

## WWW와 연계된 선박 조립 구조물의 분할 CAD 모듈 개발

최해진\*, 이수홍\*\*

### A Development of Ship-Block Dividing CAD Module Connected to WWW

Hae-Jin Choi\* and Soo-Hong Lee\*\*

#### ABSTRACT

A ship-block cutting module is being developed as part of a computational system for a ship design. The module supports product-information to other modules in design process and displays ship-blocks in cutting-operation. Welding-information is essential to process-planning and scheduling. In the module, ship-blocks are modeled by Plate-Objects and then divided into several cutting pieces. The module automatically creates Welding-Objects containing welding-information in the process. Since the module is connected to WWW (World Wide Web), users in various platforms can access the module simultaneously. In addition, users can check manufacturing constraints by inspecting a virtual model.

**Key words** : Block Cutting, Plate-Object, Welding-Object, Neighborhood-Classifer, Collaborative System

#### 1. 서 론

조선 산업은 제조되는 선박의 규모 만큼이나 다양한 기술이 복합적으로 응용되는 큰 산업 분야이다. 특히 조선의 설계 분야는 많은 특화된 기술들이 서로 나누어져 존재하기 때문에 그들 상호간의 의사 전달 및 데이터 호환은 효율적인 조선 시스템 설계를 위하여 가장 중요한 부분으로 자리잡고 있다. 이런 요구에 따라 일찍부터 조선 시스템 분야는 CIMS(Computer Integrated Manufacturing System)에 주력해 왔고, 많은 조선소들이 이를 개발 실용화하는데 관심을 가지고 있다.

그러나, 각 분야의 전산화는 통합을 고려하지 않고 각 부서의 특성에 맞게 개발된 시스템을 적용하고 있기 때문에, 설계/개발분야, 공정계획분야, 일정계획 분야의 시스템 통합을 위해서는 많은 문제점이 발생하게 된다.

현재까지 블록의 분할을 도와주는 전문가 시스템은 많이 개발되고 있지만, 이를 표현해 주고, 정보를 구조적으로 배열하며, 자동 추출할 수 있는 CAD지원 모듈은 개발되지 못하고 있는 실정이다. 본 연구에서 개발한 시스템은 기타 전문가 시스템(공정설계/일정계획 전문가 시스템)과의 동적 연결이 가능하도록 고안되었으며, CIMS구축을 위해 필수적인 개방성과 유연성을 확보하고자 웹(WWW)과의 연계도 시도하였다.

#### 2. 모듈 개발의 방향

설계에 따라 완성된 선체구조를 실제로 생산하는 단계에 선체를 여러 개의 블록으로 분할한다. 실제 생산시에는 설계측면에서의 부재의 치수들을 변화시킬 필요가 있고, 구매나 가공, 조립 가능한 부재의 크기는 제한이 있다. 따라서, 선체생산설계에 따라 부품생산을 위한 부품의 상세화가 진행된다<sup>1)</sup>.

블록분할 기능은 부품화 작업을 위한 분할은 물론, 부품이 생성되는 과정인 중간제품의 생성정보와

\*학생회원, 연세대학교 기계공학과 대학원

\*\*중신회원, 연세대학교 기계공학과

블록에 대한 생산정보의 취득도 가능하게 해야 한다. 것에 의한 선체구조의 분할은 접합관계 정보에 대해서도 2개로 나누어져야 한다.

본 연구에서는 제품모델을 기초로 하여, 선체 기본설계와 상세설계<sup>1)</sup>를 공통으로 수행하는 시스템을 구현한다. 이렇게 함으로써 선체 기본설계 단계의 정보로부터 선체 모델을 구축하고 선체 모델로부터 신속히 정보들을 추출할 수 있으며, 하류 설계단계로의 정보 일관화 및 신속한 정보산출이 가능하게 된다.

조선 시스템의 동시공학적 접근을 위한 중요한 관건은 정보의 공유로 보고, 이 문제의 해결책으로 웹(WWW)과의 연계를 시도하였다. 전문가셀에서 추출되는 제품모델 정보들을 웹호환 형식의 데이터로 전환하고, 그들 간의 링크를 통하여 구조적인 연관을 가지도록 하였다. 웹이라는 인프라(Infrastructure)를 이용하므로, 동시공학의 다자간 동시 정보교환을 이룰 수 있다.

### 3. 분할모듈 설계

분할모듈은 대상블록에 필요한 객체들을 생성하고 생성된 객체들간의 상관관계 및 분할 후의 블록간 계층구조변화를 추적하여 내부데이터구조의 일관성유지를 목적으로 설계되었다. 분할모듈설계의 핵심은 각 객체의 생성과 객체간의 상관관계, 즉 부재객체와의 정보속성전이를 어떻게 잘 활용할 수 있도록 스키마(Schema)를 정의하는데 달려있다. 다음절들에서는 이러한 객체정의와 분할 알고리즘에 관한 설명이다.

#### 3.1 객체의 정의

객체의 종류에는 3가지가 있다. 원점객체(Origin Object), 부재객체(Plate Object), 접합객체(Welding Object)가 그것이다<sup>1)</sup>.

원점객체는 모델링될 블록의 기준점을 정의한 객체이다. 이전 단계에서 정의된 블록단위의 정보를 가지고 있으며, 블록분할 모델링의 기준이 되는 점이다.

부재객체는 각각의 부재를 다각형을 기본으로 하는 Extrusion모델을 이용해 3차원 형상모델링을 하였으며, 내부 정보로서 부재의 중량, 크기, 재질, 위치 등을 가지도록 하였다. 그러나 선체구조는 판, 골 등이 접합되어 그 구조가 구성되어 있으므로, 형상과 내부정보만으로는 충분한 제품정보를 표현할 수 없다.

