

STEP 기반의 조립정보 추출과 코드 생성 알고리즘

김광수*, 최후곤**

STEP-based Algorithms for Extraction and Code Generation of Assembly Information

Kim, K.-S.* and Choi, H.-G.**

ABSTRACT

The integration of CAD and CAM is critical for computer aided process planning. Recently, STEP has been emerged and utilized as a product data exchange standard format in CAD/CAM areas. The assembly information consists of assembly component relations, assembly features, assembly directions and mating conditions. The purpose of this study is to develop an efficient algorithm to extract assembly information directly from the STEP based files and to generate automatically assembly process plan from the extracted assembly features. The developed algorithms will generate assembly information codes for STEP(AICS). The results from this study can be a reasonable link between design and manufacturing for developing better CAAPP(Computer-Aided Assembly Process Planning) systems.

Key words : STEP, Assembly information, Assembly process planning, Assembly information codes

1. 서 론

제조업체에서 생산성 향상에 관한 문제 중 조립과 관련된 작업의 효율 문제는 매우 중요한 문제이며 어렵고 복잡한 문제로 알려져 왔다. 조립공정계획은 제품의 설계로부터 정보를 추출하고, 그 정보를 분석, 가공하여 조립작업에 필요한 조립순서 및 조립도구, 조립방법 등을 계획하는 일련의 과정이다¹⁾. 제조업체의 생산활동에 있어 필수 불가결한 작업인 조립작업의 효율향상을 위해 조립공정계획의 중요한 요소인 도면 정보에서 자동으로 조립공정계획 수립을 위한 정보를 추출할 수 있다면, 제품의 변화에 적절하게 적용할 수 있는 조립공정계획을 원활히 수립할 수 있고, 전체 조립관련 업무의 효율을 향상시킬 수 있을 것이다. 최근에는 이중 CAD 시스템간의 제품 데이터 교환과 공유를 지원하는 표준의 필요성이 증가하고 있으며, 이를 위하여 국제표준화기구(ISO)와 기타 연구기관에서는 STEP(STandard for the Exchange of Product model

data)의 개발과 응용 시스템의 개발 및 검증을 계속하고 있다. 조립공정계획을 위한 STEP의 대표적인 연구로 Sugimura *et al.*^{2,3)}는 기계조립품 표현을 위한 STEP의 데이터 구조를 제안하고, 이를 이용한 조립공정계획수립에 대하여 제안하였다. Zah and Du⁴⁾는 조립계획과 설계의 통합을 위한 제품설계 데이터의 모델링 방법을 제안하였다. Mok *et al.*⁵⁾은 STEP을 이용하여 조립정보를 추출하고 조립순서 계획을 수행하였으나 단순한 조립형상에 대하여 Z축 방향의 조립만을 고려하였다. STEP 파일을 이용한 특정형상 추출 연구들은 조립공정계획을 위한 것보다 가공특정형상에 관한 논문들이 대부분이었다⁶⁾. 본 연구에서는 CAD를 이용한 조립모델을 AP203 파일로 저장하고, STEP 파일을 해석하여 다양한 조립특정형상 정보와 기하, 위상 정보를 추출하고, 이를 바탕으로 X, Y, Z축에 대한 조립방향과 조립결합조건을 결정하는 알고리즘을 구현하는 것이다.

2. 조립정보 추출 알고리즘

조립정보의 추출 알고리즘은 Fig. 1과 같은 순서로 수행된다. STEP 파일을 입력받은 뒤 조립 형상정보

*정회원, 대우정보시스템(주) 컨설팅사업팀
**교신저자, 종신회원, 성균관대학교 시스템경영공학과
- 논문투고일: 2004. 05. 25
- 심사완료일: 2005. 08. 13

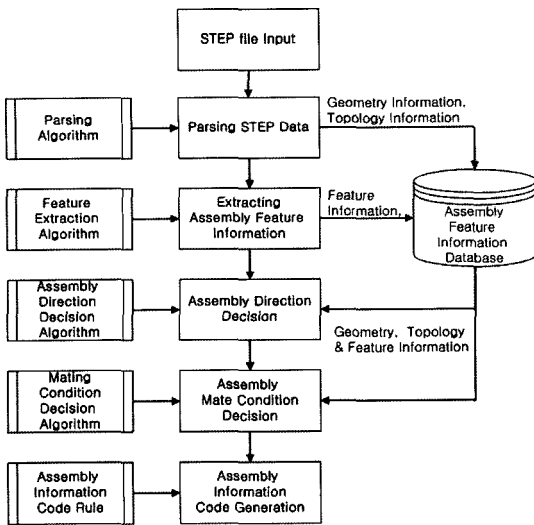


Fig. 1. 조립 정보 추출 순서.

추출과정을 통해서 나온 정보는 조립 형상정보 데이터베이스에 저장되고, 다시 이 정보를 활용하여 조립 방향과 결합조건을 결정한 후, 조립 정보 코드를 생성하게 된다.

2.1 STEP 파일 분석(parsing)

텍스트 형식의 STEP 파일에서 조립에 필요한 위상 정보와 기하정보를 추출하기 위한 분석절차는 Fig. 2 와 같다. Fig. 2에서 엔터티(entity)와 속성(attribute)은 AP203과 Part 42에 정의되었고, Part 21 정의를 따르고 있으며 여러 속성 중 조립정보 추출을 위해 필요한 속성의 순서와 이름, 데이터 형식(data type)을 나타내고 있다[8-10].

