

Web 기반형 CAE 정보 전달 시스템 구현을 위한 유한 요소 해석 결과의 VRML 변환기술 개발

황호진*, 이영석*, 오재우*, 박만진*, 장동영**

* 서울산업대학교 정보산업공학과 대학원 연구조교

** 서울산업대학교 산업정보시스템공학과 부교수

Development of a Technique to Transfer Output from FEM Analysis to VRML Format for the Web Based CAE Information System

Ho Jin Hwang*, Young Seok Lee*, Jea Woo Oh*, Man Jin Park*,
Dong Young Jang**

* R.A, ** Associate Professor

Department of Industrial & Information Systems Engineering
Seoul National University of Technology

주요어: 가상현실(Virtual Reality), 유한요소해석(FEA), 성형가공

Abstract

VR(Virtual Reality)의 개념이 제조공학 분야로 확대 적용되면서 설계분야에서의 가상조형(Virtual Prototyping), 제조분야에서의 가상제조시스템(Virtual Manufacturing System)이라는 새로운 형태의 제조 시스템 및 설계기술이 차세대 제조기술로서 발전해 오고있으며 이런 가상제조방법(Virtual Manufacturing)은 컴퓨터를 이용하여 생산과정에서 제품뿐만이 아니라 유한요소해석을 유용하여, 공정까지도 모의실험을 하며 생산성(manufacturability), 최종 형태(final shape), 잔류 응력(residual stress), 제품 수명 주기(life-cycle)등과 같은 요소들의 최적화 정보를 제공하게 된다. 본 연구에서는 이러한 가상제조방법을 구현하기 위한 기초적 단계로서 가상단조기계를 가상현실기법을 이용하여 모델링하고 Java를 이용하여 모델링된 단조기계를 제어하고 제조되는 단조물에 대한 내부 응력 및 형태변환을 유한요소법을 이용하여 분석하고 분석된 단조물의 자료를 WEB상을 통하여 볼 수 있게 유한요소 해석결과를 VRML언어로 변환하는 작업을 수행하였다.

1. 서 론

설계과정에서 CAD를 이용하여 그려진 설계도를 공학적인 관점에서 그 적합성을 판단하기 위해 공학적인 분석과 평가 (Computer Aided Engineering: CAE)가 필요하다. CAE의 활용 분야는 대략 두가지 측면에서 살펴볼수 있다. 첫째는 전통적인 적용분야로 설계된 Design의 검증이나 문제가 발생하였을 때의 해결 방안 모색을 찾기 위해 활용되며 이때에는 엄밀하고도 정확한 해석이 수행되어야 한다. 따라서 이러한 해석은 주로 CAE를 전공한 엔지니어에 의해 수행되어져 왔다. 둘째는 최적화 설계를 위한 CAE 해석 분야이다. 최근에는 설계 조건이 엄밀해지고 최적화 설계 (중량 최소화, 강도 최대화 등)가 요구됨에 따라 설계시에 CAE해석을 반영할 필요가 발생하였다. 따라서 이 분야의 해석은 주로 설계자에 의해 수행되어야 하며 해석의 정밀도와 더불어 해석 시간 및 해석의 용이성이 중요하게 된다. 이를 위하여 여러 가지 분석도구들이 개발되었고, 특히 유한요소법을 이용한 공학적 해석이 가장 널리 사용되고 있다. 유한 요소 분석(Finite Element Analysis : FEA), 혹은 유한 요소법(Finite Element Method: FEM)은 설계 해석의 도구 중 가장 중요기술이 되었으며, 점점 더 정확하고 정교한 계산을 필요로 하는 현대의 복잡한 설계 환경으로 인해 FEA가 발전되었고, 그 발전과 함께 설계 과정도 개선되었다 하겠다. 분석을 위한 거대한 양의 계산을 하면서, 동시에 그 결과를 보여줄 수 있는 컴퓨터의 발전이 FEA의 사용을 더욱 용이하게 해 주었다.

