가되는 과정에서 특정면이 트리밍되거나(Trimming), 합병(Merging) 및 분할(Splitting)된다. 문제점으로는 파라메트릭 특징형상 모델링 과정 중 형상모델 수정과정에서 야기되는 위상 변화에 따른 형상 모델 수정 실패^{8,21}, 특징형상 간섭에 따른 가공특징형상 인식의 어려움^{11,22}, 클라이언트와 서버 간의 형상 요소들의 불일치^{4,17} 등이 있다.

Fig. 2는 각각 특징형상이 추가되었을 때 일어나는 특정면의 트리밍, 합병 및 분할 과정을 도시하고 있다. Fig. 2(a)는 Stock에 특징형상 F_1 을 추가했을 때 발생하는 트리밍 과정을 보여준다. 즉, f_1 과 f_2 가 f_1' 과 f_2' 로 트리밍되었으며, $f_3 \sim f_6$ 은 새롭게 생성된 면이다. Fig. 2(b)는 특징형상 F_1 의 f_1, f_2, f_3 가 특징형상 F_2 의 추가로 말미암아 분할된 과정을 도시하고 있다. 반면에, Fig. 2(c)는 특징형상 F_2 가 추가됨으로써 F_1 의 f_1, f_3 가 F_2 의 옆면(Side Face)과 합병되어 f_1', f_3' 로 변형된 모습을 보여준다.

이러한 특징형상간섭을 올바르게 처리하지 못할 경우 특징형상인식과 같은 추후 과정에서 많은 어려움이 발생한다. 따라서, 이를 효과적으로 처리하기 위해서는 무엇보다도 특정면들의 트리밍, 합병, 분할 과정을 효과적으로 관리할 수 있는 방법론 개발이 필수적이다.

본 논문에서는 특징형상 간섭에 의해서 야기되는 위상학적 변화를 효과적으로 처리하기 위해서 고유이름 관리기법(Generic Naming)을 이용한다. 고유이름 관리기법은 특징형상의 면들이 어떻게 생성되었는지, 어떻게 분할, 변형, 합병되었는지, 또는 삭제되었는지에 관한 모든 정보를 관리한다. 관리하는

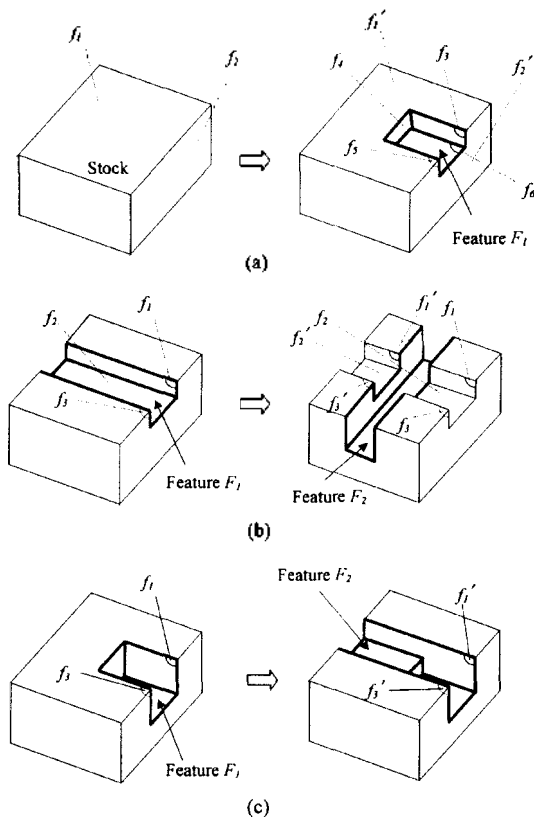


Fig. 2. Trimming, splitting and merging of faces.

FaceIdNodes로 구성되어 있는 FaceIdGraph에 의해서 이루어진다. FaceIdGraph는 방향성 비순환 그래프(Directed Acyclic Graph)이며, 입력 모서리(Edge)는 주어진 면의 Parent에 관한 정보를 출력 모서리는 어떠한 위상학적 변화가 일어났는지에 관한 정보를 나타낸다⁸. Fig. 3는 특징형상 모델링 과정에서 발생하는 면의 위상학적 변화를 관리하는 FaceIdGraph를 보여주고 있다.

FaceIdNode는 B-rep의 속성값(Attribute)형태로 존재하며 다음의 추가적인 정보를 포함하고 있다.

- 1) FaceId 정보
- 2) Real or virtual tag: FaceIdNode가 최종 형상의 특정면에 해당되면 Real, 그렇지 않으면 Virtual Tag을 저장함(Virtual Tag의 의미는 FaceIdNode에 해당되는 면이 특징형상 추가과정에서 삭제되었음을 의미함)
- 3) FaceId에 해당되는 B-rep의 면
- 4) 면이 속한 특징형상 리스트(만약 간섭이 일어날 경우 하나 이상의 특징형상을 포함할 수 있음)

위에서 정의된 FaceId는 각 면의 고유이름 (Unique FaceId)을 정의한다. 면 f 의 FaceId는 다음과 같이 정의된다:

$$FaceId(f) = [stepId, faceIndex, surfaceType]$$

여기서 $stepId$ 는 f 가 생성될 때의 모델링 순서, $faceIndex$ 는 f 의 인덱스, 그리고 $surfaceType$ 는 면의 타입을 나타낸다.