STEP 파일의 분석에 이용된 솔리드의 유형은 B-rep(boundary representation)의 정의를 이용하였다.

B-rep 모델은 면의 집합에 의해 경계를 이룬다는 위상학적 개념으로서 모델데이터는 서로 연결된 면(surface), 모서리(edge), 꼭지점(vertex)으로 구성된다[11].

STEP 파일을 분석하기 위해서는 먼저 조립품의 구성 부품들을 찾아야 한다. Fig. 2에서와 같이 구성부품은 next_assembly_component_usage_occurrence 엔터티에 그 정보를 담고 있고, 엔터티를 찾은 후 4, 5번째 속성 값을 찾아낸다. 그 중 relating_production_definition은 조립품의 개체(instance)번호이고, related_product_definition은 구성부품의 개체번호이다.

차례로 하위 엔터티(sub-entity)를 찾아가면 product 엔터티에서 부품과 구성품의 이름을 알 수 있다. Fig.

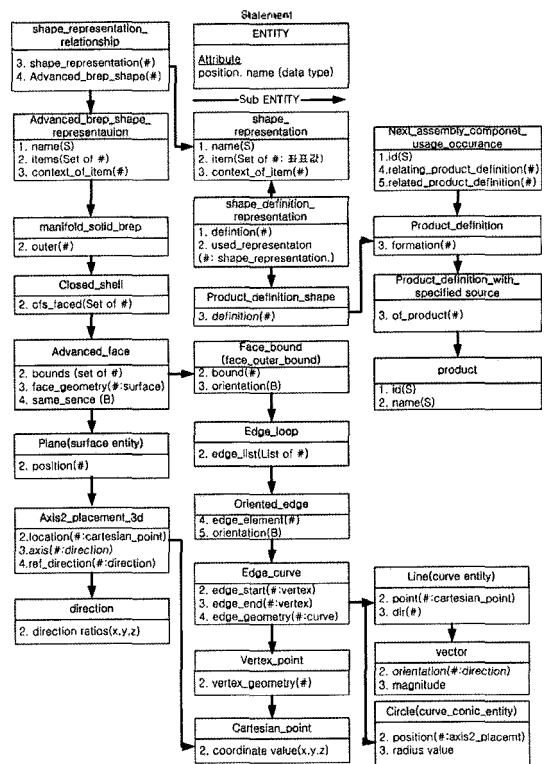


Fig. 2. STEP 파일 분석(parsing) 절차.

3은 조립품의 구성부품 정보를 나타내는 STEP 파일의 예이다.

```

#248 = MECHANICAL_CONTEXT ( 'NONE', #246, 'mechanical' ) ;
#249 = PRODUCT ( '14', 'assy_ext', ' ', ( #248 ) ) ;
.....

#256 = PRODUCT_DEFINITION ( ' ', 'UNSPECIFIED', #290, #255 ) ;
.....

#290 = PRODUCT_DEFINITION_1FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY',
'UNSPECIFIED', #249, 'NOT_KNOWN' ) ;
.....

#325 = NEXT_ASSEMBLY_USAGE_OCCURRENCE ( 'NAU01', ' ', ' ', #256, #2965, $ ) ;
.....

#351 = NEXT_ASSEMBLY_USAGE_OCCURRENCE ( 'NAU02', ' ', ' ', #256, #3047, $ ) ;
.....

#2955 = MECHANICAL_CONTEXT ( 'NONE', #2952, 'mechanical' ) ;
#2956 = PRODUCT ( '8', 'skin', ' ', ( #2955 ) ) ;
.....

#2965 = PRODUCT_DEFINITION ( ' ', 'UNSPECIFIED', #2998, #2964 ) ;
.....

#2998 = PRODUCT_DEFINITION_1FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY',
'UNSPECIFIED', #2956, 'NOT_KNOWN' ) ;
.....
    
```

Fig. 3. 부품정의 STEP 파일의 예.

AP203 파일에서는 조립 구성품들의 기하정보들이 각각 개별 모델 좌표계에 의해 정의된 값으로 저장되어 있고, 기하정보들을 최종 조립작업에 대한 조립 좌표계의 일치시키기 위해서는 변환 행렬을 사용해야 한다. 변환 행렬 좌표값에 대한 정보는 shape_representation 엔터티에 저장되어 있다.

```
#2087 = SHAPE_REPRESENTATION ('ass_ex', { #2133, #2096,
#2117, #2100, #2104, #2108, #2113, #2121 }, #2132 );

#2117 = AXIS2_PLACEMENT_3D ('NONE', #2118, #2119, #2120 );
#2118 = CARTESIAN_POINT ('NONE', ( 100.0, 0.0, 110.0 ));
#2119 = DIRECTION ('NONE', { 0, 0, -1.0 });
#2120 = DIRECTION ('NONE', (-1.0, 0, 0 ));
```

Fig. 4. STEP에서 조립 좌표계 정보.