최근 들어 급속도로 확산되고 있는 인터넷 사용에 힘입어 정보의 공유, 정보의 전달, web기반 가상작기계의 개발에 대한 연구가 활발하게 진행되고 있다. 이러한 연구들은 인터넷상의 막대한 데이터 처리 및 송수신 속도와 같은 한계로 넘어 가상 생산 모사를 web을 통해 사용자들로 하여금 공정 시물레이션을 가능케 해주고 있다. 오늘날의 기업 환경에서 신제품의 개발은 이전과 같은 제품 구상, 도면화, 금형 제작이나 가공으로 이어지는 단 방향적인 흐름에서 탈피하여 모든 공정의 정보가 쌍방향으로 유기적으로 연결되는 동시 공학적 방법이 요구된다. 즉 제품 설계의 의미는 과거와 같이

단순히 부품도와 조립도로 이루어지는 형상 정보를 제공하는 역할에 더하여 제품 개발의 전과정에 필요한 모든 정보를 포함하고 자유롭게 정보의 변경과 결합이 전체공정에 미치는 영향을 평가할 수 있는 도구를 제공하는 데 더 큰 비중이 주어지고 있다. 즉, 3차원 CAD를 이용하여 생성된 형상 정보는 정적 및 동적 역학 해석에 필요한 모델의 제공, 자동 가공을 위한 공구 경로 작성 등에 이용되어 제품 개발 공정의 통합화에 이용되고 있다. 그러나 좀 더 능률적인 설계환경 구축을 위해서는 각 사용자의 개발 제품군 분야에 맞도록 더욱 사용자 위주의 설계 도구들이 연구되어야 한다. 개발 제품군에 공통으로 필요한 조립 및 단품들을 선정하여 이런 요소들의 데이터 베이스(DB)를 구축하며, 부품들의 3차원 모델은 파라메트릭 모델링 기법을 효율적으로 활용하여 중요 인자를 변경하였을 때 관련 단품이나 준조립체, 더 나아가서는 제품의 외관 형상의 변화까지 명확히 파악될 수 있도록 하여야 한다. 아울러 이런 조립요소들의 조립성이나 조립 순서 등을 평가하고 수율을 고려한 적절한 공차 부여를 할 수 있는 도구를 구축하여 가공 정밀도를 고려한 가공 계획을 수립할 수 있는 기준을 제공하고, 향후 DB의 부품 및 조립체 들은 WEB을 통하여 기본적인 CAE해석 결과와 같이 제공되어 사용자가 더욱 많은 정보를 가지고 제품 구상 및 설계를 진행 할 수 있도록 하여야 한다. 부품의 DB는 가상 현실 기법(Virtual Reality: VR)과 결합하여 초기단계에서는 DB에서 선택한 부품으로 가상 시제품을 조립하여 보는 Digital Mock-Up 환경을 구축하고 이후는 VR 환경에서 Parameter들을 조작하여 전 설계과정을 모사할 수 있는 System을 구축한다. 또 이 DB를 이용하여 급속조형 기법으로 시제품의 형상을 신속하게 제작하여 평가하고 CAM을 이용하여 부품 및 금형을 제작할 수 있는 공구 경로를 작성하는데 오류를 최소화하는 데이터 생성을 하기 위한 모델링 기법이나 데이터 구조에 관한 연구도 수반 되어야 한다.

웹을 기반으로 하여 3차원 모델링 및 동적인 3D 장면 기술을 가장 쉽게 효율적으로 할 수 있는 파일 포맷은 VRML(Virtual Reality Modeling

Language)이다. 세계 표준이며, Java와의 결합이 아주 용이하기 때문에, 이미 많이 사용되고 있으며, 거의 모든 상용 CAD툴, 3D 그래픽 패키지 및 유한요소 해석툴들은 모두 VRML로의 export기능을 제공하고 있다. 그러나, 이렇게 변환된 VRML의 구조들은 패키지가 갖고 있는 모델링 형식을 기초로 변환되었기 때문에 중복되는 폴리곤(polygon)이 많이 발생하고, 불필요한 부분까지 변환되며, 해석툴의 경우 각 해석 단계당 결과값만 변환 할 뿐, 해석 전과정을 모사하는 기능은 현재까지 제공하지 못하고 있다.

따라서, 본 연구의 목적은 유한요소해석의 결과 값을 VRML로 변환하는, 데이터 가공 프로그램개발을 수행하고, 실제 제조과정과 함께 유한요소해석 결과가 모사될 수 있도록 가상 단조 시스템을 개발하여, 실제 공정수행 및 해석후 결과를 web을 통하여 확인할 수 있게 하는 것이다. 세부 연구 내용은 다음과 같다.