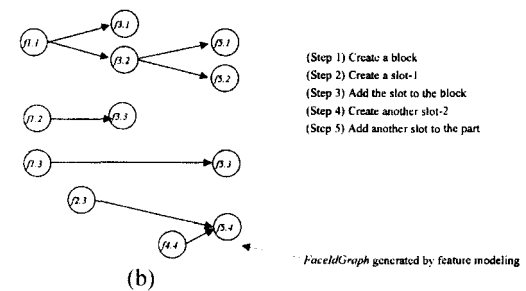
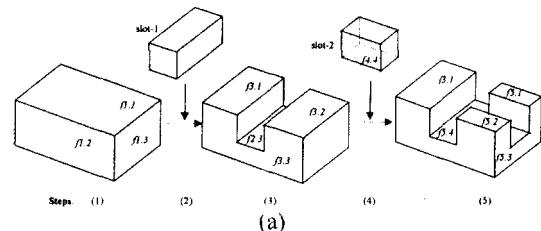
모서리(Edge)는 둘 또는 그 이상의 면이 교차해서 만들어내는 교선에 의해서 정의된다. 따라서 모서리 e 의 EdgeId는 그 모서리를 에워싸는 면들에 의해서 정의될 수 있다:

$$EdgeId(e) = [adjFaceIds, endFaceIds_{0,1}]$$

여기서 $adjFaceIds$ 는 모서리 e 를 에워싸는 면들의 리스트(Ordered Cyclic List), $endFaceIds_{0,1}$ 는 모서리의 각 점(Vertex)과 관련된 면들의 리스트를 의미한다.

결론적으로, 고유이름 관리기법은 분산 특징형상 모델링에서 다음과 같은 중요한 기능을 수행한다.

- 1) 서버와 클라이언트의 형상요소들간의 Naming Consistency 유지 및 Thin 클라이언트 달성
- 2) 분산 특징형상 모델링 과정에서 발생하는 서버와 클라이언트 간의 네트워크 부하 최소화
- 3) 파라메트릭 특징형상 모델링에서 수정에 따른 올바른 정보 매핑^{8,10,21}



Note: In this example we use notation fm,n denoting a face created during StepId m and with index n at the step m .

Fig. 3. Generic naming scheme.

4) 특징형상 간섭 확인^{11,22}
 본 논문에서는 1), 2)항에 중점을 둔다.

3.2 표준 인터페이스

3.2.1 CORBA IDL 인터페이스

서버와 클라이언트 간의 인터페이스는 CORBA IDL에 의해서 정의된다. Fig. 4은 본 연구에서 사용되는 CORBA IDL의 일부를 보여주고 있다. AgentManager는 서버에 접속된 모든 클라이언트를 관리한다. 따라서, 여기에는 Check In/Out, 에이전트 생성/삭제 등의 기능이 있다. 또한, FAgent는 접속된 각 클라이언트에 분산 특징형상 모델링이 가능하도록 여러가지 서비스를 제공한다.

3.2.2 고유이름 일관성 유지(Naming Consistency)

분산 모델링 환경 하에서는 서버와 클라이언트 모두 각자 필요한 정보를 지니고 있다. 따라서, 이들간의 원활한 커뮤니케이션을 하기 위해서는 각 형상 요소는 서버 및 클라이언트 상에서 공통된 고유한 이름을 지녀야 한다.

서버와 클라이언트 간의 커뮤니케이션을 위한 고유이름은 앞 장에서 설명된 고유이름 관리기법에서 정의된 구조를 따른다. 예를 들면, 면(Face)은 Fig. 5

```

module NetFEATURE {
    /* declarations */
    .
    .
    .
    interface AgentManager {
        FAgent CreateAgent(in string name);
        long CheckIn(in string userid,
                    in string passwd);
        long CheckOut(in FAgent agent);
        void DeleteAgentByObject(
            in FAgent agent);
        .
        .
    };

    interface FAgent {
        - Creation/deletion
        - Geometric/topological queries
        - Properties
        - Modifications
        - Analysis
        - Attribute-based mechanism
        - Visualization data access (AAB)
        - Transferring To/From Disk
        .
        .
    };
    .
};
    
```

Fig. 4. CORBA IDL interface

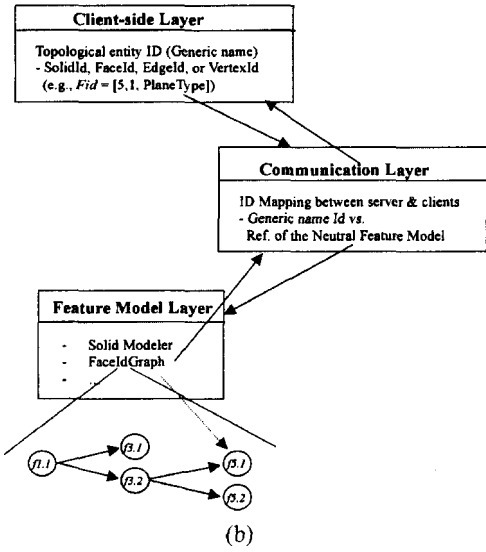
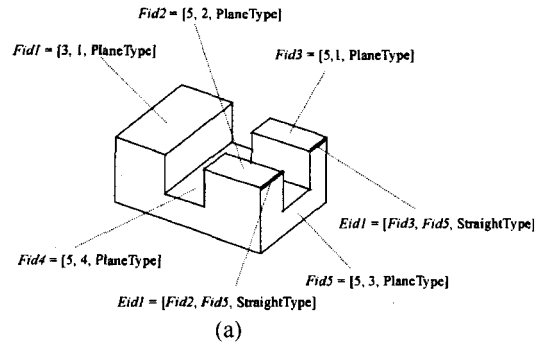


Fig. 5. ID mapping between server and client.