Fig. 4는 'ass_ex' 조립품의 구성품에 대한 변환 행렬 좌표값을 보여 주고 있다. 첫 번째 #2133은 조립 좌표계의 원점 값이고 나머지가 차례대로 조립 좌표계에 대한 구성품들의 좌표값이다. #2117 부품의 예를 보면 #2118은 조립 후 좌표값이고, #2119는 Z방향, #2120은 X방향 값을 의미한다. Y방향값은 오른손 법칙에 의해 구해질 수 있다. 그러므로 위의 예는 식 (1)과 같은 변환 행렬 값을 구할 수 있다.

$$[T] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 100 & 0 & 110 & 1 \end{bmatrix} \quad (1)$$

Fig. 1에서와 같이 조립형상 정보 데이터베이스에 입력되는 기하정보 및 위상정보 값은 변환 행렬 계산 후에 입력되게 된다. 조립품의 구성 부품들을 찾을 때에는 각 구성 부품들의 조립특징형상, 조립 방향, 체결 조건(mating condition) 등을 찾는다.

2.2 조립특징형상 정보 추출

STEP의 조립 정보 추출을 하기 위하여 각 구성품의 조립특징형상을 추출하여야 한다. STEP에서 조립특징형상 정보를 추출하기 위해서 Fig. 2의 advanced_brep_shape_representation 엔티티를 시작으로 부품의 기하 및 위상 정보를 찾아내어 특징형상을 추출한다. 모든 개별 부품들에 대하여 advanced_brep_shape_representation에서부터 추출 알고리즘은 시작되며, close_shell 엔티티는 셀을 구성하는 다면체의 면을 가리키는 것으로 cfs_facedd의 개체번호의 개수가 부품의 면의 개수가 된다.

2.2.1 평면(plane)

평면은 특별한 조립특징형상을 갖고 있지 않으며, 하나의 연결된 모서리 루프(loop)에 의해 경계를 이룬다. 평면은 특별한 조립특징형상이 없는 다면체(polyhedra)와 원기둥(cylinder)로 구분할 수 있다. Fig. 5는 형상 정보에 대한 STEP 파일의 예이다.

```
#628 = ADVANCED_FACE ('NONE', { #2505, #2506, T. });
#968 = CLOSED_SHELL ('NONE', { #3203, #628, #984, #2145 });
#984 = ADVANCED_FACE ('NONE', { #1587, #1598, T. });
#1141 = FACE_OUTER_BOUND ('NONE', #1116, T. );
#1142 = PLANE ('NONE', #1143 );
#1143 = AXIS2_PLACEMENT_3D ('NONE', #1144, #1147, #1148 );
#1587 = FACE_OUTER_BOUND ('NONE', #2491, T. );
#1598 = CYLINDRICAL_SURFACE ('NONE', #1599, 0.005);
#1599 = AXIS2_PLACEMENT_3D ('NONE', #1600, #1601, #1602 );
#1603 = FACE_OUTER_BOUND ('NONE', #3130, T. );
#1604 = PLANE ('NONE', #1605 );
#1605 = AXIS2_PLACEMENT_3D ('NONE', #1606, #1607, #1608 );
#2145 = ADVANCED_FACE ('NONE', { #1603, #1604, T. });
#2505 = FACE_OUTER_BOUND ('NONE', #2810, T. );
#2506 = CYLINDRICAL_SURFACE ('NONE', #2507, 0.005);
#2507 = AXIS2_PLACEMENT_3D ('NONE', #2508, #2510, #2511 );
#3203 = ADVANCED_FACE ('NONE', { #1141, #1142, F. });
```

Fig. 5. STEP에서 원기둥 형상 정보.

평면의 조립특징형상에 대한 추출 조건으로 조건 1은 평면 정의를 위한 충분조건이고 조건 2는 원기둥 정의를 위한 필요충분조건이며, 조립특징형상추출 과정은 다음과 같다.

조건1. $\forall f_i \in F, f_i = \{l_k | \forall l_k \in L\}, n(l_k) = 1$

임의의 면과 면을 구성하는 루프의 개수가 1개이다.

조건2. $\forall s_i \in S, s_i = \{f_k | \forall f_k \in F\},$
 $[n(f_k) = 4] \wedge [n\{f_{kgeo} = cylinder\} = 2]$

임의의 셀을 구성하는 면의 개수가 4개이고, 그 중 2개 면의 형상이 원이다.

Procedure plane or cylinder (closed_shell)

for all advanced_face
 if $n(\text{advanced_face.bound}) - 1$
 assembly feature = plane
 if $\{n(\text{closed_shell.cfs_faced}) = 4\}$
 and $n(\text{advanced_face.face_geometry} = \text{cylindrical surface}) = 2\}$ then
 assembly feature = cylinder

End Procedure

Fig. 5에서 #968의 closed_shell의 속성값 중, #628, #984의 최종 face_geometry의 값이 cylindrical_surface이므로 원기둥임을 알 수 있다.

2.2.2 Slot과 Step

Slot과 Step은 간단한 평면의 종류로서 여러 면을

연결시켜 형상을 표현하게 되며, 조건 3은 충분조건을 정의한 것이다.

