한국 CAD/CAM 학회 논문집
제6권 제1호 2001년 3월 pp. 48-58

STEP을 이용한 신속조형용 설계정보 변환체계

이병열*, 지해성**

An RP Data Exchange Model Based on STEP

Lee, B.* and Jee, H.**

ABSTRACT

One of the biggest problems of rapid prototyping(RP) technologies lies in their standard file format for CAD data exchange. Current methods using the *de facto* industry standard 'STL' have at times resulted in problems such as accuracy, redundancy, and integrity. In this paper we propose a STEP based data exchange framework for rapid prototyping systems. In this paradigm of data exchange, STEP models can be imported and converted into faceted B-rep. solid models for visualization and 2-D layer data for RP. Also an STL model, on the other hand, can be converted into a faceted B-rep. STEP model and exported as a new data exchange model with RP information.

Key words : STEP, CAD, RP, STL

1. 서 론

Rapid prototyping(RP) 기술이 실제 생산 공학 분야에서 효율적으로 활용되기 위해서는 제품 형상의 완벽한 3차원 CAD 모델링을 위한 기하학적 표현 수단의 완벽성과 함께 그에 따른 설계 정보를 조형 정보로 온전히 변환(translate/transform) 하여 넘겨줄 수 있는 기술의 구비가 선행 조건으로 요구된다[1]. 현재 대부분의 RP 기술은 'STL' 파일이라는 형상 정보 교환 체계에 기반을 두고 운용이 되고 있으나 3차원 CAD 모델이 STL파일로 바뀌어, 교환되는 과정에서 파일의 정확도(accuracy), 완성도(integrity), 중복성(redundancy)의 관점에서 여러 가지 문제점들이 보고되어 왔고, 이러한 STL체계의 문제점들을 극복하기 위한 현재까지의 연구 결과들은[2,3] CAD시스템들과 RP 조형 장치 사이에서 교환되는 기하학적 형상 정보의 사후 적합성(compatibility)이라는 면에만 치우쳐 해결 방법을 제시한 것으로, RP 조형 기술을 효과적으로 지원하기 위한 장기적이고 근본적인 해결책으로 보기 어렵다.

STEP(Standard for the Exchange of Product Data)은 현재 가장 우수한 차세대 정보 교환 표준 체계로서 국제적인 인정을 받고 있으며, 국내에서도 여러 분야에서 그 활용에 관한 연구가 발표되어 왔다[4-7]. STEP을 RP 조형 기술을 위한 설계 자료 표준 체계로서 이용하자는 제안은 Pratt과 Sriram[8]에 의해 1995년에 미국 NSF(National Science Foundation)가 주관한 학회에서 처음으로 제기되었고, 동년 8월에는 미국 Gilman과 Rock이 매우 초보적이지만 실질적인 정보 전달 체계에 관한 연구 결과를 제시하였다[9]. 이후 1996년에 동 주제에 관한 첫 번째 국제 워크샵이 개최되었고[10], 1998년 6월 ISO TC184/SC4회의에서 열린 'STEP for RP'라는 주제 하의 비공식 모임에서는 STEP을 RP에 적용하기 위한 장·단기적 전략이 제시되었다[11]. 이를 간략히 소개하면 단기적으로는 STL 모델을 triangular faceted B-Rep. model로, 그리고 중기적으로는 triangular faced model을 STEP을 기반으로 정의된 advanced B-Rep. model로 대체하자는 것이며 궁극적으로는 RP조형 공정을 지원하기 위한 STEP AP(Application Protocols)를 공