에서 볼 수 있듯이 정수와 상수의 벡터로 정의된다. 따라서, 클라이언트 상에서는 형상 요소의 고유이름을 벡터 형태로 지니고 있으며, 필요할 경우 이 이름을 표준 인터페이스를 통하여 전송 및 수신과정을 통하여 서버와 커뮤니케이션 할 수 있다. 서버에서는 입력된 이름 벡터를 FaceIdGraph상에서 매칭과정을 통하여 원래의 형상 요소를 찾을 수 있다(Fig. 5 참조). 모서리도 유사한 형태로 커뮤니케이션이 이루어진다.

3.3 클라이언트 프로세싱

웹 또는 네트워크 환경하에서 이루어지는 분산 모델링과 관련된 중요한 이슈중의 하나는 클라이언트 프로세싱이다¹⁹. 즉, 서버가 모든 프로세싱을 행하는 것이 아니라 클라이언트가 프로세싱의 상당부분을 독립적으로 처리할 수 있어야 한다는 의미이다. 즉, 클라이언트는 사용자가 특징형상 모델링을 수행할

수 있도록 여러 기능을 제공해야 한다. 주된 기능으로는 각 영역에 적합한 형상 모델 표현, 실시간의 디스플레이 및 네비게이션, 모델링을 위한 사용자 인터페이스 그리고 여러가지 기하학적 프로세싱 등이 있다. 특히, 서버와 클라이언트간의 네트워크 지연을 최소화시키기 위하여 서버와 클라이언트 간의 프로세싱 균형을 어떻게 유지시키느냐가 중요한 관건중의 하나이다.

3.3.1 Attributed Abstracted B-rep(AAB)

본 연구에서는 이를 위하여 Attributed Abstracted B-rep(AAB)을 클라이언트 형상 모델로 정의한다. AAB는 함축적이며 단순화된 B-rep으로서 형상모델 데이터 양을 줄여 가벼운 클라이언트 프로세싱을 가능케 한다.

AAB에는 두 종류가 있다: 1) Face-based AAB와 2) Edge-based AAB(Fig. 6. 참조). Face-based AAB는 FacetedFaces로 구성되어 있으며, 이는 형상의 곡면 표현(Surface Representation)을 근사화 한 것이다. 반면에, Edge-based AAB는 FacetedEdges로 구성되어 있으며, 이는 형상의 와이어프레임 표현(Wireframe Representation)을 근사화 한 것이다.

FacetedFace는 고유이름인 FaceId, 면의 방정식인 FaceEq, 그리고 IndexedTriangleList를 포함하고 있다(Fig. 6, 7 참조). 특히, FaceId는 서버상의 특징형상 모델과 클라이언트 모델 사이의 일관성 유지에 이용된다. 따라서, Face-based AAB를 이용하게 되면, 실시간 디스플레이 및 네비게이션, 크기·부피 정보(Mass Property) 추출, 레이 테스트(Ray Test)

```
<Attributed Abstracted B-rep (AAB)> ::= <Face-based AAB> | <Edge-based AAB>
<Face-based AAB> ::= { <FacetedFace> {, <FacetedFace>}
<FacetedFace> ::= { <FaceId>, <FaceEq>, <VertexList>, <IndexedTriangleList> }
<FaceId> ::= { stepId, faceIndex, <surfaceType> }
<surfaceType> ::= PlaneType | CylinderType | SphereType | TorusType | SplineType
<FaceEq> ::= <PlaneEq> | <CylinderEq> | <SphereEq> | <TorusEq> | <SplineEq>
<VertexList> ::= { <Vertex> {, <Vertex>}
<Vertex> ::= { <Point>, <Normal> }
<Point>, <Normal> ::= {float, float, float}
<IndexedTriangleList> ::= { <IndexedTriangle> {, <IndexedTriangle>}
<IndexedTriangle> ::= {int, int, int}
<Edge-based AAB> ::= { <FacetedEdge> {, <FacetedEdge>}
<FacetedEdge> ::= { <EdgeId>, <EdgeEq>, <PointList> }
<EdgeId> ::= { <adjFaceIdList>, <edgeType> }
<adjFaceIdList> ::= { <FaceId> {, <FaceId>}
<edgeType> ::= LineType | EllipseType | SplineType
<EdgeEq> ::= <LineEq> | <EllipseEq> | <SplineEq>
<PointList> ::= { <Point> {, <Point>}}
```

Fig. 6. Attributed Abstracted B-rep.

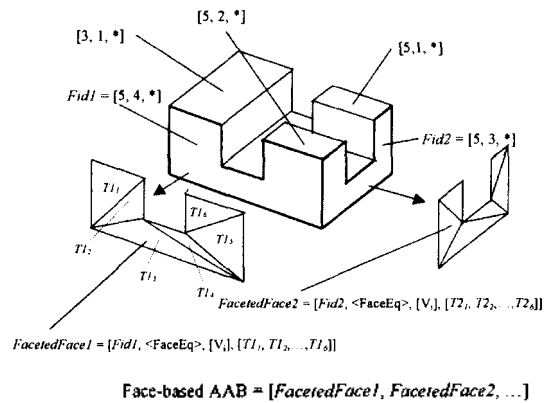


Fig. 7. Face-based AAB.

등의 형상 검색을 행할 수 있다.

