

제품 정보의 Web-기반 시각화

Web-based Visualization of Product Information

97. 11. 21

울산대학교

서윤호 정덕호

UOU IE

제품 및 공정 설계 연구실
PPD Lab

1

제품 정보 공유

- 제품 개발자 간의 체계적인 의사교환 및 업무 조정
- 설계 변화에 대한 정보의 불일치 요소 최소화
- 정보 교환의 미비로 인한 시행착오 최소화
- 어플리케이션 간 정보의 활용도 최대화
- 완제품과 설계와의 편차 최소화

UOU IE

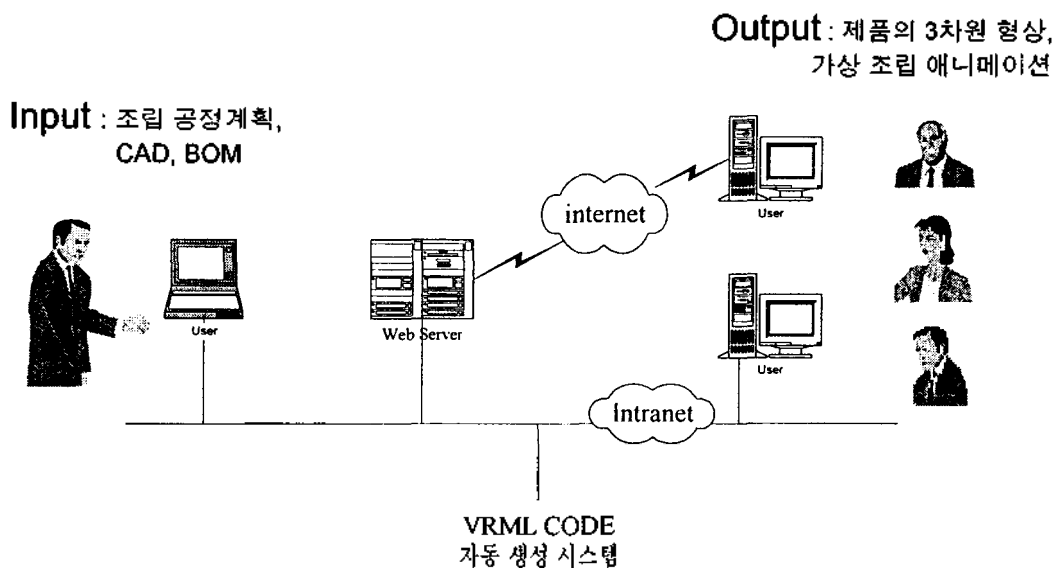
제품 및 공정 설계 연구실
PPD Lab

2

제품 정보란 ?

- **Electronic Documents**
(specifications, configurations, purchasing orders)
- **CAD**
(drawings, models, parts, assemblies)
- **CAE**
(engineering analysis)
- **Process Plan**
(assembly & machining)
- **BOM**
(product structure & material requirements)

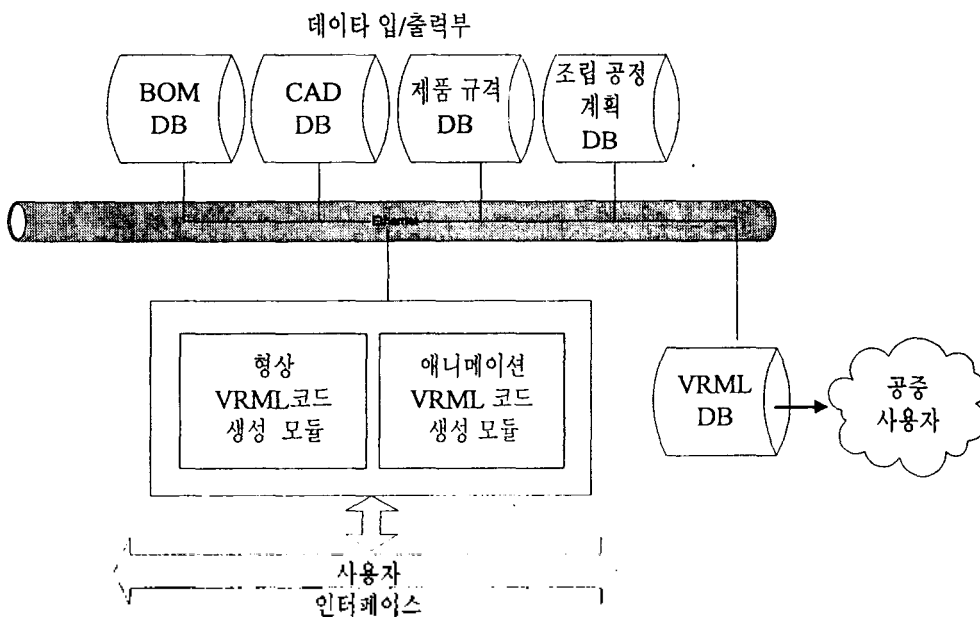
Web-based Visualization of Product Information