조건 3. $\forall s_i \in S, \forall f_j \in F, \forall l_k \in L, \forall e_m \in E,$

$$s_i = \{f_j | \forall f_j \in F\}, f_j = \{l_k | \forall l_k \in L\},$$

$$l_k = \{e_m | \forall e_m \in E\}, [n(f_j) \geq 8] \wedge [\{n(e_m) \geq 6\}]$$

임의의 셀을 구성하는 면의 개수가 8개 이상이고, 그중 face를 구성하는 edge의 개수가 6개 이상인 면이 존재한다.

Fig. 6은 slot형상정보의 예이고, 위의 충분조건을 만족하면 다음의 알고리즘으로 slot과 step으로 구분할 수 있다.

Procedure slot or step (closed_shell)

if closed_shell.cfs_faced < 8 Exit Procedure
for all advanced_face

if(advanced_face.bounds = face_outer_bound) and
(face_outer_bound->edge_loop.edge_list >= 6)
direction = advanced_face->plane->axis2.axis

//axis2.axis = (0,1,0)이면 +y 방향

case direction = +y

x_i = edge_loop->cartegian_point.x

z_i = edge_loop->cartegian_point.z

i=1,2,...n

for all x_i, z_i

{x_min = min(x_i), x_max = max(x_i)}

z_min = min(z_i), z_max = max(z_i)}

if (x_min, z_min < x_i, z_i < x_max, z_max)

x_j = x_i, z_j = z_i j=1,2,...m

if ((x_j, z_j) = edge_curv.edge_start) and

((x_j, z_j) = edge_curv.edge_end)

feature = slot

else feature = step

if(new advanced_face->direction -direction)

and (new feature = feature)

assembly feature = through slot(step)

else assembly feature = blind slot(step)

End Procedure

Fig. 6에서 #182의 좌표값이 (0,1,0)이므로 X, Z 평면이다. X의 좌표값 중 최대는 0.089, 최소는 -0.07이며, Z 값의 최대 최소는 0.05, 0.00이므로 그

```
#10 = VERTEX_POINT ('NONE', #85);
#85 = CARTESIAN_POINT ('NONE', (0.000, -0.070, 0.030));
.....
#87 = CARTESIAN_POINT ('NONE', (-0.070, -0.070, 0.050));
.....
#109 = CARTESIAN_POINT ('NONE', (0.089, -0.070, 0.000));
.....
#114 = CARTESIAN_POINT ('NONE', (0.040, -0.070, 0.050));
.....
#131 = VERTEX_POINT ('NONE', #133);
#133 = CARTESIAN_POINT ('NONE', (0.040, -0.070, 0.030));
.....
#178 = FACE_OUTER_BOUND ('NONE', #253, .T.);
#179 = PLANE ('NONE', #180);
#180 = AXIS2_PLACEMENT_3D ('NONE', #181, #182, #183);
#181 = CARTESIAN_POINT ('NONE', (-0.070, -0.070, 0.050));
#182 = DIRECTION ('NONE', (0.000, 1.000, 0.000));
#183 = DIRECTION ('NONE', (0.000, 0.000, 1.000));
#184 = CARTESIAN_POINT ('NONE', (0.000, -0.070, 0.050));
.....
#202 = CARTESIAN_POINT ('NONE', (-0.070, -0.070, 0.000));
#203 = ADVANCED_FACE ('NONE', (#178), #179, F.);
#204 = VERTEX_POINT ('NONE', #184);
.....
#209 = CARTESIAN_POINT ('NONE', (0.089, -0.070, 0.050));
.....
#218 = EDGE_CURVE ('NONE', #131, #336, #191, .T.);
.....
#253 = EDGE_LOOP ('NONE', (#331, #317, #60, #314, #347, #310, #150, #8));
.....
#324 = EDGE_LOOP ('NONE', (#359, #186, #11, #362, #343, #289, #351, #311));
.....
#336 = EDGE_CURVE ('NONE', #359, #80, #110, .T.);
#338 = VERTEX_POINT ('NONE', #114);
#346 = EDGE_CURVE ('NONE', #10, #204, #153, .T.);
#347 = ORIENTED_EDGE ('NONE', +, +, #367, .T.);
.....
#355 = CLOSED_SHELL ('NONE', (#348, #316, #349, #203, #334, #309, #341, #345,
#132, #7));
#360 = EDGE_CURVE ('NONE', #10, #131, #15, .T.);
```

Fig. 6. STEP에서 slot 형상 정보.

사이의 존재하는 점(vertex)은 #10번과 #131번이고, 그 점들을 연결하는 edge #360이 존재한다. 이 점들을 공유하는 다른 edge는 #218번과 #346으로서 다른 점들 #336과 #204번의 좌표값은 각각 Z값만 증가함을 알 수 있다. 그러므로 Fig. 7과 같이 조립가능 방향이 +Z 방향의 slot 조립특징형상을 알 수 있다.

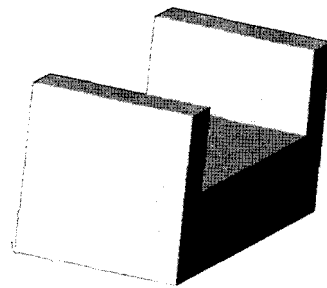


Fig. 7. Slot 형상.