3.3.2. AAB의 동적 갱신(Dynamic Update)

분산환경 하에서 특징형상 모델링을 수행하면 클라이언트는 서버로부터 형상이 수정될 때 마다 AAB를 전송받아야 한다. 만약, 형상모델이 변경될 때 마다 모든 AAB정보를 받아야 할 경우, 형상 모델이 복잡하면 클라이언트와 서버간의 네트워크 지연으로 말미암아 모델링이 불가능하게 된다. 따라서, 본 연구에서는 앞에서 제시한 고유이름 관리기법과 AAB를 이용하여 새로운 특징형상이 추가되었을 경우 수정되어야 할 AAB만을 선택적으로 전송받아 네트워크 지연을 최소화한다.

Fig. 8은 고유이름 관리기법이 어떻게 AAB의 국부적인(Local) 갱신을 어떻게 지원하는지를 보여준다. D_1 이 초기 특징형상으로 설계되었을 때, 모든 FacetedFace $f_{1,1} \sim f_{1,n}$ 은 클라이언트로 전송된다. 하지만, 그 다음부터 추가되는 특징형상부터는 국부적인 갱신이 일어나는 것을 볼 수 있다. D_2 가 추가되었을 때 $f_{1,3}$ 과 $f_{1,7}$ 은 FaceIdGraph상에서 트리밍되어 $f_{2,1}$ 과 $f_{2,2}$ 로 각각 변화되었다. 그리고, $f_{2,1}$ 은 FaceIdGraph에 새롭게 등록되었다. 따라서, 아래의 면들이 생성되고, 삭제되고, 갱신된다.

$$\begin{aligned} \text{NewFaceIds} &= \{f_{2,1}\} \\ \text{DeletedFaceIds} &= \{f_{1,3}, f_{1,7}\} \\ \text{UpdatedFaceIds} &= \{f_{1,1}, f_{1,2}\} \end{aligned}$$

하지만, 관련이 없는 다른 면들은 변화가 없다. 유사하게 D_3 가 추가되었을 경우, $f_{1,5}$ 는 $f_{1,2}$ 와 $f_{1,3}$ 으로 분할되고, $f_{1,4}$ 와 $f_{1,6}$ 은 각각 $f_{3,1}$ 과 $f_{3,4}$ 로 트리밍되었다. 결과적으로 아래의 AAB 갱신이 이루어진다.

$NewFaceIds = \{f_{4.1}, f_{4.2}, f_{4.3}\}$
 $DeletedFaceIds = \{f_{1.4}, f_{1.5}, f_{1.6}\}$
 $UpdatedFaceIds = \{f_{5.1}, f_{5.2}, f_{5.3}, f_{5.4}\}$

위의 예는 제시된 접근방법이 특징형상 모델링 과정에 따라 서버와의 커뮤니케이션을 어떻게 최소화시키는 지에 관한 사실을 잘 보여주고 있다. 이러한 사실은 현재까지 대부분의 연구 결과들이 다루지 못했던 분야이며, 앞으로 보다 많은 연구 결과들이 나오리라 예상된다.

4. 시스템 구현

제시된 방법은 네트워크 기반 가상모형 시스템 (Network-centric Virtual Prototyping; NetVP)의 주요 모듈중의 하나인 네트워크 기반 특징형상 모델링 시스템(Network-centric Feature-based Modeling; NetFEATURE)으로 구현되었다. Fig. 9에서 볼 수 있듯이 주요 시스템 모듈은 NetFEATURE 서버, 웹

기반 NetFEATURE 클라이언트, 세션관리기(Session Manager), 데이터베이스 서버, 그리고 이들을 연결해주는 CORBA기반 커뮤니케이션 인터페이스로 구성되어 있다.

서버는 C++로 구현되었으며, ACIS™를 기반으로 하고 있다. 클라이언트는 Java로 구현되었으며 렌더링은 Java3D로 구현되었다. 클라이언트와 서버의 CORBA 기반 표준 인터페이스로 Visibroker가 사용되었다.

4.1 특징형상 모델링 과정

웹기반 특징형상 모델링은 다음 과정을 따른다 (Fig. 9 참조). 우선 사용자가 Web 서버에 접속하여 Java Applet을 다운로드 받는다(1). 이때 Applet상의 클라이언트는 분산 모델링 서버에 접속하여 서버의 FAgent와 연결한다(2). 연결이 이루어지면, FAgent는 데이터베이스 서버와 연결이 되며, 이에 따라 데이터베이스 상의 검색 및 트랜잭션이 가능하게 된다(3).