그러므로 본 연구에서는 부재객체 외에 접합관계를 하나의 물체로 생각하여 접합객체를 정의하였고, 이에 3차원 선분 형상을 부가하여 부재객체와 같이

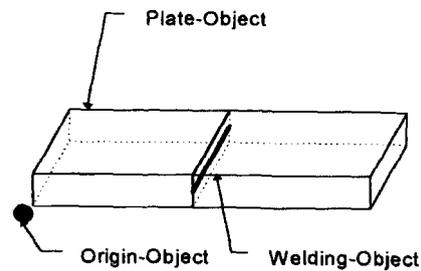


Fig. 1. An example of object definition.

모델링에 포함시켰으며, 내부정보로서 용접시 포함되는 공작정보를 갖도록 하였다. 이렇게 함으로써 부재객체와 유사하게 형상정보와, 내부정보를 모두 갖게 된다<sup>1)</sup>.

#### 3.2 객체의 생성 및 분할

##### 3.2.1 원점객체의 생성

블록이나 단위블록을 모델링하기 위한 초기화로서 지역좌표 원점객체를 생성한다. 원점객체를 기준으로 하여 연관된 다른 객체들을 생성한다. 원점객체는 이전 설계단계에서 입력된 블록의 내부정보를 가지도록 설계되었으며 해당 블록에 속하는 모든 부재/접합객체에 동일하게 존재하는 내부정보를 포함하도록 설계된다. 이러한 정보는 원점객체의 하위부품인 부재 및 접합객체의 내부정보로 자동 속성전이(Property Inheritance)되어 정보입력의 중복을 피할 수 있고 정보관리에 일관성을 유지할 수 있게된다.

##### 3.2.2 부재객체의 생성/추가

대상블럭의 설계도면을 전문가셀내의 CAD 시스템으로 전환하기 위한 방법은 여러가지 있을 수 있다. 사용중인 CAD시스템(예: AutoDef)과의 직속연결을 통한 방법이 있고 다른 하나는 중간파일(예: IGES, DXF, STEP 등)을 이용하는 방법이다. 전자는 사용중인CAD시스템의 원전파일까지 접근하여 수 정보완이 되면 가능하나 현실상 불가능하므로 본 연구에서는 후자를 택하였다. 가장 널리 쓰이는 IGES 파일을 이용하여 설계정보를 전문가셀내로 전환하였다. 전환된 와이어프레임 모델중 솔리드화 하는데 필요한 최소한의 요소들을 선택하여 부재객체의 솔리드 모델을 생성한다. Fig. 2는 와이어 프레임 모델이 솔리드 모델로 다시 생성되는 과정을 보여주고 있다.

예를 들어 모서리의 형상이 곡선을 가지고 있는 부재에 대해서는 원호를 일정 각도로 분할하여 직선으로 근사화시킨 후 필요한 형상으로 변환시킨다.

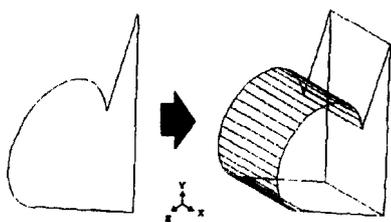


Fig. 2. An example of creating a solid-model from IGES wire-frame model.

앞서 언급한 원점객체를 중심으로 필요한 부재객체의 생성은 다른 파일의 데이터를 다시 읽어와 부재객체를 추가하거나, 같은 파일의 다른 부재의 와이어프레임 모델을 부재객체로 추가하여 생성한다. 이때에도 원점객체의 하위부품으로서 생성되어 추가되며, 이때 이미 존재하는 다른 부재객체와의 간섭을 점검하여 간섭이 발생하면, 그에 따른 접합객체를 생성한다.

이러한 인터페이스 방식은 기존의 설계흐름을 크게 바꾸지 않고도, 3차원 형상화가 가능하기 때문에 설계단계 어느 곳에든지 적용할 수 있는 시스템이 될 수 있다.

3.2.3 추가부재로 인한 접합객체의 생성

접합객체의 생성은 부재객체가 추가되는 유형에 따라 그에 맞는 접합객체 생성 알고리즘을 적용한다. 그 유형은, Fig. 3와 같이 크게 3가지로 나뉘어진다.

- (1) face-to-edge: 기존 부재의 면과 새로 추가되는 부재의 모서리가 만나 접합객체가 생성되는 경우.
- (2) edge-to-face: 기존 부재의 모서리와 새로 추가되는 부재의 면이 만나 접합객체가 생성되는 경우
- (3) face-to-through(through-to-face): 기존 부재의 면에 새로 추가되는 부재(또는 기존의 부재가 새로

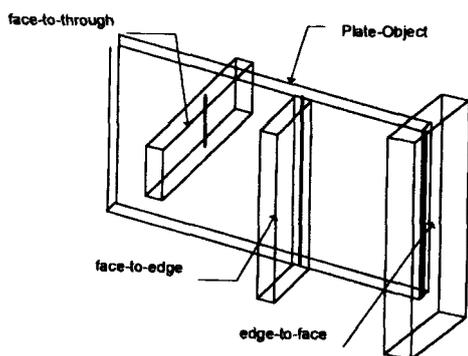


Fig. 3. An example of contact-relationship between Plate-Objects.