2.2.3 Through hole과 Through pocket

Through hole과 Through pocket은 면이 둘 이상의 루프에 의해 경계를 이루며 바깥쪽 루프와 안쪽 루프들을 가진다. 조건 4는 충분조건을 정의한 것이다.

조건 4. $\forall f_i, f_k \in F, \forall l_j \in L$

$$f_i = \{l_k | \forall l_k \in L\}, f_j = \{l_s | \forall l_s \in L\}$$

$$[\{n(l_k) \geq 2\}] \wedge [(\exists f, dir) = (\exists f, dir)]$$

입의의 면에 대한 loop의 개수가 2개 이상이고 방향값이 같은 면이 두개가 존재한다.

Fig. 8은 through hole형상 정보의 예이고 추출 알고리즘은 다음과 같다.

Procedure through hole or through pocket (closed_shell)

```
for all advanced_face
  if {n(advanced_face.bounds) ≥ 2}
  and n(advanced_face->direction x,y,z)=2}
  if {(face_bound->edge_loop.edge_list)=2}
  and {edge_curve.edge_geometry=circle}
  assembly feature = through hole
  if {(face_bound->edge_loop.edge_list)>3}
  assembly feature = through pocket
```

End Procedure

Fig. 8에서 #2072의 edge의 개수가 2개인 through hole과 #2286에서 edge의 개수가 4개인 through pocket이 함께 있는 복합 특징형상이며 Fig. 9와 같다.

```
#81 = FACE_BOUND ('NONE', #2286, .T.);
#82 = FACE_BOUND ('NONE', #2072, .T.);
#83 = FACE_BOUND ('NONE', #376, .T.);
#84 = FACE_OUTER_BOUND ('NONE', #2101, .T.);
#85 = PLANE ('NONE', #86);
#86 = AXIS2_PLACEMENT_3D ('NONE', #87, #88, #89);
#87 = CARTESIAN_POINT ('NONE', (-0.050, -0.050, 0.0299));
#88 = DIRECTION ('NONE', (0.000, 0.000, 1.000));
#89 = DIRECTION ('NONE', (1.000, 0.000, 0.000));
.....
#307 = ORIENTED_EDGE ('NONE', ., ., #1970, .T.);
.....
#311 = PLANE ('NONE', #312);
#312 = AXIS2_PLACEMENT_3D ('NONE', #313, #314, #315);
#313 = CARTESIAN_POINT ('NONE', (-0.050, -0.050, 0.000));
#314 = DIRECTION ('NONE', (0.000, 0.000, 1.000));
#315 = DIRECTION ('NONE', (1.000, 0.000, 0.000));
.....
#474 = CIRCLE ('NONE', #475, 0.00499999999999997500);
.....
#1894 = ADVANCED_FACE ('NONE', (#81, #82, #83, #84), #85, .T.);
.....
#1928 = ADVANCED_FACE ('NONE', (#306, #308, #309, #310), #311, .F.);
.....
#1970 = EDGE_CURVE ('NONE', #1987, #1885, #474, .T.);
.....
#2072 = EDGE_LOOP ('NONE', (#307, #3271));
.....
#2286 = EDGE_LOOP ('NONE', (#2298, #2350, #2477, #2937));
```

Fig. 8. STEP에서 through hole형상 정보.

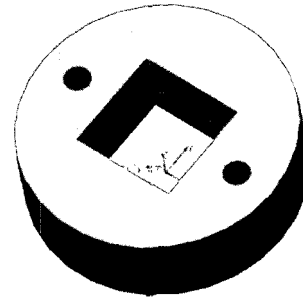


Fig. 9. Through hole과 Through pocket 형상.

2.2.4 Boss와 Blind

Boss와 Blind 특징형상은 면이 둘 이상의 루프에 의해 경계를 이루며 외부 루프와 내부 루프들을 가질 수 있으며, 조건 5는 충분조건을 정의한 것이다.

조건 5.

$$\forall f_i \in F, \forall l_j \in L, [\exists f_i = \{l_j | l_j \in L\}, n(l_j) \geq 2]$$

입의의 면에 대한 loop의 개수가 2개 이상이 존재한다.

Fig. 10은 STEP 파일의 예이며, 다음과 같은 알고리즘으로 조립특징형상을 추출할 수 있다.

Procedure through hole or through pocket (closed_shell)

```
for all advanced_face
  if {n(advanced_face.bounds) ≥ 2} //충분조건
  direction = advanced_face->axis2.axis
  //axis2.axis = (0,0,1)이면 +Z 방향
  case direction +z
  x_i = edge_loop->vertex_point->cartegian_point.x
  y_i = edge_loop->vertex_point->cartegian_point.y
  i=1,2,...n
  for all x_i, y_i
    if (new x_i = x_i) and (new y_i=y_i)
    and (new z_i ≠ z_i)
    if new z_i > z_i feature = boss
    else feature = blind
  if {(face_bound->edge_loop.edge_list)=2}
  and {edge_curve.edge_geometry=circle}
  assembly feature = cylinder boss (blind hole)
  if {(face_bound->edge_loop.edge_list)≥3}
  assembly feature=prismatic boss(blind pocket)
```