우선 사용자는 NetFEATURE 클라이언트의 특징형상 라이브러리로부터 미리 지정된 특징형상(Pre-

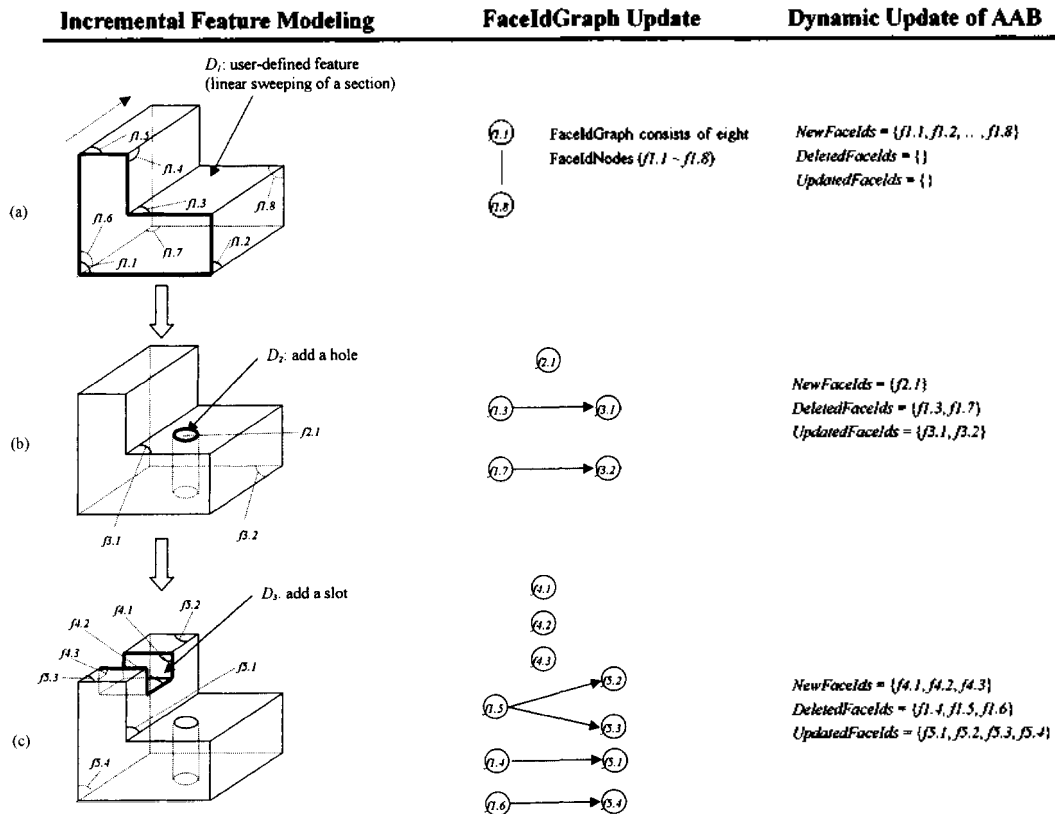


Fig. 8. Local update of AAB.

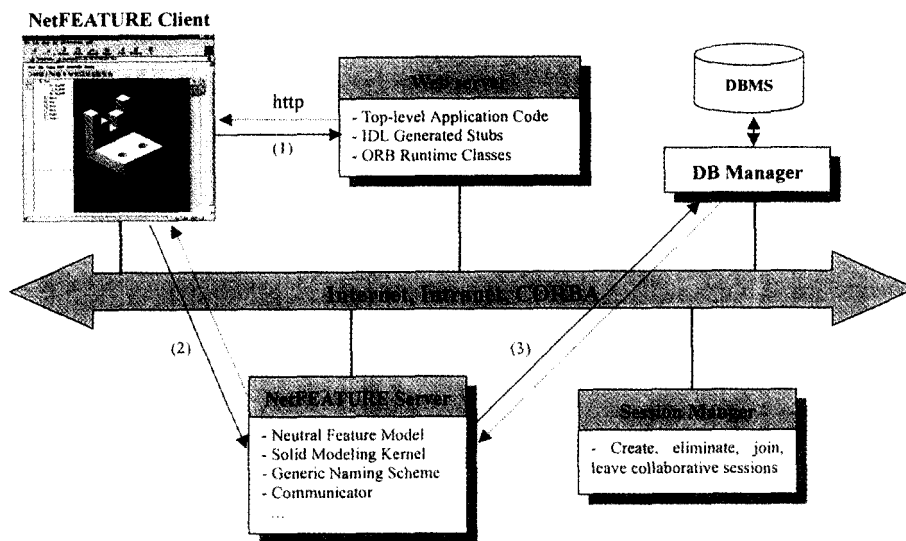


Fig. 9. System implementation.

defined Features)을 선택하여 생성시키거나, 스케치를 통해서 사용자 정의 특징형상(User-defined Features)을 만들 수 있다. 특징형상 라이브러리는 슬롯, 스텝, 홀, 포켓 등 일반적으로 통용되는 특징형상에 관한 정보를 객체지향 모델로 저장하고 있다. 따라서, 사용자는 각 특징형상에 적합한 파라미터 리스트, 위치 정보, 기하학적 제약조건 그리고 다른 속성값을 지정함으로써 각 특징형상을 정의할 수 있다. 입력된 정보는 표준 인터페이스를 통하여 NetFEATURE 서버에 전송이 된다. 전송된 정보에 아무런 오류가 없으면 서버는 입력된 특징형상의 중립특징형상모델을 생성시키고 사용자 정의에 의한 여러가지 프로세싱을 거친다. 최종적으로 중립특징형상모델의 AAB가 NetFEATURE 클라이언트로 전송된다. 전송을 받은 클라이언트는 이를 바탕으로 형상의 가시화 및 여러가지 프로세싱을 처리할 수 있다. 원하는 최종형상을 설계할 때까지 위의 모델링 과정을 반복한다.