추가되는 부재의 면을)가 관통하여 접합객체가 생성되는 경우

3.2.4 부재/접합객체의 분할과 접합객체의 생성

본 연구에서는 지금까지의 조선 CAD시스템과는 다르게 형상정보를 이용하여 접합객체가 생성되므로, 분할평면으로 블록을 분할할 때 부재객체와 동시에 접합객체도 동일한 알고리즘을 이용하여 분할한다. 분할된 부재/접합객체들은 분할 전의 부재/접합객체의 하위부품으로 인식되기 때문에 분할전의 내부정보 및 특징정보를 속성전이하므로 형상정보만이 바뀌게 된다. 선체설계의 특징을 볼 때, 이러한 기능은 매우 효율적이다. 즉, 이전 설계단계에서 정의된 객체들의 속성을 다음 단계에서 재정의할 필요가 없기 때문이다.

예를 들어 Fig. 4의 왼쪽 그림과 같은 부재를 분할한다고 할 때의 기본 알고리즘은 다음과 같다.

- ① 각 선분이 차례로 분할면을 통과하는지 여부를 점검한다.
- ② 분할면을 가르지 않는다면 선분이 분할면의 방향벡터 방향에 위치해 있는지, 아니면 분할면의 방향벡터 반대 방향에 위치하는 지를 구분한다.
- ③ 선분이 분할면을 가르지 않고 분할면의 방향벡터 방향에 위치하면 선분에 양 끝점을 A저장소에 차례로 저장한다.
- ④ 선분이 분할면을 가르지 않고 분할면의 방향벡터 반대방향에 위치하면 선분의 양끝점을 B저장소에 차례로 저장한다.
- ⑤ 선분이 분할면을 가르다면, 선분과 분할면의 교점을 구하고, 선분의 시작점과 그 교점을 A, 교점 하나만을 B, 교점과 선분의 끝점을 C, 저장소에 각각 저장한다.
- ⑥ 차례대로 부재의 각 선분을 구별하여 저장한 다음, 각 저장소에서 중복된 점들을 한개만 남겨놓

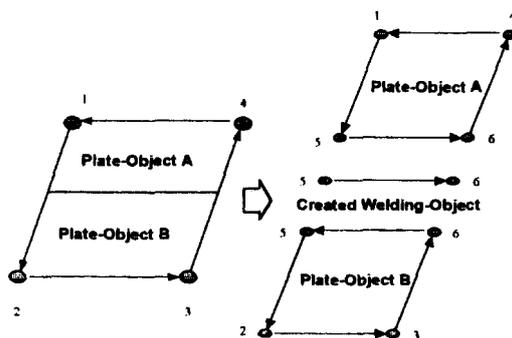


Fig. 4. An algorithm of cutting and creating objects.

고 삭제한다.

같은 방식으로 집합객체도 분할이 가능하다. 단, 집합객체가 분할될 때에는 부재객체가 분할될 때와는 달리 따로 생성되는 집합객체 없이 기존의 집합객체가 분할되기만 한다. 시스템의 실제 블록의 분할에 있어서도 분할면을 통과하는 모든 부재객체와 집합객체는 이러한 알고리즘을 적용하여 각각 순차적으로 분할하게 된다.

3.2.5 인접구분자(Neighborhood-Classifer)를 이용한 선분 재분류

선체를 구성하는 부재의 형상은 다양한 형태를 가지고 있다. 이러한 다양한 형태의 부재를 정상적으로 분할하기 위해서는 부재를 이루는 꼭지점을 잇는 선분을 상황에 따라 적절히 분류해야만 한다. 3.2.4절의 방법으로는, 기본적인 모양의 부재분할은 가능할 지 모르나, 선체구조중 Web과 같은 부분, 즉 접평면 분할(Coplanar Section)이 발생하는 부분과, 부재의 꼭지점을 포함하는 평면 분할은 좀 더 세밀한 선분의 재분류가 필요하다.

이러한 점을 보장하기 위하여 Fig. 6과 같이 인접구분자(Neighborhood Classifier)<sup>[5]</sup>를 적용하여, 선분과 꼭지점을 재분류한다.

Fig. 5과 같은 부재를 접평면으로 분할하기 위한 알고리즘의 기본 개념은 3.2.4절에서 설명한 것과 동일하지만, 선분분류에 따라 프로세스를 다양화하였다.

먼저, 3.2.4절에서는 저장소를 3곳으로 분류했지만, 이번에서는 4곳으로 정한다. 편의상, 첫번째 저장소를 A, 두번째 저장소를 B, 세번째 저장소를 C, 그리고 네번째 저장소를 D라 하자. 여기서 새로 추가된 네번째 저장소에는 ON에 해당하는 꼭지점 정보가 들어가게 된다.

차례로 부재의 모서리를 인접구분자에 의해 분류한다. 인접구분자에 따라 그에 해당하는 프로세스를 행하게 되는데, 인접구분자가 ABOVE-ABOVE, BELOW-BELOW, THROUGH일 경우에는 3.2.4절에서

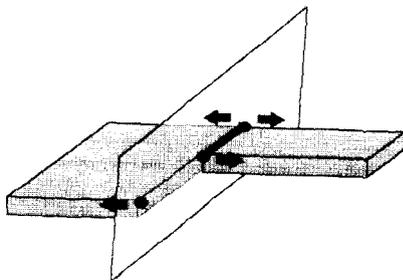


Fig. 5. An example of coplanar section.

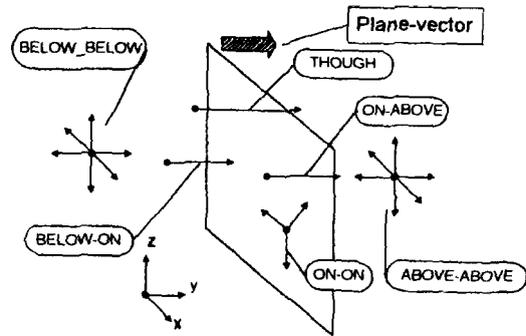


Fig. 6. Reclassification of all lines by Neighborhood-Classifer.