연구 목적

- CE구현을 위한 제품 정보 공유 도구 개발
- Web 상에서의 제품 정보 공유를 위한 VRML2.0 코드 자동 생성 시스템 개발
 - CAD 데이터의 동적인 3차원 형상 표현
 - BOM 의 제품 구성 표현
 - 제품의 조립 순서와 방법의 표현

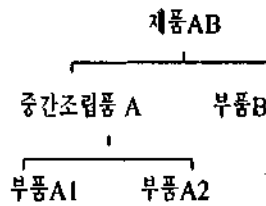
시스템 구성



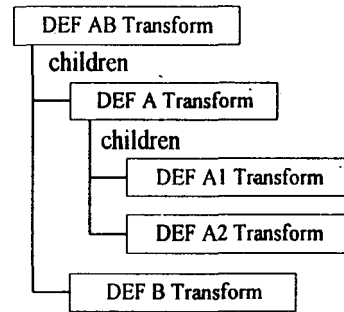
BOM 과 VRML의 관계

PART	PARENT PART
부품 A1	중간조립품 A
부품 A2	중간조립품 A
중간조립품 A	제품 A
부품 B	제품 A

BOM 데이터베이스의 구조



BOM Tree



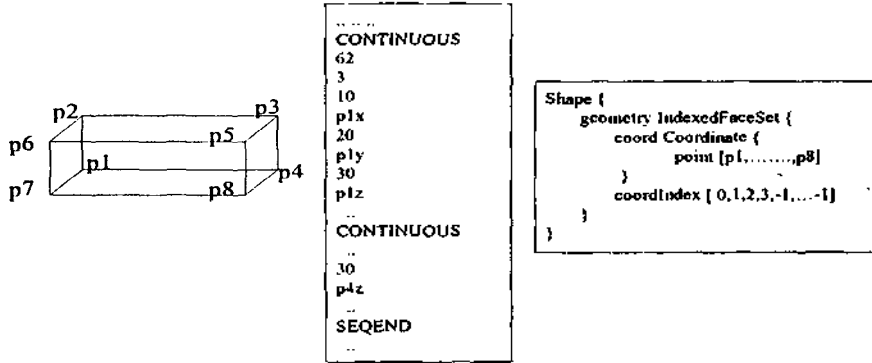
VRML Tree

BOM TO VRML 변환 알고리즘

```

part = VRML 코드를 생성하고자 하는 제품명;
position = 0;
List.AddTail(part);
while (position == List.GetCount())
{
    part = List.GetAt (position);
    List.AddTail (Query_BOM_Database (part));
    position ++;
}
while (List.GetCount () != 0)
{
    part = List.GetFirst ();
    if (part.parent == none) Make_Node (part);
    else Insert_Child_Node (Find_Parent_Position (part),part);
    List.DeleteFirst ();
}
    
```

DXF 와 VRML



형상

DXF

VRML

기업이 소유하고 있는 기존의 CAD 데이터를 이용하고 VRML Tree Leaf 노드의 VRML Code를 생성하기 위하여 Data Interchange File Format(*.DXF) 과 같은 중립형식 CAD 데이터를 이용하였다