4.2 구현 예

Fig. 9와 Fig. 10에 도시된 그림은 Web 브라우저 상에서 설계된 형상을 보여준다. 이 형상은 초기 특징형상(Basic Stock Feature), 측면 리브(Lateral Rib), 두개의 사각 슬롯(Rectangular Slot), 그리고 두개의 홀(Hole)로 구성되어 있다. 측면 리브는 초기형상의 윗면에 설계되었으며, Slot1은 리브의 윗면에 추가되었고, Slot2는 Slot1의 아래면에 추가되었다. 마지막으로 두개의 홀은 초기형상의 윗면에 설계되

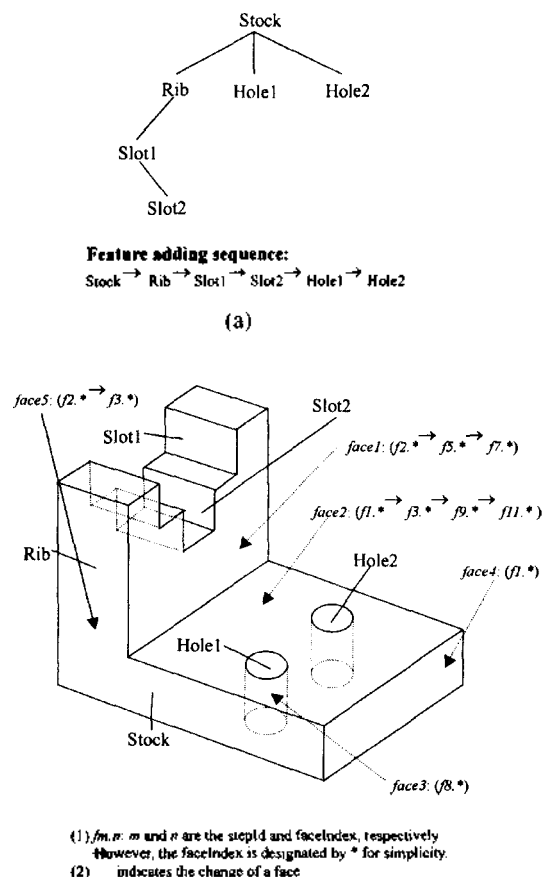


Fig. 10. An example part in Fig. 9 and related information: (a) design feature tree, (b) features and their AAB.

었다. 또한, 특징형상들 간의 Parent-child 관계도 설정된다(Fig. 10(a)).

Fig. 10(b)는 Fig. 10(a)의 특징형상 모델링 과정에 의해서 생성된 AAB결과를 보여준다. 특징형상이 추가되는 과정에서 발생하는 특징면의 트리밍, 합병, 분할로 야기되는 AAB의 전송과정을 보여주고 있다. 예를 들면, 초기형상의 윗면(face2)은 다음과 같은 변화를 거친다:

- 1) 우선, $f1.*$ 을 초기형상의 윗면이라고 하자.
- 2) 리브의 추가: $f1.*$ 은 $f3.*$ 으로 트리밍.
- 3) Slot1과 Slot2의 추가: $f3.*$ 은 변화가 없음.
- 4) 첫번째 홀의 추가: $f3.*$ 은 $f9.*$ 로 트리밍.
- 5) 두번째 홀의 추가: $f9.*$ 는 $f11.*$ 로 트리밍.

유사하게, 리브의 옆면(face1)은 Slot1과 Slot2가 추가될 때 다음과 같이 변경된다($f2.* \rightarrow f5.* \rightarrow f7.*$) 반면에, face3과 face4는 아무런 변화가 없다. 이 예는 NetFEATURE 클라이언트 상에서 특징형상이 어떻게 설계되며, 네트워크 부하를 최소화시키기 위해서 서버와 어떻게 커뮤니케이션하는 지를 잘 설명하고 있다. 여기에는 FaceIdGraph에 기반을 둔 중립특징형상의 역할이 매우 중요하다.

Fig. 11은 현재 연구가 진행중인 가상모형개발 시스템(NetVP)의 구현결과를 보여주고 있다. NetVP는 STEP AP203에 기반을 둔 조립체 모델 가시화 및 분석도구이다. Fig. 11에 도시된 모델은 상업용 CAD 시스템인 SolidWorks™에서 설계된 드릴이다. NetVP는 STEP파일을 읽은 후, 우선 조립체 구조를 찾고 각 부품을 찾아서 중립특징형상 모델로 변환시킨다. 클라이언트상에서는 조립체 구조에 기반을 둔 각 컴포넌트의 AAB를 서버로부터 전송받는다.

Fig. 11은 조립체 모델의 가시화 뿐만 아니라 클라이언트 프로세싱의 좋은 예를 보여준다. 즉, 조립체 모델의 간섭확인 과정을 단번절단을 통하여 확인할 수 있다. AAB는 클라이언트 상에서 원하는 공차(Tolerance)를 만족시키는 근사화 된 모델이므로 클라이언트 상에서 간섭확인이 가능하다. 뿐만 아니라, 사용자가 정확한 간섭유무를 확인하고자 할 경우에는 서버에 요청하여 조립체 컴포넌트 간의 불리언 연산을 통하여 결과를 얻을 수 있다. 따라서, 클라이언트는 독립적인 객체로 존재할 수 있을 뿐 아니라 서버에 종속적인 객체로 존재할 수도 있다.

4.3 Collaborative session manager

Fig. 9에서 볼 수 있듯이 Collaborative Session

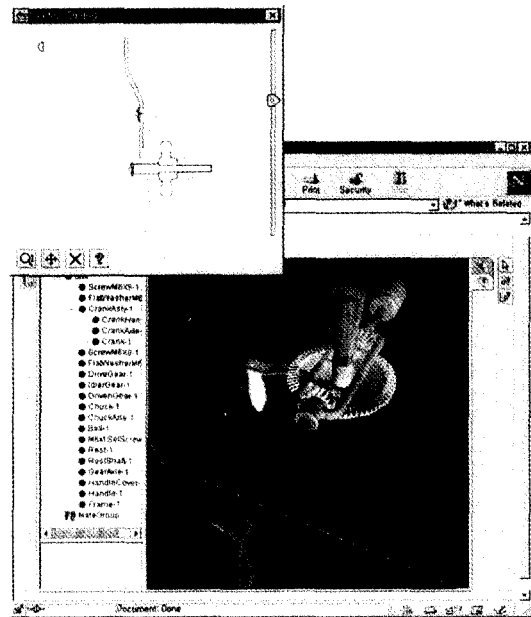


Fig. 11. STEP-based virtual prototyping: visualization and client-side processing.