① 데이터 가공 프로그램 개발

본 연구에서는 단조공정 중 딥드로인(Deep drawing)공정을 대상으로 하여 유한요소해석을 수행하였고, ASCII형식의 해석 결과값을 VRML로 변환하는 프로그램을 개발하였다.

② 가상 단조 시뮬레이션 시스템 개발

실제 딥드로인(Deep Drawing)을 수행하는 프레스를 모델링하고, 기계동작에 대한 핸들링 및 사용자 인터페이스는 JAVA를 이용하여 구성하였다. 또한 VRML로 변환된 유한요소 해석 결과 값을 가상 프레스에 적용하여, 실제 사용자가 web을 통해 가상프레스를 작동하여 딥드로인공정을 수행하게 하였으며, 이 때 발생하는 요소(element)들의 변화를 직접 확인할 수 있게 하였다.

2. 시스템 구현 요소기술

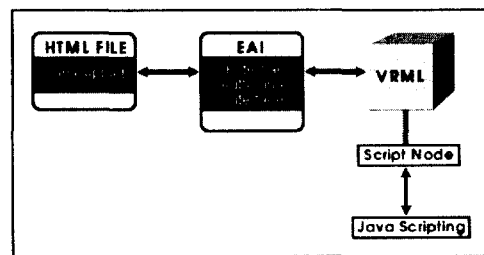
2.1 VRML

VRML(Virtual Reality Modeling Language)는 인터넷을 통하여 3차원의 모형을 구현하고 상호 작용할 수 있는 장면 묘사(scene description language)그래픽언어이다. VRML은 컴파일 되지 않고 ASCII 파일 형태로 사용자 컴퓨터로 다운로드

된 후 VRML 브라우저에 의해 해석된다. 따라서 VRML은 일반 텍스트 에디터를 통해 작성이 가능하며, 저작툴을 통해 만들어진 VRML 파일 역시 텍스트 에디터를 통해 편집이 가능하다. VRML은 노드(node)안에 있는 필드(field)값을 다른 노드의 필드값에 연결하여 필드값을 변화시키고, 따라서 장면(Scene)을 변화시키는 이벤트중심(Event-driven)의 구조이다. 최근들어 대부분의 CAD 패키지들은 디자인된 모델을 WRL 파일 포맷으로 변환하는 기능들을 제공하고 있다. 실제 외부환경과 연결하기 위해 VRML과의 인터페이스가 필요한데 현재 존재하고 있는 인터페이스는 SAI(Script Authoring Interface)와 EAI(External Authoring Interface)가 있으며, 자바언어를 사용하여 VRML의 내용 변경이 가능하도록 인터페이스를 제공하는 VRML 브라우저의 사용이 요구된다.

2.2 EAI

애플릿에서 VRML 가상세계를 제어하기 위해, EAI는 VRML세계와 외부 환경과의 인터페이스(Interface)를 제공하는 것으로 Script 노드의 자바 프로그램과 유사하나, VRML 파일에 포함되지 않고 VRML 외부 환경인 애플릿에서 비동기적으로 가상세계를 제어한다. HTML의 EMBED 태그를 사용하여 VRML 문서를 HTML 문서내에 포함시킬 수 있다. EAI로 다양하고 친숙한 사용자 인터페이스를 구성할 수 있으며, JAVA로 가능하면 어떤 것이든 VRML과 연결시킬 수 있다. [그림 1]에서 보듯이 EAI는 외부프로그램(주로 Java 애플릿)이 VRML 장면(Scene)에 있는 노드에 접근할 수 있도록 기능을 제공한다. 다시 말해 Java 애플릿의 버튼을 눌러 HTML에 포함되어 있는 여러개의 VRML파일의 노드에 접근할 수 있는 것이다.



[그림 1] Java와 EAI, VRML의 관계

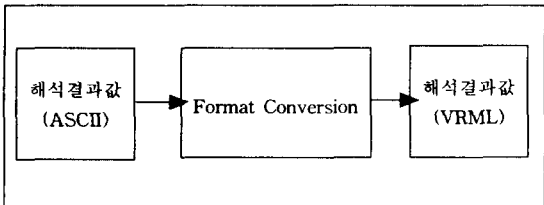
3. 유한요소 해석 결과의 VRML Format 변환

3.1 해석 프로그램에 의한 산출된 파일 분석

단조물에 대한 해석은 상용 유한요소법 프로그램과 개발자가 직접 개발하는 프로그램으로 계산이 가능하다. 본 논문에서는 개발한 해석 프로그램을 이용하여 얻어진 결과값을 가지고 VRML File로 변환하도록 하였다.