End Procedure

```

#9 = VERTEX_POINT ('NONE', #75);
.....
#12 = ORIENTED_EDGE ('NONE', *, *, #154, .F.);
.....
#15 = ORIENTED_EDGE ('NONE', *, *, #40, .F.);
.....
#21 = ADVANCED_FACE ('NONE', ( #103, #104), #105, .T.);
#22 = VERTEX_POINT ('NONE', #110);
.....
#26 = VERTEX_POINT ('NONE', #116);
.....
#29 = EDGE_CURVE ('NONE', #22, #26, #117, .T.);
.....
#40 = EDGE_CURVE ('NONE', #9, #254, #140, .T.);
.....
#75 = CARTESIAN_POINT ('NONE', (0.029, 3.670E-018, 0.020));
.....
#102 = CARTESIAN_POINT ('NONE', (-0.029, 0.000, 0.020));
#103 = FACE_OUTER_BOUND ('NONE', #250, .T.);
#104 = FACE_BOUND ('NONE', #255, .T.);
#105 = PLANE ('NONE', #106);
#106 = AXIS2_PLACEMENT_3D ('NONE', #107, #108, #109);
#107 = CARTESIAN_POINT ('NONE', (0.000, 0.000, 0.020));
#108 = DIRECTION ('NONE', (0.00, 0.00, 1.00));
.....
#110 = CARTESIAN_POINT ('NONE', (0.029, 3.67E-018, 0.059));
.....
#116 = CARTESIAN_POINT ('NONE', (-0.029, 0.000, 0.059));
.....
#154 = EDGE_CURVE ('NONE', #254, #9, #64, .T.);
.....
#254 = VERTEX_POINT ('NONE', #102);
#255 = EDGE_LOOP ('NONE', ( #12, #15 ));

```

Fig. 10. STEP에서 cylindrical boss형상 정보

Fig. 10의 예에서 #21의 advanced_face는 bound의 값이 두 개이므로 루프가 두개가 존재함을 알 수 있고, #108의 방향값이 (0,0,1)이므로 +Z방향이 조립가능 방향이다. #104 face_bound의 #255 edge_loop의 개수가 2개이므로 cylinder 형상임을 알 수 있고, 구성 꼭지점은 #75, #102이다. 이 꼭지점들을 구성요소로 하고 있는 다른 edge_curve를 구하여 구성요소인 다른 꼭지점들을 구하면 #110, #116로 각각 #75, #102와 비교하였을 때 +0.039 큰 것을 알 수 있다. #107이 +Z 방향이므로 0.049 만큼 +Z 방향으로 뺄출된 원통 boss임을 알 수 있다.

위의 조립정보 추출 알고리즘들은 Fig. 2의 분석 절차에 따라 각 꼭지점의 좌표 값과 형상 정의를 데이터 베이스에 저장하고, 모든 부품의 각 면에 대해 구현하여야 한다.

2.3 조립 방향 및 결합 조건 정보

본 연구에서 조립 방향은 -Z 방향, ±X, ±Y 방향을 고려하였다. 부품이 조립되는 방향은 부품별 조립 특징형상에 대한 추출과정에서 얻은 기하정보를 바탕으로, 조립가능 방향 값을 구하고 저장하였다. 또한 결합조건으로 끼워맞춤, 약한 접촉(against), 강한 접촉(contact)을 고려하였다. 끼워맞춤은 결합되는 두 부품의 중심축이 일치하는 원통형 특징형상에 적용된다. 약한 접촉은 두 특징형상의 결합면이 붙어 있는 상태를 의미하며, 각주형 특징형상 및 일반 평면에 적

용 된다. 강한 접촉은 약한 접촉 조건하에 회선과 이동이 없는 것을 의미한다. 특히 끼워맞춤과 강한 접촉을 하는 특징형상들은 내부루프가 존재함을 알 수 있다. 그러므로 다음과 같은 알고리즘으로 결합조건을 추출한다.

단계 1. 부품마다 조립특징형상 추출 단계에서 추출한 조립특징형상과 방향별 결합조건에 대한 조립가능 방향을 구한다.

단계 2. 조립품의 구성 부품들을 찾은 후에는 각 구성 부품들의 조립특징형상, 조립 방향, 체결 조건 등을 찾는다. 조립작업에 대한 조립 좌표계와 일치하기 위해서 변환 행렬 계산 후 다음 규칙에 따라 구성 꼭지점의 값의 일치여부 및 조건에 의해서 결합조건을 정의한다. 먼저 부품별로 내부루프에 해당하는 face_bound 엔터티의 edge_loop들을 검출하며 다음 규칙을 따른다.

규칙 1: 부품의 face_bound의 edge_loop의 구성 꼭지점의 값과 일치하는 edge_loop가 존재하고, 꼭지점의 개수가 2개이고 지름이 같은 경우 원통형으로서 강한 끼워맞춤(tight fit)으로 정의하고, 꼭지점의 개수가 3개 이상이면 강한 접촉으로 정의한다.

규칙 2: 두 edge_loop 중 한 edge_loop의 구성 꼭지점이 다른 edge_loop의 구성 꼭지점 내부에 포함된 경우 꼭지점 개수가 2개이고 지름이 서로 다를 경우 약한 끼워맞춤(clearance fit)으로 정의하고, 꼭지점 개수가 3개 이상이면 약한 접촉으로 정의한다.