Manager는 분산된 환경에서 여러 작업자들 간의 상호 협동 작업을 지원하기 위한 서버 프로그램으로써, 현재 수행 중인 모델링의 내용을 일치시키기 위한 동기성(Synchronization)과 작업자의 참여를 유도하는 비동시성(Non-simultaneity)의 특성을 함께 제공한다. Fig. 12는 Collaborative Session Manager의 구조를 보여준다. 본 논문에서는 세션 중에 대상 모델을 공유하고 작업자들 간의 커뮤니케이션을 지원하기 위해 CORBA 서비스 중의 하나인 이벤트 서비스를 이용하였다. 동기성을 위한 프로세스는 Session Server가 담당하며 비동시성을 위한 프로세스는 Notice Server가 담당한다.

동기성은 협동작업에서 모든 참여자들이 동일한 작업 내용을 수행할 수 있도록 보장하는 것으로써 이에 저촉되는 요인은 두 가지로 나눌 수 있다. 첫째는 Collaborative Session Manager 상의 작업자들이 서버와 혹은 작업자들 간의 원거리 작업에서 비롯된다. 이는 전송 거리나 네트워크 상태에 의해 통신에 지연이 생길 수 있음을 뜻하며 이로 인해 작업자들이 보는 작업내용이 서로 다를 수 있다. 본 논문에서는 참여자의 입력을 일단 서버에 전송하고 서버가 이의 결과를 각 참여자에게 전송하는 방식을 통해 해결하도록 하였다. 결국 모든 참여자는 서버가 수신한 순서와 동일한 순서로 작업의 내용을 수신하게

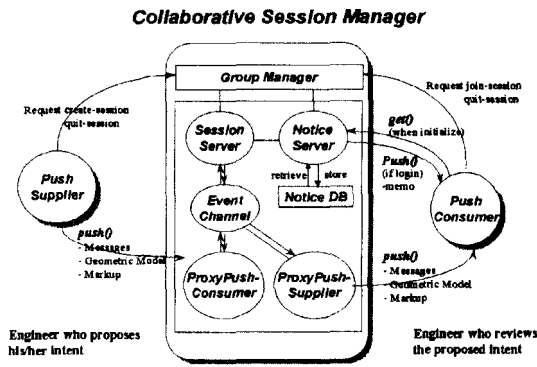


Fig. 12. Structure of collaborative session manager.

된다. 둘째 요인은 여러 작업자가 동시에 동일한 작업 대상에 작업을 함으로써 비롯된다. 이는 한 순간에 작업 대상에 변화를 가할 수 있는 작업자를 한 명으로 제한함으로써 해결할 수 있는데 본 논문에서는 서버에 작업 권한을 두고 이를 먼저 획득한 작업자가 유일하게 작업을 수행하도록 함으로써 작업 내용의 불일치를 방지할 수 있도록 하였다.

비동시성은 상호 작업에 요구되는 참여자가 네트워크 상의 문제 등 뜻하지 않은 이유로 인해 불참할 수 있음을 인정하고 참여를 유보시키는 것으로서, 본 논문에서는 통보(Notification)와 수락(Acceptation)의 형태로 이를 실현하였다. 통보는 참여 대상자에게 Collaborative Session이 시작되었으며 Session에 참여할 것을 알리는 것이며, 수락은 통보를 받은 참여 대상자가 이 통보에 수락함으로써 협동작업에 참여하는 것이다. 통보와 수락은 참여자의 참여와 협동작업의 수행 자체를 독립적으로 관리함으로써 작업의 지연을 없앨 수 있다. 또한 작업자가 자신이 해당된 작업 그룹을 찾을 필요 없이 통보의 수락과 함께 작업에 참여할 수 있도록 함으로써, 현재 Session에 참가할 수 없는 작업자의 경우 시스템의 접속과 함께 통보 메시지를 받음으로써 작업요청이 있었음을 알릴 수 있도록 하였다.

5. 결 론

본 논문은 네트워크 기반 특징형상 모델링 기법을 제시하였다. 클라이언트/서버 구조 하에서 이루어지는 모델링 기법으로서, 특히, 중립특징형상 모델과 클라이언트/서버간의 커뮤니케이션, 클라이언트 프로세싱 등에 관한 방법론을 제시하였다. 특히, 형상모델의 단순한 가시화 기능뿐만 아니라 특징형상모델

링 기능이 가능하도록 설계되었다. 본 논문의 장점을 정리하면 다음과 같다.

- 1) 중립특징형상 모델 서버는 분산 모델링을 가능케 하는 서비스 제공자이며, 고유이름 관리기법을 제공한다. 따라서, 특징형상간섭해결, 클라이언트/서버의 형상모델 일치성 등을 가능케 한다.
- 2) AAB는 B-rep을 단순화시키고 추상화 시킨 모델로서 가벼운 클라이언트를 실현시킬 수 있다. 뿐만 아니라 고유이름 관리기법의 근간이 되는 FaceIDGraph와 함께 특징형상 모델링에 따른 네트워크 딜레이를 최소화시킬 수 있다.
- 3) NetFEATURE는 Java에 의해서 프로그램되어 플랫폼에 독립적이다. 단순히 Web 브라우저에 모든 작업이 가능하며, 소프트웨어 설치 및 업그레이드가 불필요하다.