먼저 이 해석 프로그램에 의하여 얻어지는 결과값의 Mesh Output File 저장 형식은 크게 4부분으로 나누어진다.

- (1) 상태 표시부
- (2) header 부
- (3) Object의 형상부
- (4) status 부



[그림 2] 해석결과 값이 VRML로 변환하는 과정

대상 공작물의 형상을 나타내기 위해 필요한 데이터(node의 개수, elements의 개수, 각 노드의 x,y,z 위치정보)는 모두 Object의 형상부와 status부에 포함되어 있기 때문에 이후의 설명은 이 둘에 관한 것이다. 먼저, 대략적인 Mesh Output File 형식에서 Object 형상부에 대한 세부 설명이다.

3.1.1 Object 형상부

다음과 같은 규칙에 따라서 출력의 형식이 정해진다.

- rule 1) header부에 기입한 순서대로 읽는다.
- rule 2) Object간에 줄은 삽입하지 않는다.
- rule 3) Object 형상은 모든 state에서 기록한다.

○ (format)

- no. of nodes
- no. of elements
- no. of active nodes

```

no. of active elements internal node no
user_node_no
current_x current_y current_z ....
    
```

object 형상부의 도입부분은 간단한 해석규칙을 몇 전제적인 형상정보를 나타내고 있다. 이 곳에서 정의하고 있는 형상정보는 노드(node)의 개수, 요소(element)의 개수, 현재 활성화된(active) 노드의 개수, 활성화된 요소의 구성, 현재 노드의 x,y,z 좌표값을 말한다.

```

1681 1600 1681 1600
1 30001 0.0000e+000 0.0000e+000 3.9000e-001
.....
1681 31681 7.5000e+001 7.5000e+001 3.9000e-001
    
```

[그림 3] 해석결과 파일(ASCII)의 Object 형상부 1

예를 들어 [그림 3]처럼 Object 형상부가 정의되어 있으면, 1681개의 elements가 있으며 모두가 active하다는 것을 보여주고, 두 번째 줄은 1번 노드의 위치값(x, y, z)을 표시하고 있다. 마지막 줄은 1681번째 노드의 위치값을 나타낸다.

○

```

internal el. No user el. no group no
node-1 node-2 node-3 node-4
... ..
... ..
    
```

이 부분은 각 elements가 몇번 노드들로 이루어졌는지를 설명하는 부분이다. 예를들면,

```

1 10001 5 1 42 43 2
... ..
1600 10160 5 1639 1680 1681 1640
    
```

[그림 4] 해석결과 파일(ASCII)의 Object 형상부 2

[그림 4]는 각 elements가 몇번 노드들로 이루어졌는지를 설명하는 부분으로 첫 번째 줄은 1번 elements가 1번 node와 42번, 43번, 2번 노드로 이

루어졌음을 말한다. 따라서 이 element는 사각형 구조이다.

3.1.2 status 부

Status부는 각 object의 현재 상태를 기록한다. header부에서 지정한대로 1번에 1개의 데이터씩을 기록하고, 각 데이터 사이는 줄을 띄우지 않는다. 예를 들어, node가 1681개이고 displacement, velocity, Normal_Force, Penetration, n_plane_velocity를 기록하는 경우, 먼저 displacement를 기록하고, 그다음에 각 노드의 변위량과 속도의 정보를 적어준다.

3.2 Mesh Output File을 기반으로 VRML Format으로 만들기

해석결과데이터를 분석하여서 VRML File로 만든다. 3.1에서 설명된 결과데이터의 Object부의 노드와 elements값들을 이용하여 VRML File을 만든다.