UOU IE

제품 및 공정 설계 연구실
PPD Lab

9

DXF TO VRML 변환 알고리즘

```
dxf_index = DXF_Filter (dxl);
```

```
index_table = DXF_Coord_Filter (dxl_index);
```

```
point = IndexTable_To_VRML_PointSet (index_table);
```

```
coord_index = DXF_SEQEND_To_VRML_Index (index_table, dxl_index);
```

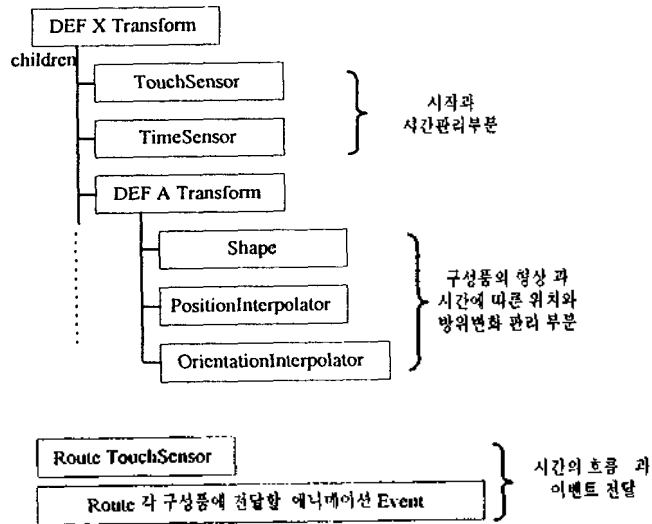
```
Make_Shape (point, coord_index);
```

UOU IE

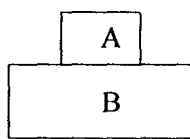
제품 및 공정 설계 연구실
PPD Lab

10

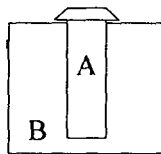
Animation VRML Structure



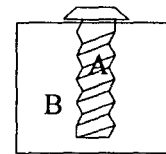
조립 관계 표현



Contacts (A,B)



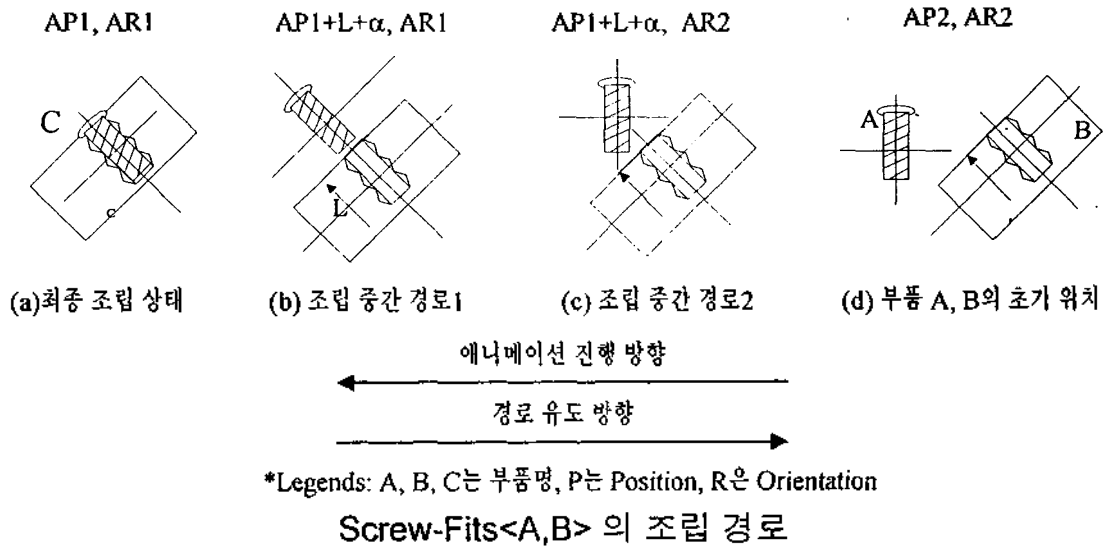
Fits (A,B)