*학생회원, 홍익대학교 기계공학과
**종신회원, 홍익대학교 기계공학과
- 논문투고일: 2000. 7. 24
- 심사완료일: 2001. 1. 8

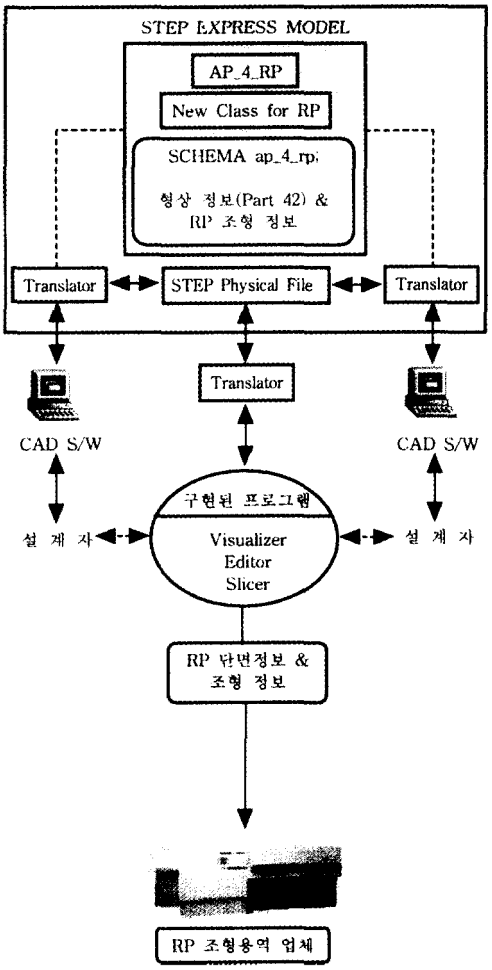


Fig. 1. CAD/RP interface using AP_for_RP.

동 개발하자는 것이 그 주요 골자이다. 동년 9월에는 Dutta, Pratt, 그리고 Sriram[12]이 주축이 되어 보다 구체적인 실현 방안에 대하여 언급한 학술 논문과 Technical Report[13]가 보고되었다. 본 연구에서는 이와 같은 국제 연구 동향에 부응하여 STEP에 근거한 RP조형 정보용 모델을 시험적으로 구축하여 제안하고자 한다.

2. 연구 내용

본 논문에서는 Fig. 1에서 보는 바와 같이 기존의 형상 정보에 조형 정보를 연계하여 RP용 조형 정보 모델을 구성하였다. 그림을 예를 들어 정보 모델의 활용을 설명하면 우선 설계자 측에서는 CAD 시스템에서 생성된 형상 정보와 관련된 조형 정보 추가하여 정의할 수 있고 이렇게 구성된 조형 정보 모델은 다른 설계자나 조형 용역 업체에 전달되어 정의된 조형 공정이 적절하게 정의되었는지, 혹은 원하는 조형 작업이 이루어지기 위해 필요한 조건 등이 어떤 것인지 판단할 수 있도록 하였다. 정의된 조형 정보의 변경이 필요할 경우에는 그 변경된 정보를 다시 조형 정보 모델에 재 입력하여 조형 용역 업체에 전달함으로서 최적의 조형 작업을 수행하는 것이 가능하도록 하였다. 본 연구에 의하여 구현한 프로그램은 STEP 형상 정보의 인식 및 수정, 좌표 변환에 의한 재배치와 같은 기능을 제공하며, 또한 형상 정보에 조형 정보를 입력하는 편집 작업 기능을 통하여 변경된 설계 내용은 새로운 STEP 물리 파일로 재 정의되어 설계자와 설계자간의 혹은 설계자와 생산자간에 재 교환이 가능하도록 하였다. 형상 정보는 part42의 기하학적/위상학적 표현의 내용을 따르는데, 파트 정보를 기준으로 할 때에는 형상 정보 모델 교환을 위해 현재 RP 업계에서 일반적으로 널리 통용되는 STL 표현을 Faceted B-Rep. 형태로 변환하여 표현할 수 있도록 하였다. 또한 파트 형상을 2차원 단면 정보로 전환하여 정의할 때에는 단면 형상 정보에 RP 조형 정보 정의용 엔터티들을 추가하여 조형 정보 모델이 구성되었으며, 정보 모델의 구현을 위한 프로그래밍 작업은 STEP Tools사의 ST-Developer와 C++ class를 사용하여 이루어졌다.

3. STEP 기반 RP 조형 정보 모델 구성

조형 정보 모델의 구성 원리를 설명하기 위하여 다수의 RP 조형 장치의 공통적인 운용 원리를 간단히 살펴보면, 우선 3차원 CAD 형상 모델을 그 형상의 가장 밑 단면에서 시작하여 일정한 두께의 2차원 박 층(thin layer)들의 기하학적인 집합으로 재구성하고, 다음 이렇게 분할된 박 층들의 두께만큼의 기하학적인 부파를 RP 조형 장치들을 이용, 준비되어 있는 조형 재료를 이용하여 층 단면에 수직한 방향으로 밑에서부터 순서대로 적층(accumulation)함으로서 원래의 CAD 모델 형상을 물리적으로 재현해 내는 것으로 설명될 수 있다. RP 조형용 정보 모델 역시 이러한 기계 조형 공정을 반영하여 이루어 져야 한다. 예를 들어 RP 조형을 전제로 한 임의의 기하학적 형상 모델은 기하학적인 정보 이외에도 조형 층의 두께나 그 층을 조형하는 데에 필요한 조형 정보들과 관계된 보다 복잡한 개념의 설계 정보(high level design information)를 포함하는 자료 구조로