설명한 것과 동일한 프로세스를 수행하게 되고, ON-ON일 경우는 프로세스를 수행하지 않는다. ABOVE-ON, ON-ABOVE, BELOW-ON, ON-BELOW일 경우는 인접점이 저장되는 장소에 ON위치의 점을 저장하고, 다시 그 ON위치의 점을 저장소 D에 저장한다. 예를 들어 ON-ABOVE일 경우, ON 위치의 점을 저장소 A와 D에 저장하고, ABOVE에 해당하는 점을 저장소 A에 저장한다.

본 절에서 언급한 선분 재분류 방법은 기존의 Boolean Operator에 사용되었던 방법과 유사하나 선박블록의 특성을 고려하여 기존 알고리즘을 보완한 알고리즘이라 할 수 있다. 왜냐하면 면에 직접 접하는 경우이외의 경우(예: ABOVE-ABOVE, BELOW-BELOW, THROUGH 등)까지도 포함하기 때문이다.

3.2.6 데이터 정리 프로세스

이러한 프로세스를 수행하여 얻은 결과는 부재객체와 용접객체를 생성하는 데 적합한 데이터 구조가 아니다. 즉 저장소 B와 저장소 D에 저장되어 있는 각각의 데이터를 이용하여 다시 용접객체에 대한 정보를 생성해야만 한다.

알고리즘의 구조는 Fig. 7과 같다. 저장소 D의 꼭지점중 집합객체의 정보에 해당하는 것을 저장소 B로 옮긴 후 저장소 D를 소거하는 방식이다. B저장소의 크기는 절대 두개 이상이 될 수 없다. 그 이유는 알고리즘 적용의 전제 조건이 2개의 부재로 분할하는 경우에만 한정하기 때문이다.

저장소 B의 크기를 각각 NIL, 1, 2 세가지 경우로 나누어 평가한다. 저장소 B의 크기가 NIL의 경우, 저장소 A또는 C가 0이면 분할되지 않는다는 의미이므로, 분할 시에 저장소 D의 크기는 반드시 2가 되어야 하며, 1이 될 경우는 기하학적 조건상 발생할 수 없고, 2를 초과할 경우는 분할이 발생하지 않는 경우이다.

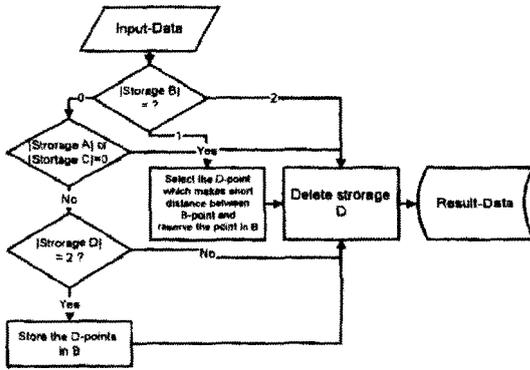


Fig. 7. Refining process for Welding-Object's vertex-data.

### 3.3 제품정보의 계층구조

조립체를 구성하기 위하여 부재객체를 생성하면, 시스템은 부재명 입력을 기다린다. 부재명이 입력되면, 이 명칭을 가진 인스턴스가 생성되며, 다시 부재가 추가되면 부재객체 생성과 동시에 접합객체가 생성된다.

#### 3.3.1 분할 전 제품 정보의 계층구조

Base판을 이루는 부재명을 P1라하고, 접속되는 부재를 P2라 할때, 생성되는 접합객체의 인스턴스는 P1\_P2라는 이름으로 시스템이 자동 생성시킨다. 즉 '-'을 사이에 두고 기존부재와 추가부재를 잇는 접합객체를 생성하는 것이며, '-'은 접합의 의미를 내포하고 있는 것이다. 이렇게 함으로써 객체의 이름만 보더라도, 어떤 정보가 포함되어 있는지 알 수 있게 된다. 객체의 생성이 끝나면, 어셈블리를 정의하고 블록 이름을 지정하여 블록을 만들어낸다. 이렇게 함으로써 어셈블리에 포함되는 객체는 P1, P2, P1\_P2가 된다.

#### 3.3.2 분할 후 제품 정보의 계층구조

분할 후의 제품정보는 분할 전의 제품정보가 세분화되어 가는 과정이다.

부재 인스턴스의 이름은 모부재의 이름 뒤에 분할면의 방향벡터 방향의 부재에는 '-1'을, 반대방향에 있는 부재는 '-2'를 덧붙이게 되며, 이렇게 생성된 부재 인스턴스는 분할전 모부재 인스턴스의 하위부품으로 속하게 된다. 또한 부재의 속성중 Split-Side의 값을 분할도중 입력하게 되는 데, 분할면의 양의 방향의 부재에는 1을, 음의 방향의 부재에는 2가 입력된다. 이렇게 하는 이유는 분할을 마친 후 어셈블리 셋(Set)을 실행하기 위함이다.