단계 3: 단계 2에서 정의된 face_bound의 면들을 제외한 나머지 면들에 대해 외부루프에 해당하는 face_outer_bound의 엔터티의 edge_loop들을 검출하며 다음 규칙을 따른다.

규칙 3: 두 edge loop 중 한 edge_loop의 구성 꼭지점이 다른 edge_loop의 구성 꼭지점과 같은 경우 스텝(step), 슬롯(slot) 형상은 강한 접촉, 평면(plane)은 약한 접촉으로 정의하고, 내부에 포함되었으면 약한 접촉으로 정의한다.

단계 4: 단계 2와 단계 3에서 두 부품의 방향별 결합조건이 정의되면 특징형상에 따라 조

립 방향을 설정 한다. 기준부품과 연결 부 품은 각 부품의 최소 Z 좌표값이 작은 부 품을 기준부품으로 하고 기준부품과 연결 되어 있는 부품들을 차례대로 추적하여 코 드를 부여할 수 있다. 최소 Z 좌표값을 갖는 부품이 중복되어 있을 경우에는 X 좌표값이나 Y 좌표값이 원점과 가까운 부 품을 기준부품으로 우선한다. 끼워맞춤은 원통형 특징형상의 원통면에 대한 법선벡 터 방향이 조립방향이 되고, 접촉은 일치 한 면의 법선 벡터 방향이 조립방향이 된 다. 각 부품별 조립방향은 연결부품을 기 준으로 정의한다.

단계 5: 단계 4에서 설정된 부품에 대한 조립특징 형상별 조립가능 방향은 모든 부품들에 대 해서 단계 2와 단계 5를 반복한다.

2.4 조립체결 정보 코드 생성

조립정보 추출알고리즘에 의해 구한 두 부품간의 조립정보를 표현해주기 위하여 STEP을 위한 조립정보 코드(AICS: Assembly Information Code for STEP)를 정의 하였다. AICS의 코드 내용은 Table 1 과 같다.

Table 1. AICS 정의

코드자리	의미	코드값
Digit 1	첫 번째 부품의 조립특징형상 정보	0 : plane 1: cylinder 2 : through pocket 3 : through hole
Digit 2	두 번째 부품의 조립특징형상 정보	4 : prismatic boss, 5 : cylinder boss 6 : blind pocket 7 : blind hole 8 : step 9 : slot
Digit 3	조립 방향 정보	0: -Z 1: -X 2: +X 3: -Y 4: +Y
Digit 4	결합조건정보	t : tight fit f: clearance fit c: contact a: against
Extend Code 1	기준부품 정보	STEP 파일에서 product 엔터티 번호
Extend Code 2	연결부품 정보	

3. 사례 연구

조립정보 추출 알고리즘을 적용하기 위하여 Fig. 11 과 같은 조립 설계를 SolidWorks[®]2003으로 구현한 후, STEP 파일로 변환하였다.

알고리즘의 예제로 사용되었던 Fig. 3에서 Fig. 10 까지는 Fig. 11에 대한 예제 STEP 파일이다. STEP 파일에 대한 분석은 ST-Developer[®]9.0을 이용하였고, 구현된 프로그램은 Microsoft Visual Basic[®]6.0과 데이터의 처리를 위해 Microsoft[®]Access2002를 사용하여 개발하였다.

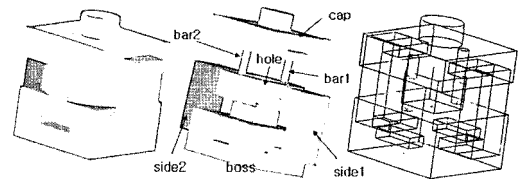
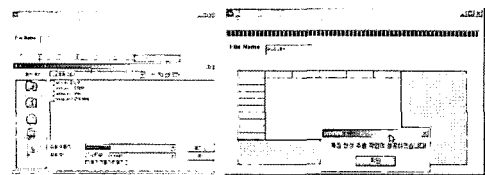


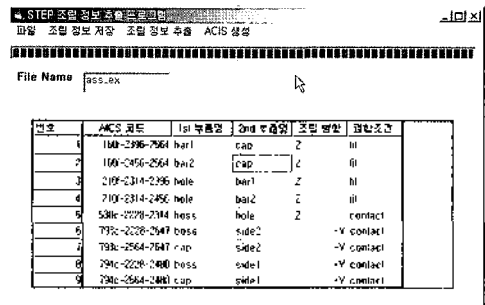
Fig. 11. 예제 모델링(조립도, 분해도, B-rep).

```
#2228 = PRODUCT ('boss', 'boss', . ( #2227 ) ) :
.....
#2314 = PRODUCT ('hole', 'hole', . ( #2313 ) ) :
.....
#2396 = PRODUCT ('bar1', 'bar1', . ( #2395 ) ) :
.....
#2480 = PRODUCT ('side1', 'side1', . ( #2479 ) ) :
.....
#2456 = PRODUCT ('bar2', 'bar2', . ( #2521 ) ) :
.....
#2564 = PRODUCT ('cap', 'cap', . ( #2563 ) ) :
.....
#2647 = PRODUCT ('side2', 'side2', . ( #2646 ) ) :
```

Fig. 12. STEP 파일의 product 엔터티.