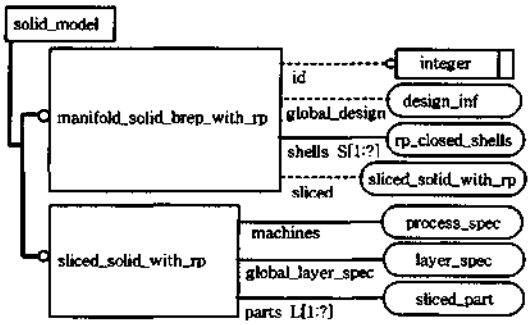


Fig. 2. Configuration of AP_4_RP schema.

이루어 져야 한다. 그러한 개념의 설계 정보의 예로서는 조형 파트를 구성하는 재료(material), 색깔(color), 재료 강도(strength), 표면 거칠기(surface finish) 등을 꼽을 수 있다. 이러한 설계 정보들은 RP 조형을 목표로 하여 제품 형상을 모델링 할 때 이미 설계자에 의하여 결정되고 그 결정 사항이 설계 자료에 반영이 되어 있어야 하는데 STEP은 이러한 용도에 매우 적합하다. 즉 STEP에서 개념적 정보 모델(Conceptual Data Model)을 정의 혹은 생성하는 정보 모델링 언어(Information Modeling Language)인 EXPRESS는 하나의 객체인 entity, 속성(attributes), 데이터 모델간의 관계, 데이터에 대한 제약 조건들을 문서화하는데 사용되며, EXPRESS의 선언 구조체 중 schema는 대상 분야를, type과 entity는 데이터를 표현하고 그들 간의 관계를 정의하므로[14] 전술한 RP 조형 공정 관련 설계 정보를 기하학적 정보와 함께 표현하는 것이 가능하다. 이에 따라 본 논문에서 STEP 에 근거하여 구현한 RP 조형 정보 모델용 스키마(schema)인 AP_4_RP는 Fig. 2에서 보듯이 솔리드 형상에 설계 정보를 결합하여 정의한 부분(manifold_solid_brep_with_rp)과, 단면 형상과 조형 정보를 연계하여 정의한 부분(sliced_solid_with_rp), 이렇게 두 부분으로 나누어 구성되었다.

이들 두 부분의 각각에 대하여 설명하면 다음과 같다. 일반적으로 RP 조형용 CAD 모델의 교환 체계인 STL은 그 자료 성격상 STEP part42 의 솔리드 모델인 Faceted B-Rep.으로 다시 정의할 수 있다. 그러므로 솔리드 형상에 설계 정보를 결합한 manifold_solid_brep_with_rp entity를 정의하여 기하학적 형상 정보에 거칠기, 색상, 재질, 강도 등과 같은 설계 정보를 추가하였다. 예를 들면 STL을 입력받아 처리하는 경우에 STL의 형상을 구성하는 각 삼각형 면 정보들을 part42에 정의된 plane entity로 변환한 후 closed shell을 구성하였다. 한편, RP 조형 작업을 지원하기 위하여 2차원 단면 정보와 조형 정보 연계한 부분에서도 기하학적 정보에 미리 정의된 설계 정보에 근거한 조형 정보를 추가하여 sliced_ solid_with_rp entity를 정의하였다. 이러한 조형 정보 모델을 구성하는 여러 가지 entity들에 대한 보다 상세한 설명은 다음과 같다.

3.1 Manifold_solid_brep_with_rp entity

이 절에서는 주어진 솔리드 형상에 재료, 색깔, 재료 강도, 표면 거칠기와 같은 설계 정보를 추가하여 저장하기 위하여 필요한 여러 가지 entity들을 다룬다. 이 entity를 구성하는 부분 중에서 shells는 형상 정보를, global_design은 조형 정보 정의한다. 여기서 global_design은 형상이 특징에 따라 재구성되지 않고 하나의 closed_shell로 표현되는 경우와 여러 개의 closed_shell들로 재 정의된 형상의 경우이라도 다수의 shell들이 같은 설계 정보를 공유할 경우에 한하여 정의된다. 이때 새로이 정의 된 shell들 중 다수의 shell들이 같은 조형 정보 공유할 경우에는 모든 shell에 동일한 조형 정보 일일이 정의하지 않고 global_design의 정보를 상속받아 사용하도록 fill_design_inf() 함수를 정의하였다(하단 Function참조). 따라서 shells에 있는 local_design이 정의되어 있지 않으면 global_design의 내용을 상속받아 모든 local_design에 조형 정보가 정의되도록 하였다. 예를 들어 LOM장비에서는 조형시 색상, 층 두께 등의 정보가 변경되는 경우가 없으므로 global_design에서 한번만 정의되면 하위의 형상에서도 이를 그대로 상속받아 정의된다.