접합 인스턴스의 이름은 다음의 두가지 경우로 나누어진다. 첫째, 부재객체 분할로 인한 접합객체의 생성시에는 분할된 두개의 부재 이름을 '-'으로 연결

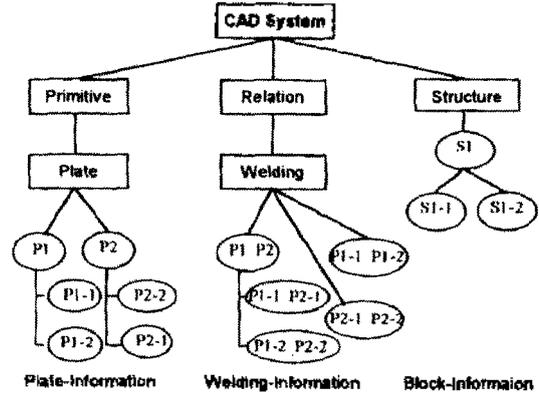
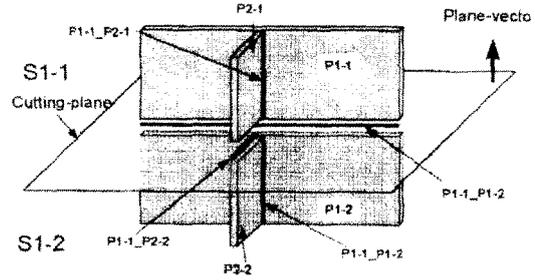


Fig. 8. A hierarchy of information-data after cutting block.

하며, 분할면의 방향벡터쪽의 부재 인스턴스명이 앞에 오게 된다. 둘째, 접합객체의 분할시에는 분할전의 모 접합 인스턴스명을 이용하는데, 분할면의 방향벡터쪽의 접합인스턴스명은 모 접합 인스턴스명의 각 부재명에 '-1'을 붙이고, 반대쪽의 접합인스턴스명은 모접합 인스턴스명의 각 부재명에 '-2'를 붙인다. 이러한 규칙에 따라 접합인스턴스명을 붙여야만 지금까지의 명칭규칙에 위배되지 않고 일관성을 가지는 인스턴스명이 될 수 있다.

이렇게 분할을 모두 마치고 나면, 각 부재의 Split-Side 슬롯 값을 판단하여 모 조립체의 하위 조립체로 어셈블리 셋(Set)을 실행하게 된다. Fig. 8에서 보면, 양의 방향 블록은 S1-1, 음의 방향 블록은 S1-2가 생성된다.

## 4. 월드 와이드 웹(World Wide Web)과의 연계

### 4.1 전문가넷과 WWW연계의 필요성

조선업의 특성상 수주 및 제품 사양 결정 등을 전세계에 흩어져 있는 선주와 협의해야 하며, 사내의

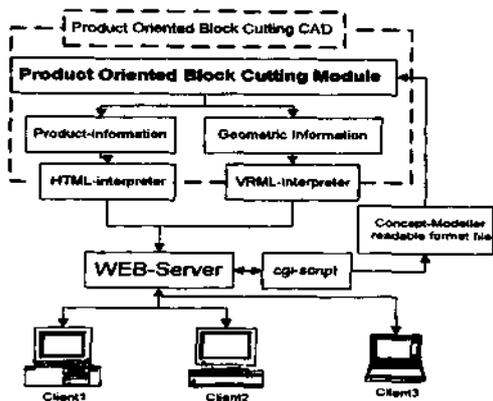


Fig. 9. A relationship between block-cutting module and WWW.

다른 부서와의 협동 작업이 부득이 필요한 상황이다. 또한 작업 현장과의 연계성을 고려한다면, 다양한 OS와 플랫폼에서 각종 정보를 검색하고, 편집할 수 있어야 진정한 협동시스템(Collaborative System)이라고 할 수 있다.

이런 필요성 때문에 필자는 본 연구에서 전문가웹(Concept Modeler, Concentra사, 미국)과 Web과의 연계를 제안한다. 즉, 필요 정보를 웹 서버에 저장함으로써, 개방적이고, 동시적인 인프라(Infrastructure)를 제공하고 있는 월드 와이드 웹과 연결함으로써, 이런 필요를 충족시키고자 한다.

Fig. 9는 블록분할 CAD와 웹과의 연계를 설계한 개념도이다. 블록분할 모듈에서 제공되는 블록에 대한 기하학적 정보는 VRML<sup>[6]</sup>로 변환하고, 기타 제품 정보는 HTML로 변환하여, 관련된 VRML정보와 HTML정보를 서로 링크시켜, 전 세계 어느곳에서나 사용자는 가상현실 속에서 작업될 블록사이클을 이리저리 이동해 가며, 블록모델을 점검하고, 제품정보가 필요하다면 그 부재를 클릭하여 관련 정보를 참조하면 된다. 실제로 블록을 조립하는 현장에서는 블록의 모양이 용접 자세 등의 작업 환경에 많은 영향을 주고 있으므로, 블록 설계시 가상 현실 속에서 설계자가 미리 작업 환경을 체험하게 된다면, 설계 시의 오류를 많이 줄일 수 있을 것이라 판단된다. 또한 클라이언트로 분할 모듈에 접속하여, 분할을 실행함으로써, 전문가웹의 플랫폼 한정이라는 단점을 극복하고, 진정한 협동 블록 분할 CAD시스템이 될 수 있다.

4.2 전문가웹과 WWW와의 인터페이스모듈설계  
인터페이스 모듈은 크게 2가지로 나눌 수 있다. 전

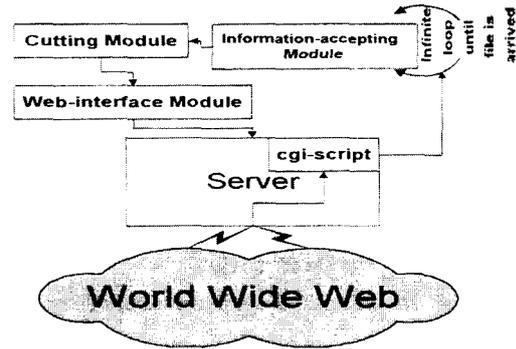


Fig. 10. An algorithm of remote-cutting by web-client.