(a) STEP 파일 선택



(c) AICS 결과 값

Fig. 13. 조립정보 추출 프로그램.

Fig. 11에 대한 STEP 파일 product 엔터티의 값이 Fig. 12와 같을 때 알고리즘을 통해 구현된 결과는 Fig. 13과 같다.

Fig. 12와 Fig. 13에서 첫 번째 AICS 코드 값을 살펴보면 160f-2396-2564에 대한 조립정보를 알 수 있다. #2396의 부품명은 bar1으로 cylinder 형상이고, #2564의 부품명은 cap으로 blind pocket 형상이며, -Z방향으로 약한 끼워맞춤(clearance fit)으로 조립됨을 알 수 있다.

4. 결 론

STEP은 제품의 형상정보 뿐만 아니라, 제품의 전수명 주기에 걸친 모든 데이터를 포함하고 있으며, 제품 정보를 어느 시스템에서나 사용할 수 있도록 하는 중립 형태의 표준으로 현재 다양한 산업분야에서 활용되고 있는 기술이며 많은 연구들이 활발히 진행되고 있다. CAD를 이용한 도면 정보에서 자동으로 조립공정계획을 위한 정보를 추출할 수 있다면, 전체 조립관련 업무의 효율성을 높일 수 있을 것이다. 본 연구에서는 STEP 파일에서 조립공정계획을 위한 조립정보 자동추출 알고리즘을 개발하였고, STEP을 위한 조립정보 코드를 생성하였다. 특히, 조립정보 추출과정에서 복잡한 계산과정 없이 단순히 위상, 기하학적인 정보만으로 조립정보를 생성하고 추출하는 알고리즘을 개발하였다. 조립정보를 정리한 조립 정보 코드는 조립공정계획에 중요한 입력 자료로 활용될 수 있다. 추후 연구과제로는 본 연구에서는 나사선 모양의 부품과 자유 곡선으로 표현되는 부품의 조립특징형상은 추출하지 못하였기 때문에 다양한 조립부품들에 특징형상을 추출할 수 있는 알고리즘의 개발이 요구된다. STEP 파일에서 직교 좌표 방향이 아닌 다양한 방향의 조립과 공차정보를 활용하여 조립정보를 추출하는 알고리즘이 필요하다.

참고문헌

1. 어명구, 최후근, 김광수, "조립BOM 생성을 위한 병렬순서 추출 알고리즘", 대한산업공학회지, Vol. 29, No. 1, pp. 49-64, 2003.
2. Sugimura, N., Moriwaki, T., Kita, T., Ohtaka, A., Hiraoka, H., Kitajima, K. and Kuroiwa, T., "A Study on Product Model for Design and Analysis of Mechanical Assemblies Based on STEP", *Transactions of the Institute of Systems, Control and Information Engineers*, Vol. 8, No. 9, pp. 466-473, 1995.
3. Sugimura, N., Moriwaki, T. and Kakino, T., "A Study on Assembly Model based on STEP and Its Application to Assembly Process Planning", *Proceedings of ASME Japan/USA Symposium on Flexible Automation*, pp. 791-794, 1996.
4. Zah, X. F. and Du, H., "A PDES/STEP-based Model and System for Concurrent Integrated Design and Assembly Planning", *Computer-Aided Design*, Vol. 34, pp. 1087-1110, 2002.
5. Mok, S. M., Ong, K. and Wu, C. H., "Automatic Generation of Assembly Instructions using STEP", *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, pp. 313-318, 2001.
6. 강무진, "STEP AP224를 이용한 특징형상의 가공순서 계획", 한국 CAD/CAM학회 논문집, 제9권, 제2호, pp. 175-182, 2004.
7. 손정모, 유상룡, 김영호, 이수홍, "분산환경에서 표준제품 모델(STEP)을 이용한 내용검색", 한국 CAD/CAM 학회 논문집, 제4권, 제4호, pp. 285-294, 1999.
8. ISO 10303-21, Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure, 1994.
9. ISO 10303-42, Part 42 : Product Data Representation and Exchange, 1997.
10. ISO 10303-203, Part 203: Application Protocol: Configuration Controlled Design, 1994.
11. Zeid, I., *CAD/CAM Theory and Practice*, McGraw-Hill, Inc., 1991.



김 광 수

1996년 서울산업대학교 산업공학과 학사
 1998년 성균관대학교 산업공학과 석사
 2004년 성균관대학교 산업공학과 박사
 2004년~현재 대우정보시스템(주) 건설팀
 사업팀
 관심분야: CAD/CAM, CAPP, STEP,
 ERP, 6시그마



최 후 곤

1975년 서울대학교 공과대학 산업공학과
 학사
 1979년 서울대학교 대학원 산업공학과
 석사
 1981년 Iowa State University, Dept of
 Industrial and Manufacturing
 Systems Engineering, 석사
 1985년 Iowa State University, Dept of Industrial and Manu-
 facturing Systems Engineering, 박사
 1985년~1989년 Montana State University, Dept of Industrial
 and Management Engineering, 조교수
 1989년~현재 성균관대학교 시스템경영공학과, 교수
 관심분야: 공정계획 알고리즘 개발, 조립순서 생성, machinability/
 assemblability/producibility 향상을 위한 제품 및 공정개발