1) 각 노드값을 VRML point 값으로 변환

각 노드값을 먼저 실수형으로 바꾸고, 단조물의 처음 형상 노드값과 단조공정이 수행된 후 각 노드값의 변화량을 읽어서 각 노드값의 현재 위치를 알아낸다. 이는 해석결과 데이터에서 Object부의 노드의 위치값과 status에서 displacement값을 더해주면 된다. 아래의 프로그램은 3111개의 노드와 3000개의 element를 갖는 단조물을 대상으로 계산을 해준 것이다.

```
String texts2 = log.getText();
log.setText("");
String[] tmp = new String[20000];
float n[] = new float[20000];
float d_n[] = new float[20000];
float result_n[] = new float[20000];
int i=0;
String d_newLine = " \n";
StringTokenizer st=new
StringTokenizer(texts2,d_newLine);
while (st.hasMoreTokens()) {
```

```
// 남은 토큰이 있을때까지
    tmp[i] = st.nextToken();
// 첫번째 토큰(0.3)
    tmp[i] = tmp[i].trim();
// 좌우 공백 제거
    n[i]=Float.parseFloat(tmp[i]);
    i++;
}
String test[] = new String[20000];
for(i=0;i<=3*3111-1;i++){
    result_n[i]=n[i]+n[3*3111+i];
}
for(i=0;i<=3111*3-1;i=i+3){
    test[i]= Float.toString(result_n[i]);
    test[i+1]=Float.toString(result_n[i+1]);
    test[i+2]=Float.toString(result_n[i+2]);
    log.append(test[i]+" "+test[i+1]+"
"+test[i+2]+newLine);
}
}
```

2) 각 포인트의 변화량에 따른 색 지정 프로그램

각 포인트의 변화량에 따라 색을 지정하게 되는데 이 논문에서는 총 20가지의 색을 사용하여 그 변화된 량에 맞게 그 값을 지정하게 된다. 먼저 각 포인트의 변화량을 산출하고 그 변화량의 최대값과 최소값을 추출한 다음 그 값들의 차이를 20단계로 나누어서 각 단계에 해당하는 색을 지정하게 된다. 그 변화량의 값은 Object부의 status의 displacement값을 이용하게 되는데 각 노드의 displacement값을 제공하여 더해준 다음 다시 제곱근을 씌워서 변화량을 얻어내게 된다.

○ 각 노드의 변위량

$$\sqrt{x_n^2 + y_n^2 + z_n^2} \quad n=(0,1,2,\dots,3110)$$

이렇게 얻어낸 각 노드의 변위량에서 최대값과 최소값을 추출하여 그 값의 차이를 20개로 나눠서 각 단계에 맞는 색을 지정한다.

각 포인트의 변화량을 계산

```
result_m[(i+3)/3]=Math.sqrt(Math.pow(n[i],2.0)+Math.pow(n[i+1],2.0)+Math.pow(n[i+2],2.0));
result_n[(i+3)/3]=(float)result_m[(i+3)/3];
```

변화량의 maximum값과 minimum값을 추출

```

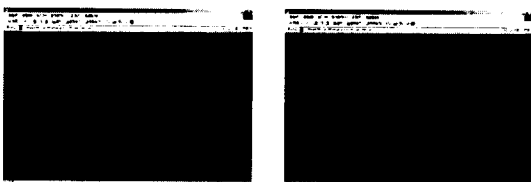
for(i=0;i<=3110;i++){
  if(t-result_n[i]>=0){
    t=t; }
  else{
    t=result_n[i]; }
}
for(i=1;i<=3110;i++){
  if(s-result_n[i]<=0){
    s=s; }
  else{
    s=result_n[i]; }
}
    
```

3) elements를 이용하여 point 연결

VRML에서 각 point를 연결하기 위해서 element 값을 이용한다. 하나의 element는 4개의 point로 구성이 된다. 본 논문에서 적용된 사례에서는 3000개의 element가 존재한다. element는 4개의 인자값을 가지게되는데 4개의 인자값은 이 element가 4각형임을 의미하게 된다. 또한 각각의 인자값의 value로는 위에서 구한 포인트의 위치값을 가지게되는데 예를 들면 { 311, 433, 122, 144 }는 311번째 point와 433번째, 그리고 122, 144번째 point가 하나의 element를 이룬다는 의미이다.

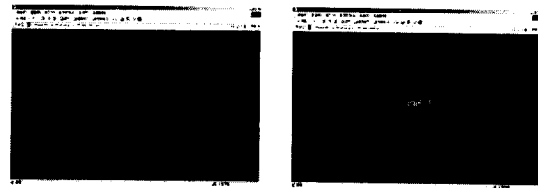
3.3 VRML format으로 변환한 결과값

Deep drawing의 경우, 해석프로그램을 이용하여 분석한 단계적 변환 과정을 VRML로 변환시킨 결과가 [그림 5]에 나타나 있다.