추후연구과제는 다음과 같다.

- 1) 다중 서버/ 다중 클라이언트 구조
- 2) 파라메트릭 모델링 접목
- 3) Web 기반 디지털 Mockup
- 4) 분산 조립체 모델링과 접목

참고문헌

1. CAM-I, Application Interface Specification AIS 2.0. Technical Report R-90-PM-03. Consortium for Advanced Manufacturing, 1990.
2. Gadh, R. and Sonthi, R., "Geometric shape abstractions for internet-based virtual prototyping", *Computer-Aided Design*, Vol. 30, No. 6, pp. 473-486, 1998.
3. Gupta, S. K., Automated manufacturability analysis of machined parts, *Ph.D. Thesis*, University of Maryland, 1994.
4. Han, J. H. and Requicha, A. A. G., "Modeler-independent feature recognition in a distributed environment", *Computer-Aided Design*, Vol. 30, No. 6, pp. 453-463, 1998.
5. Hardwick, M., Spooner, D. L., Rando, T., and Morris, K. C., "Sharing manufacturing information in virtual enterprises", *Communications of the ACM*, Vol. 39, No. 2, pp. 46-54, 1996.
6. Kim, H. and Chung, M. J., "A Web-based framework for engineering design process in a concurrent engineering environment", *Proc. Advanced Concurrent Engineering*, pp.417-423, 1998
7. Kim, S., Yang, U., and Kim, J., "COVRA-CAD: A CORBA-based distributed VR CAD Systems", *Int. Conference on Virtual Systems and Multimedia*, 1998.
8. Kripac, J., "A mechanism for persistently naming topo-

- logical entities in history-based parametric solid models", *Computer-Aided Design*, Vol. 29, No. 2, pp. 113-122, 1997.
9. Lee, J. Y., A knowledge-based approach to feature-based parametric modeling, Ph.D. Thesis, POSTECH, Korea, 1998.
 10. Lee, J. Y. and Kim, K., "A 2D geometric constraint solver using DOF-based graph reduction", *Computer-Aided Design*, Vol. 30, No. 11, pp. 883-996, 1998.
 11. Lee, J. Y. and Kim, K., "A feature-based approach to extracting machining features", *Computer-Aided Design*, Vol. 30, No. 13, pp. 1019-1035, 1998.
 12. Martino, T. D., Falcidieno, B., and Hasinger, 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.
 13. Phang, F., Senin, N., and Wallace, D., "Distribution modeling and evaluation of product design problems", *Computer-Aided Design*, Vol. 30, No. 6, pp. 411-423, 1998.
 14. Regli, W. C., "Internet-enabled computer-aided design", *IEEE Internet Computing*, pp. 39-50, 1997.
 15. Regli, W. C., Gupta, S. K., and Nau, D. S., "Towards multiprocessor feature recognition", *Computer-Aided Design*, Vol. 29, No. 1, pp. 37-51, 1997.
 16. Shah, J. J. and Mantyla, M., *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*, John Wiley & Sons Inc. New York, 1995.
 17. 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.
 18. Trika, S. N., Banerjee, P., and Kashyap, R. L., "Virtual reality interfaces for feature-based computer-aided design systems", *Computer-Aided Design*, Vol. 29, No. 8, pp. 565-574, 1997.
 19. Wang, F.-C. and Wright, P. K., "Web-based CAD tools for a networked manufacturing service", *Proceedings of DETC'98 ASME Design Engineering Technical Conference*, Atlanta, Georgia, CIE-5517, 1998.
 20. 최영, 양상욱, "STEPShare: 웹 환경에서의 CORBA 기반 3D STEP 브라우저", CAD/CAM 학회 학술

발표회 논문집, pp. 24-29, 1998.

21. 이재열, 김광수, "파라메트릭 접근방법에 의한 특징형상을 이용한 모델링", 한국 CAD/CAM 학회 논문집, 제 1권, 제 3호, pp. 242-256, 1996.
22. 이재열, 김광수, "특징형상 접근방법에 의한 가공특징형상 추출", 한국 CAD/CAM 학회 논문집, 제 4권, 제 2호, pp. 139-152, 1999.

이재열

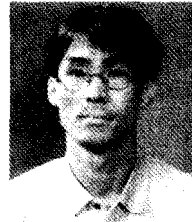


1992년 포항공과대학교 산업공학과 학사
1994년 포항공과대학교 산업공학과 석사
1998년 포항공과대학교 산업공학과 박사
1998년-현재 한국전자통신연구원(ETRI) 컴퓨터-소프트웨어 기술연구소 동시공학연구팀 선임연구원

관심분야: network-centric CAD, feature-based modeling, parametric design, virtual prototyping, assembly modeling

E-mail: jaelee@etri.re.kr

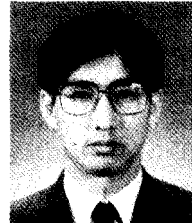
김 현



1984년 한양대학교 기계설계학과 학사
1987년 한양대학교 기계설계학과 석사
1997년 한양대학교 기계설계학과 박사과정
1993년-현재 한국전자통신연구원 동시공학연구팀 선임연구원

관심분야: Concurrent Engineering, Virtual Engineering, CAD/CAM/CAE/PDM

한성배



1977년 고려대학교 산업공학과 학사
1984년 고려대학교 산업공학과 석사
1996년 고려대학교 산업공학과 박사
1980년-현재 KIMM, SERI, ETRI 책임연구원, 동시공학연구팀장

관심분야: 동시공학, CALS