한편, 일반적으로 shells는 단일한 솔리드 형상인 part를 구성하는 변수로서 서로 다른 설계 정보를 가지는 여러 개의 shell들로 구성될 수 있다. 현재 shell 의 특성은 closed shell로만 한정하여 정의하도록 하고 있는데 이는 open shell인 경우 RP 조형 작업의 특성상 조형 정보 영역이 명확하지 않게 되어 조형이 불가능하게 되고 이에 따라 조형 정보 영역을 제한할 필요에서이다. 마지막으로 sliced 는 파트 형상 정보와 단면 정보가 동시에 조형 정보 모델로 정의되어 있는지 의 여부를 인식하기 위한 목적으로 정의되었으며 3.2절에서 단면 정보 모델에 대하여 보다 상세히 설명하였다.

3.1.1 design_Inf.

design_inf entity는 솔리드 형상 정보 추가될 설계 정보를 정의한다. 이 때 모든 설계 정보 변수는 OPTIONAL로 정의하여 설계자가 필요로 하는 경우

```
ENTITY manifold_solid_brep_with_rp
SUBTYPE OF (solid_model);
  id : OPTIONAL INTEGER;
  global_design : OPTIONAL design_inf;
  shells : SET [1:?] OF rp_closed_shells;
  sliced : OPTIONAL sliced_solid_with_rp;
WHERE
 WH1: id >= 0;
 WH2: fill_design_inf(global_design, shells);
END_ENTITY;
```

```
FUNCTION fill_design_inf
( global_design_inf : design_inf;
  shells : SET OF rp_closed_shells) : BOOLEAN;
IF EXISTS(global_design_inf) THEN
  IF SIZEOF(shells) > 0 THEN  REPEAT i := 1 TO
HIINDEX(shells) BY 1;
    IF(NOT(EXISTS(shells.local_design_inf))) THEN
  shells.local_design_inf := global_design_inf;
    END_IF;
  END_REPEAT;
  END_IF;
END_IF;
   RETURN(TRUE);
  END_FUNCTION;
```

에만 정의되도록 하였다. 또한 하위 entity인 color_table entity는 색상을 나타내는 RGB형식으로 정의되어 List에 순차적으로 대입하게 하였다.

```
ENTITY design_inf;
  surface_finish : OPTIONAL REAL;
  material : OPTIONAL STRING;
  color : OPTIONAL color_table;
END_ENTITY;
```

```
ENTITY color_table;
  ratio : List[3:3] OF INTEGER;
  WHERE
WR1:(0 <= ratio[1]) AND (ratio[1] <=255);
WR2:(0 <= ratio[2]) AND (ratio[2] <=255);
WR3:(0 <= ratio[3]) AND (ratio[3] <=255);
END_ENTITY;
```

3.1.2 rp_closed shells

rp_closed_shells entity에서는 shell을 구성하는 면 정보인 face들과 그 이외의 국부적인 설계 정보를 정의한다. faces는 SET [1:?] OF closed_shell로 표현하여 동일한 조형 정보에 해당할 경우 closed_shell에 단일한 설계 정보가 적용되도록 하였다. local_design은 앞의 3.1절에서 전술하였듯이 공란일 경우에는 manifold_solid_brep _with_rp entity의 global_design정보를 상속받아 정의되며, support는 임의의 shell이 support structure인지 아닌지 그 여부를 정의한다. 예를 들어 RP 조형 장비의 일종인 SLA의 경우 liquid resin을 사용하므로 조형 시 support structure가 필요하며 이를 위한 support 항목은 boolean 변수로 정의한다.

```
ENTITY rp_closed_shells
SUBTYPE OF (connected_face_set);
        faces : SET [1:?] OF closed_shell;
        local_design : OPTIONAL design_inf;
        support : BOOLEAN;
END_ENTITY;
```