Screw-fits (A,B)

- Contact (A, B): A와 B가 Contact관계에 있다는 것은 조립 시 A와 B가 Contact면을 기준으로 움직일 수 있다는 것을 의미한다.
- Fits (A,B): 두개의 부품 중 어느 하나는 오목, 다른 하나는 볼록 형태를 가진 부품의 조립형태를 갖는다.
- Screw-Fits (A, B): 두개의 부품 중 어느 하나는 오목, 다른 하나는 볼록 형태를 가진 부품으로 회전의 조립형태를 갖는다.

Animation 조립 경로 유도(1)



Animation 조립 경로 유도(2)

- Screw-Fits<A,B> 조립명령에 대한 부품의 A의 Position과 Orientation Interpolator의 변화

```

PositionInterpolator
{
  key[ t1 ]
  keyValue[AP2 ]
}
OrientationInterpolator
{
  key[ t1 ]
  keyValue[AR2 ]
}
  
```

초기 입력상태

```

PositionInterpolator
{
  key[t1,t2,t3,t4]
  keyValue[AP2, AP1+L+α, AP1+L+α, AP1 ]
}
OrientationInterpolator
{
  key[t1, t2, t3, t4 ]
  keyValue[AR2, AR2, AR1, AR1 ]
}
  
```

조립경로 추가

Animation 조립 경로 알고리즘

```

index = 0;
pos_ori[index++] = Read_VRML_Final()

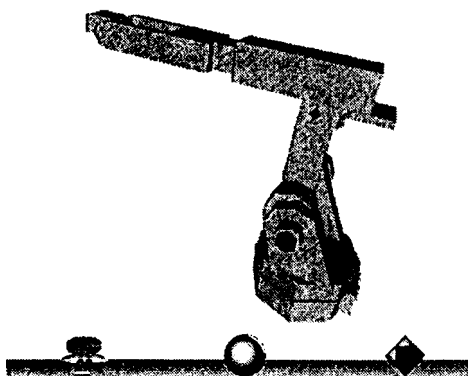
if(Read_Relation() == Contact)
    pos_ori[index++] = Cal_Contact(pos_ori [index-1], Read_Regulation());
if(Read_Relation() == Fit)
    pos_ori [index++ ] = Cal_Fit(pos_ori [index-1], Read_Regulation());
if(Read_Relation() == Screw_fit)
    pos_ori [index++] = Cal_Screw(pos_ori [index-1], Read_Regulation());

pos_ori[index++] = Calculate_Ori(pos_ori[index-1]);
pos_ori[index] = Read_VRML_Init();

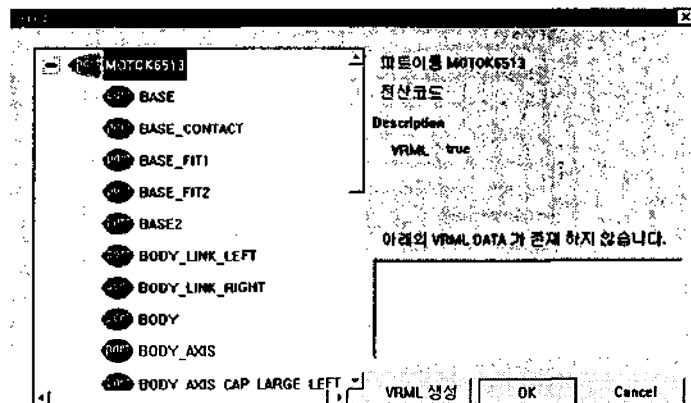
While(index>=0) {
    Add_P&O(pos_ori[index--],Time, t);
}
    
```

시스템 구현과 응용(1)