문가웹 안에 내장되는 네트워크 모듈과 서버 측에서 작동하는 모듈이다.

전문가웹 안에 내장되는 모듈은 다시 두개의 서브 모듈로 나누어지게 되는데, 하나는 출력 모듈이고, 다른 하나는 입력 모듈이다.

출력 모듈은 시스템이 가지고 있는 각종 정보들을 클라이언트측의 브라우저로 읽을 수 있는 정보로 변환하는 모듈이다. 즉, 기하학적 정보는 VRML문서로 변환하고, 기타 제품정보는 HTML문서로 변환하는 모듈이다.

입력 모듈은 제품의 변환된 정보가 서버로부터 도착할 때까지 일정장소에서 기다리는 무한루프이다. 클라이언트에서 보내온 정보가 서버 측에서 가공되어 일정장소로 보내지게 되면, 입력 모듈은 이 정보를 받아 전문가웹에서 작업을 수행하며, 그 결과는 다시 출력 모듈로 보내지게 된다.

다른 하나의 모듈인 서버측의 모듈은 CGI-Script<sup>[7]</sup>를 이용한 것으로, 클라이언트, 서버, 전문가웹을 연결시켜주는 모듈이다. 즉 클라이언트측에서 서버로 넘어온 정보를 전문가웹에서 읽을 수 있는 파일로 변경하여, 입력 모듈이 기다리고 있는 일정장소로 보내주는 기능을 하고 있다.

Fig. 10은 클라이언트 측에서 블록 분할을 실행했을 때 데이터 흐름을 나타내는 그림이다.

4.3 웹 서버를 통한 다자간 협동 시스템(Multi-user Collaborative System)

본 연구에서는 블록 분할 모듈의 사용자와 다수의 클라이언트들 간의 작업이 서버를 통하여 실시간 상호작용 하도록 하고 있다. 즉 분할 모듈에서 정보의 변화가 생겨 서버측에 전달되면, 서버는 참여 중인 모든 클라이언트들에게 이를 알린 다음, 클라이언트

측에서 피드백 되는 정보를 기다리는 방식이며, 이는 Webchat(웹을 통하여 대화방을 가능하게 하는 시스템) 시스템의 구조와 매우 흡사하다고 볼 수 있다.

#### 4.4 VRML/HTML 변환기

이미 여러종류의 VRML변환기가 개발되어 있는 상태이지만, 이런 변환기는 기존의 CAD 데이터를 변환하는 것들로서 AutoCad의 인터페이스 형식인 DXF를 변환하는 변환기, 또는 IGES파일을 변환하는 변환기들이다. 그러나, 이러한 변환기를 사용하기 위해서는 전문가셀 내의 기하정보를 기존의 CAD데이터 호환 형식으로 바꾸어 다시 VRML형식으로 전환해야 하는 번거로움이 발생하고, 이러한 과정에서 DATA의 분실이 발생할 수 있다. 그래서, 본 시스템에서는 제품정보 중 객체의 기하학적 속성값을 전문가 시스템에서 직접 추출하여 VRML형식의 파일로 변환하는 방법을 쓰고 있다. VRML변환은 지정된 블록별로 행하여지며, 블록에 속하는 모든 부재들이 일시에 변환되며, 사용자는 변환하고자 하는 블록을 마우스로 지정하면 된다. 또한 해당 부재가 VRML변환이 끝날 때마다, 그 부재의 기타 제품정보는 HTML파일로 자동 전환되며, 그 부재의 VRML파일과 자동 링크가 형성된다.

VRML 형식은 제품의 형상표현을 위해 몇가지 기본적인 특징형상들을 지원하는데, 이들은 노드(Node) 형식을 취하고 있다. 이 노드중에는 Cubic, Cylinder, Cone, Face 등이 존재하는데, 본 시스템에서 필요한 특징형상은 부재객체의 형상표현을 위하여 Face를 구현하는 노드명 'Coordinate 3'와, 접합객체의 표현을 위한 노드명 'Cylinder'이다. 또한, 해당부재의 기타 정보(HTML정보)를 형상정보(VRML정보)에 연결하기 위하여 'WWWAnchor'노드를 사용하고, Topology 정보를 부여하기 위해 'IndexedFaceSet'노드를 사용하여 'coordIndex'를 추가시킨다.

본 시스템에서 적용하고 있는 HTML형식은 이 "Mozilla extension"을 가지고 표현하고 있는데, 그 이유는 속성값을 구별하기 쉽게 Table형식을 갖출 수 있기 때문이다. 이런 형식을 취함으로써 전문가셀 내의 각 객체의 내부 속성값을 Property/Value형식으로 간단 명료하게 기술할 수 있다. 시스템은 슬롯값으로 이미 지정되어 있는 값뿐 아니라, Undemanded상태의 모든 슬롯값을 역추적하여 찾아내며, 이는 변환 실행 이벤트가 발생하면서 연달아 실행되게 된다. 예를 들어 시스템은 블록 전체의 높이, 너비, 길이 등의 정보들을 이러한 이벤트 발생시, 블

록에 포함된 모든 부품의 높이, 너비, 길이들을 비교하여 계산한다.

##### 4.4.1 전문가셀 밖의 연결 모듈

전문가셀 밖의 연결 모듈은 2가지로 나누어진다. CGI-Script를 이용하여 서버 측에 내장되는 모듈과 WWW서버로부터 정보가 도착되면, 이를 전문가셀 안의 모듈에 도착 소식을 알리는 모듈이다.