3.2 sliced_solid_with_rp entity

이 절에서는 주어진 솔리드 형상의 2차원 단면화 작업 후 생성된 조형 층 단면 정보와 조형 정보를 함께 저장하기 위한 entity들을 다룬다. 이 entity를 구성하는 부분 중에서 global_layer_spec은 조형될

```
ENTITY sliced_solid_with_rp
SUBTYPE OF (solid_model);
  process : process_spec;
  global_layer_spec : layer_spec;
  parts: List[1:?] OF sliced_part;
WHERE
 WH1: fill_rpdata(global_layer_spec, parts);
END_ENTITY;
```

```
FUNCTION fill_rpdata
( global_layer_spec : layer_spec;
  parts : SET OF sliced_part) : BOOLEAN;
IF SIZEOF(parts) > 0 THEN  REPEAT i := 1 TO
HIINDEX(parts) BY 1;
    IF(NOT(EXISTS(parts.local_layer_spec))) THEN
  parts.local_layer_spec := global_layer_spec;
    END_IF;
IF SIZEOF(parts.layers) > 0 THEN  REPEAT i := 1
TO  HIINDEX(parts.layers) BY 1;
    IF(NOT(EXISTS(parts.layers.local_layer_spec)))
THEN  parts.layers.local_layer_spec :=  parts.local_layer_spec;
    END_IF;
  END_REPEAT;
  END_IF;
  END_REPEAT;
  END_IF;
   RETURN(TRUE);
  END_FUNCTION;
```

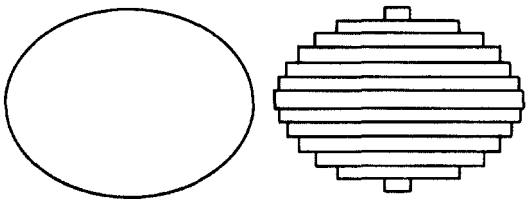


Fig. 3. Expression of 2-D contour using loop(left) and bid scan(right).

part에 전반적으로 적용되는 변수로서 layers에서는 2차원 단면의 형상과 각 단면 층의 국부적인 조형 정보들의 집합이 정의되었다. fill_rpdata() 함수에서는 (하단 Function참조) 3.1 절의 fill_design_inf()함수와 같이 sliced_part의 local_layer_spec항목이 공란일 경우 global_layer_spec의 정보를 상속받아 정의된다.

3.2.1 Process spec.

process_spec entity에서는 사용할 RP 장비의 공정 특성을 정의한다. 이들을 상세하게 서술하면, 먼저 다음에 예시된 entity 구성에서 보듯이 max_layer_thickness와 min_layer_thickness 에서는 조형 장비가 조형할 수 있는 층의 조형 한계 즉 층 두께의 범위를 정의한다. 한편 scan_width_change 에서는 조형 장비가 임의의 조형 층 내에서의 조형 단면을 채우는 방식을 나타낸다. 이는 Fig. 3에서 보듯이 'false'값을 가질 경우 loop 밖의 영역을 제거한(decubing) 후 단면 층을 조형하며, 'true'값을 가지는 경우 선택된 영역을 scan방식으로 붙여 가면서 단면 층을 조형한다.

```
ENTITY process_spec;
  max_layer_thickness : REAL;
  min_layer_thickness : REAL;
  scan_width_change : width_ability;
  max_scan_width : OPTIONAL width_max;
  min_scan_width : OPTIONAL width_min;
  material_change : BOOLEAN;
  color_change : BOOLEAN;
  max_volume : OPTIONAL REAL;
  part_number : OPTIONAL INTEGER;
  estimate_time : OPTIONAL local_time;
  production_time : OPTIONAL local_time;
  origin : OPTIONAL origins;

DERIVE
  ok : BOOLEAN :=
 check_scan_width_change(scan_width_change,
max_scan_width, min_scan_width);
WHERE
 WH1: ok;
END_ENTITY;
```

그리고, scan_width_change가 true의 값을 가지게 되면 그 scan 폭의 한계를 명시해 주어야 하는데, scan 폭의 최대 값과 최소 값을 정의하기 위해 사용되는 변수가 max_scan_width와 min_scan_width이다. 이때 scan_width_change, max_scan_width, min_scan_width의 관계는 유도 속성인 DERIVE와 local rule인 WHERE를 이용하여 정의하였다. 유도 속성인 ok는 check_scan_width_change( ) 함수에서(하단 Function참조) 결정된 값이며, scan_width_change가 true의 값을 가지게 되면 max_scan_width와 min_scan_width가 반드시 존재해야 함을 한다.