MOTOK6513 Robot 의 조립에 응용

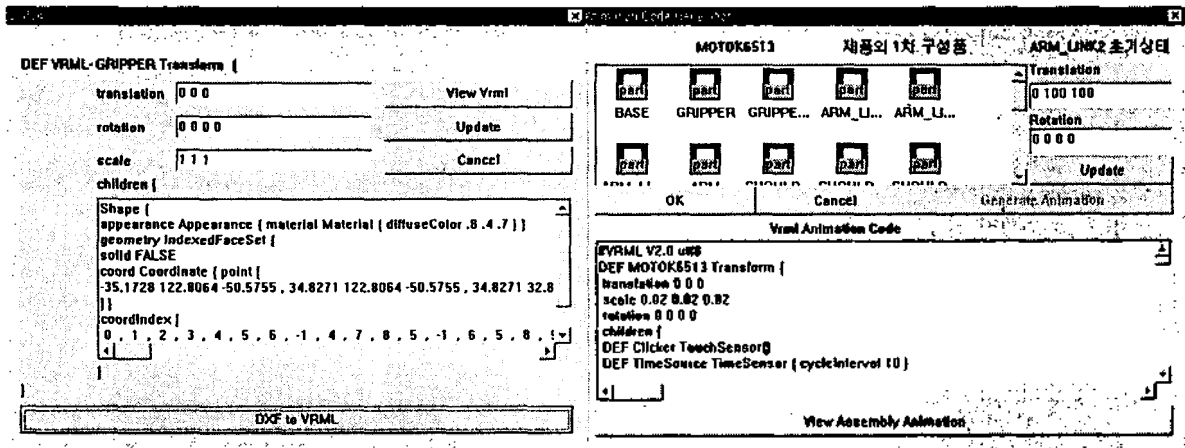


MOTOK6513



VRML Tree 구성

시스템 구현과 응용(2)



DXF 형상을 VRML 형상으로 변환

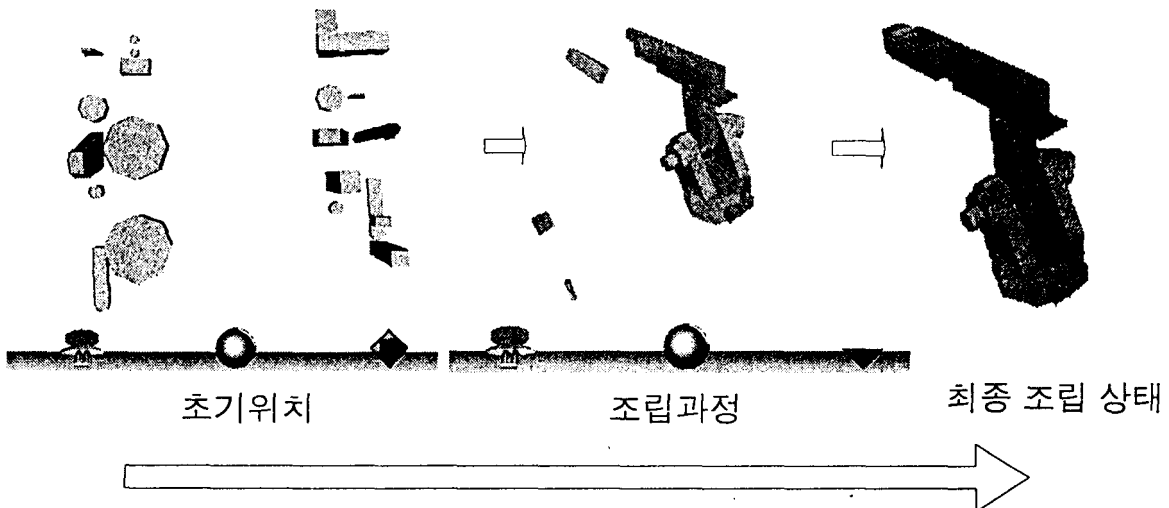
Animation VRML Code 생성 모듈

UOU IE

제품 및 공정 설계 연구실
PPD Lab

17

시스템 구현과 응용(3)



UOU IE

제품 및 공정 설계 연구실
PPD Lab

18