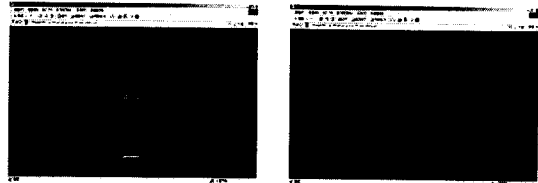
step 1

step 2



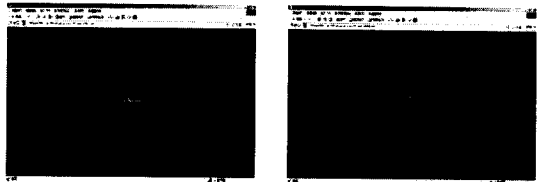
step 3

step 4



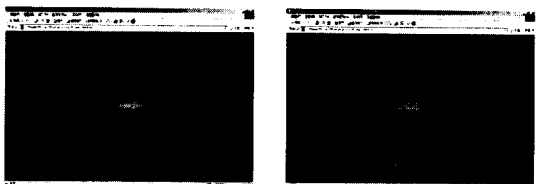
step 5

step 6



step 7

step 8.



step 9

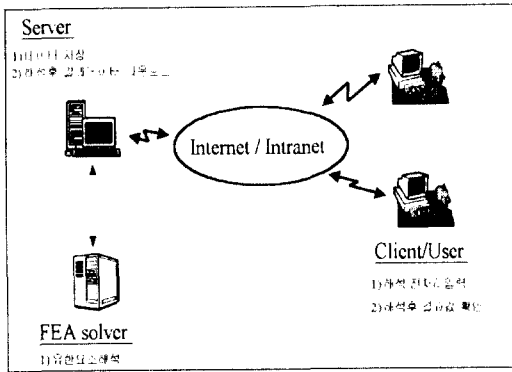
step 10

[그림 5] 각 단계별 VRML 결과값

4. 가상 성형 단조시스템 개발

4.1. 시스템 설계

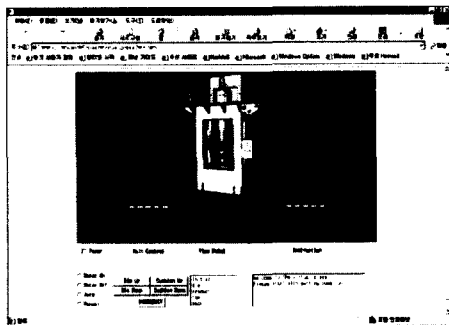
본 시스템의 구성도는 본 연구에서 개발된 『가상 성형가공 시스템』의 구성도를 나타내고 있다 [그림 6참조]. 사용자가 웹 브라우저를 통하여, 가상 단조프레스를 가동하여 단조공정시물레이션을 수행하고, 그 해석결과를 웹을 통하여 확인할 수 있게 된다.



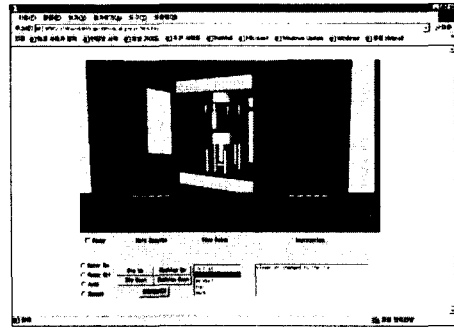
[그림 6] 시스템 구성도

4.2. 시스템 구현

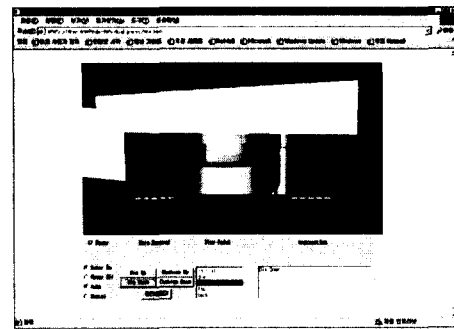
시스템 구현은 먼저 Pro/Engineer(ver.20)을 이용하여 단조프레스와 금형 및 가공물을 모델링하여 3D객체로 생성하였다. 이를 Cosmo Worlds 2.0을 이용하여 금형을 프레스와 결합한 후, 단조프레스의 동작제어를 위해 java를 이용하여 사용자 인터페이스를 구현하였다.[그림7 참조]. 3절에서 소개된 방법론을 사용하여 유한요소해석결과를 VRML로 변환한 후, 가상단조기계에 함께 구현하여, 사용자가 직접 가상 프레스를 가동시켜 덩드로인공정을 수행시킨 후, 그 결과를 직접 확인할 수 있도록 하였다. 그림 8,9,10은 가상 프레스의 구동 및 각 단계별 변형 모습 및 스트레스를 나타내고 있다.