서버 측에 내장되는 모듈의 역할은 블록분할 CAD 모듈에서 전달된 정보를 클라이언트들에게 브로드캐스트하고, 클라이언트측에서 보내온 정보를 전문가셀 입력모듈과 호환되는 형식으로 바꾸어 전달하는 기능이다. 이 과정에서 블록분할 정보를 보내온 클라이언트의 정보를 첨가하여 전문가셀의 입력모듈로 보낸다. 이렇게 함으로써, 블록분할 CAD 모듈은 블록분할에 필요한 정보 뿐만 아니라 분할정보가 어느 클라이언트에서 도착된 정보인지 구분할 수 있게 된다.

전문가셀 밖의 또 다른 모듈은 클라이언트들로부터 정보가 서버를 거쳐 도달할 때를 기다려 정보가 도착되면 사용자에게 정보도착을 알리는 모듈이다. 이 모듈은 전문가셀 밖에 C언어를 이용하여 구현되었는데, 클라이언트로부터 정보를 기다리는 중에도 전문가셀 안에서 분할 및 기타 작업을 할 수 있다.

#### 4.5 버전 컨트롤(Version-Control)

전문가셀이 웹과 연동함에 따라, 블록 컷의 주체가 복수화 된다. 즉 블록 분할을 하는 사용자에게 따라 한 개의 블록을 대상으로 다양한 분할이 가능해진다. 따라서, 분할 주체별로 블록모델을 따로 관리하는 버전 컨트롤 기능이 필요하다. 이를 위하여 분할 정보에는 반드시 분할주체의 IP주소와 분할하고자 하는 블록명이 명시되어야 한다.

분할정보를 생산한 클라이언트의 IP주소를 모델명에 덧붙임으로써, 클라이언트로부터 도착한 각각의 정보를 따로 관리하게 된다. 전문가셀 사용자가 모델을 구축하여 저장하면, 원시 블록 데이터가 생성된다. 클라이언트로부터 정보가 도착되면, 구축된 모델을 블록 원시 데이터(Reserved Block Source-Data)를 이용, 다시 열어 블록 분할을 실시한다. 이때 앞서 언급한 바와 같이 시스템은 모델명을 클라이언트의 IP주소를 덧붙여 달리 설정하여, 각각의 클라이언트별로 블록모델을 관리한다.

#### 4.6 메시지 전달(message sending)

본 연구의 메시지 전달 기능은 시스템이 보여주는

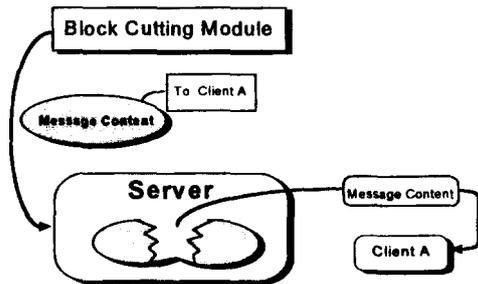


Fig. 11. An information flow of message-sending to a client.

오류내용을 분할 주체인 웹 클라이언트에게 보내주는 것이다. 메시지 전달기능은 종전의 블록 분할 모델을 브로드캐스팅(Broadcasting)하는 것과 달리, 담당 클라이언트에게만 메시지가 전달된다.

예를 들어 A와 B클라이언트가 동시에 접속해 분할을 실행하여 A는 성공적인 분할을 실행하였고, B는 분할평면을 블록경계 밖에 위치시켜 오류를 발생하면, 클라이언트 A에게는 분할 성공 메시지를 보내고 클라이언트 B에게는 오류의 메시지를 보내준다.

Fig. 11과 같이 분할모듈은 발생한 메시지에 수신할 클라이언트의 주소를 덧붙여 서버에 전달하고, 서버는 분할모듈로부터 오는 메시지 정보를 분류하여 해당되는 클라이언트에게 전달한다. 또한 서버는 분할모듈로부터 오는 오류 메시지를 열어서 그 내용을 해당 클라이언트에게 보여 준다.

클라이언트로 전달되는 오류에는, 첫째 분할평면이 블록경계 밖에 위치했을 경우, 둘째 일단 분할되어 완성된 블록을 다시 분할하고자 했을 경우, 그 밖에 분할모듈에서는 삭제된 모델을 분할하고자 했을 경우 등이다.

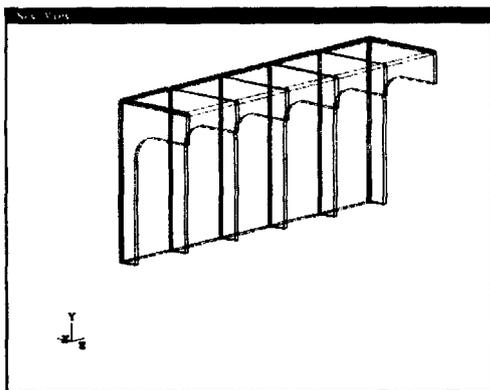


Fig. 12. An example view of one-side block after cutting in system.

### 5. 시스템 구현

본 연구에 사용된 서버는 Sun-Sparc10 워크스테이션이며 CGI-script 프로그래밍 언어로 Perl(v.4.0)<sup>16)</sup>을 이용하였다. 클라이언트의 웹 브라우저는 MicroSoft

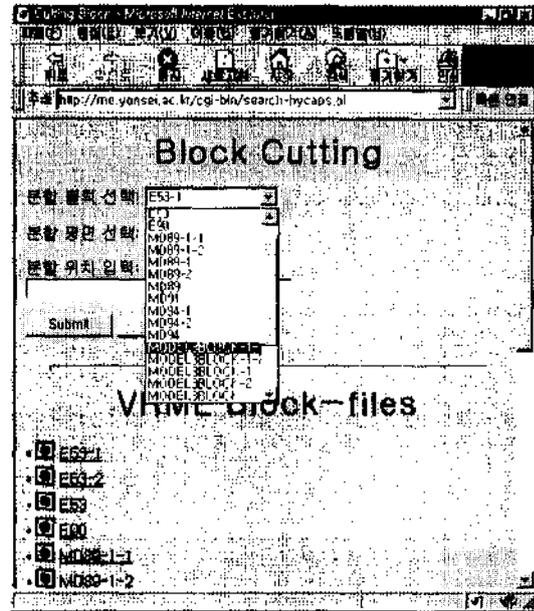


Fig. 13. A HTML-browser view when remote-users access a sever to cut a block.

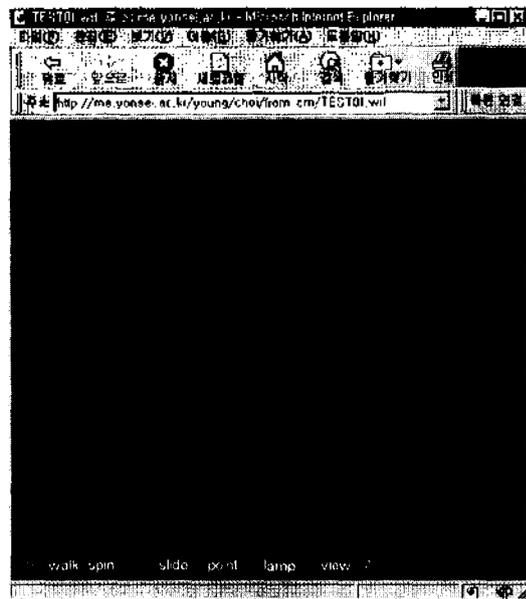


Fig. 14. A user is navigating a virtual block-model with VRML-browser.

사의 Internet-Explorer를 이용하였고, VRML브라우저는 Netscape사의 Live3D를 이용하였다. 또한 시스템의 검증을 위하여 화물선의 중앙부(Bulkhead) 구조를 모델로 하였다. Fig. 12, 13, 14는 구현된 시스템의 출력화면을 나타낸다.

## 6. 결 론

설계 및 생산 단계에서 필요한 정보를 설계 초기부터 용이하게 획득할 수 있는 시스템인 생산지향성 선박의 블록분할 CAD모듈을 개발한 본 연구에서 다음과 같은 결과를 얻었다.

첫째, 기존의 블록 분할 방식 즉, 분할선을 설정한 후, 부재를 다시 정의(Part Generation)하는 이중작업 방식을 한번의 분할평면 설정으로 가능하게 하여 블록 설계시의 편의성이 향상된다.

둘째, 선박의 블록 조립시 필수적인 접합 관련정보를 블록 모델링과 동시에 시스템 자체에서 자동으로 생성해주는 접합객체에 연결시켜 줌으로써, 선박 설계시 용접과 관련된 기타 수작업의 번거로움을 없앨 수 있고, 접합정보를 체계적으로 다룰 수 있는 데이터 구조를 생성할 수 있다.

셋째, 가상 현실 체험으로 실제 생산시에 발생할 문제점을 미리 파악할 수 있다. WWW와 연동함으로써, 이종간의 하드웨어 및 OS환경에서도 호환 가능하게 하여 개방성, 동시성 및 협동성을 갖춘 시스템이 된다.

## 참고문헌

1. 강원수, 서승환, 이규열, "객체 지향 선박 구획 정의 표현 방법론", 대한조선학회 논문집 제 30권, 제 3호, pp. 248-307, 1993.
2. Cho, K.K., Lee, S.H. and Jeong, D.S., "An Automatic Process -Planning System for Block Assembly in Shipbuilding", *Annals of the CIRP*, Vol.

45/1, pp. 41-44, August, 1996.

3. Taber, M., *HTML and CGI unleashed*, SamsNet publish, pp. 252-301, 1995.
4. Koyama, T. and Honda, K., "A Fundamental Study on Manufacturing Planning and Control System for Shipbuilding CIM", 일본조선학회논문집 제 166호, pp. 415-423, 1989.
5. Mäntylä, M., *An Introduction to Solid Modeling*, ADTEC, pp. 242-248, 1987.
6. Ames, A., Nadeau, D. and Moreland, J., *The VRML Sourcebook*, John Wilcy, pp. 1-7, 1996.
7. Taber, M., *HTML and CGI unleashed*, SamsNet publish, pp. 252-301, 1995.
8. *The Perl Programming Language*, at "http://pubweb.nexor.co.uk/public/perl/perl.html".



## 최 해 진

1995년 연세대학교 기계공학과 학사  
1997년 연세대학교 기계설계학과 석사  
1997년 ~ 1998년 미국 Georgia Institute of Technology 연수  
관심분야: 동시공학설계, CIM, 네트워크 기반 설계 시스템



## 이 수 홍

1981년 서울대학교 기계공학과 학사  
1983년 서울대학교 기계설계학과 석사  
1991년 Stanford대학 Design Division Concurrent Engineering 전공 박사  
1991년 ~ 1992년 Lockheed Missile and Space Co. Cable Harness Design System 개발 Post-Doc.  
1983년 ~ 1994년 KJMM CAD/CAM실, 선임연구원  
1994년 ~ 현재 연세대학교 기계공학과 부교수  
관심분야: 동시공학 설계, 지식기반시스템 설계, DFM