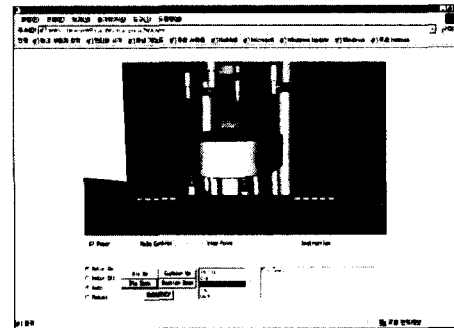
[그림 7] 가상 단조 시스템



[그림 8] 덩드로인 공정 수행(단계1)



[그림 9] 덩드로인 공정 수행(단계2)



[그림 10] 덩드로인 공정 수행(단계3)

5. 결론

최근들어 급속도로 향상되고 있는 초고속 통신망에 힘입어 가상 제조(Virtual Manufacturing) 기술을 인터넷과 결합하려는 연구가 활발하게 진행되고 있고, 그 중 한가지가 web기반 가상공작기계의 개발이다. 그러나, 단조공정은 기계부품의 대량 생산을 목적으로 광범위하게 사용되고 있음에도 불구하고 막대한 데이터양과, 공정상의 특성, 즉,

공정변수간의 밀접한 상호연관관계와 소성유동의 복잡성으로 인해 web기반 시스템구축은 미흡한 실정이다.

웹을 기반으로 하여 3차원 모델링 및 동적인 3D장면 기술을 가장 쉽게 효율적으로 할 수 있는 파일 포맷은 VRML(Virtual Reality Modeling Language)이다. 세계 표준이며, Java와의 결합이 아주 용이하기 때문에 거의 모든 상용 CAD툴, 3D 그래픽 패키지 및 유한요소 해석툴들은 모두 VRML로의 export기능을 제공하고 있다. 그러나, 이렇게 변환된 VRML의 구조들은 패키지가 갖고 있는 모델링 형식을 기초로 변환되었기 때문에 중복되는 폴리곤(polygon)이 많이 발생하고, 불필요한 부분까지 변환되며, 해석툴의 경우 각 해석 단계당 결과값만 변환 할 뿐, 해석 전과정을 모사하는 기능은 현재까지 제공하지 못하고 있다.

본 연구에서는 단조공정 중 딥드로인(Deep Drawing)공정을 대상으로 유한요소해석의 결과 값을 VRML로 변환하는, 데이터 가공 프로그램개발 및 방법론을 소개하였고, 실제 제조과정과 함께 유한요소해석결과가 모사될 수 있도록 가상 단조 시물레이션 시스템을 개발하였다.

참 고 문 헌

- [1] 서윤희, 서석환 외 4명, “Web 기반 가상 공작기계 구현을 위한 시스템 개발“, 대한산업공학회 '99. 추계공동학술대회 논문집, Session A01.5 CAD/CAM1
- [2] Arnold, Ken, “Java programming language” Addison-Wesley 1998
- [3] Jed Hartman, Josie Wernecke “The VRML 2.0 Handbook”, Silicon Graphics 1998
- [4] Michael J. Ryken, Judy M. Vance, “Applying virtual reality techniques to the interactive stress analysis of a tractor lift arm”, Finite Elements in Analysis and Design 35, pp141-1555, 2000.
- [5] T.P Yeh, J.M. Vance, Interactive design of structural systems in a virtual environment, Comput. Modeling Simulation Egn. 3(30)(1998) 166-169.

[6] Virtual Forging Press,

http://camsys.snut.ac.kr/Virtual_Press.html/

[7] External Authoring Interface working

Group <http://www.vrml.org/WorkingGroups>

[8] Claudio Riva “Java and VRML”,

<http://pent224.infosys.tuwien.ac.at/~riva/Docs/Report1/Report1.html>