이 밖에도 material_change에서는 조형 작업 중에 재료의 변경이 가능 한지의 여부를, 그리고 color_change는 조형 작업 중에 색상의 변화가 가능 한지의 여부를 정의한다. 고가인 RP장비의 운영 효율성을 높이기 위한 변수들도 정의되었는데 먼저 max_volume에서는 제작 가능한 최대 부피를, part_number에서는 정의된 파트(하나의 완성된 솔리드)의 수를, estimate_time에서는 예상 조형 시간을, 그리고 production_time에서는 실제 조형 시간을 정의하였다. 이를 통해 조형 시간의 데이터 베이스가 구축되면 이를 비교 예측하는데 이용될 수 있다. 시간을 나타내는 항목인 local_time entity는 AP203에서 정의된 내용을 사용하였다.

또한, origin에서는 현재 통상적인 RP용 솔리드 형상 정보가 STL로 교환되고 있는 현실을 고려하여 형상 정보의 원본이 처음부터 STEP AP203으로 정의된 것인지, 혹은 STL로 정의된 형상을 변환한 것인지를 선택하도록 하였다. 한편 process_spec과 sliced_solid_with_rp의 관계를 정의한 control_rpdata_ material에서는 rule에서 정의된 조형 정보 오류를 검사한다. 가령, process_spec entity의 material_change의 값이 'false'일 경우에 한하여 global_layer_spec의 material 값과 sliced_part entity의 local_layer_spec의 material 값이 동일함을 검사한다.

3.2.2 layer_spec.

layer_spec entity는 주어진 3차원 솔리드 형상의 2차원 단면 정보에 연계될 조형 정보를 정의한다. 다음에 예시된 entity 구성 요인의 각각을 설명하면, thickness는 조형 층의 두께, width는 vector scan 방식으로 조형할 경우 scan 폭, material은 조형 재료나 바인더, 그리고 color에서는 색상 정보를 정의한다. 본 조형 정보 정의에서는 thickness 항목은 반듯이 지정하도록 하되 width와 material, color 항목은 선택적으로 지정하도록 하였는데 그 이유는 조형

```
HIINDEX(sliced_solid_with_rp.parts) BY 1;
        IF((NOT ability) AND
 (sliced_solid_with_rp.parts.local_layer_spec.
material <>sliced_solid_with_rp.
global_layer_spec.material)) THEN ok := FALSE;
          END_IF;
IF   SIZEOF(sliced_solid_with_rp.parts.layers)   >   0
THEN REPEAT i := 1 TO
HIINDEX(sliced_solid_with_rp.parts.layres) BY 1;
        IF((NOT ability) AND
 (sliced_solid_with_rp.parts.layres.local_layer_spec.
material <>sliced_solid_with_rp.parts.
local_layer_spec.material)) THEN ok := FALSE;
 END_IF;
   END_REPEAT;
 END_IF;
  END_REPEAT;
END_IF;
 WHERE
   WH1: ok;
END_RULE;
```

```
TYPE origins = ENUMERATION OF (STL, AP203);
END_TYPE;
```

```
FUNCTION check_scan_width_change
( check1 : width_ability;
  check2 : width_max;
  check3 : width_min ) : BOOLEAN;
IF((check1)     AND     EXISTS(check2)     AND
EXISTS(check3)) THEN RETURN (TRUE);
 END_IF;
 IF NOT (check1) THEN RETURN (TRUE);
 END_IF;
RETURN (FALSE);
END_FUNCTION;
```

```
TYPE width_ability = BOOLEAN;
END_TYPE;
TYPE width_max = REAL;
END_TYPE;
TYPE width_min = REAL;
END_TYPE;
```

시, 조형 층의 두께는 반드시 필요한 정보이지만 나머지 정보는 조형 장비의 특성에 따라 선택적으로 반영될 수 있는 정보이기 때문이다.

3.2.3 sliced_part

sliced_part entity는 파트 형상을 구성하는 2차원 단면들의 집합체로써, 2차원 단면의 기하학적 형상 정보와 이 단면 층에 관계된 국부적인 조형 정보 정

```
ENTITY layer_spec;
  thickness : REAL;
  width : OPTIONAL REAL;
  material : OPTIONAL STRING;
  color : OPTIONAL color_table;
END_ENTITY;
```

의한다. 여기서 조형 정보인 local_layer_spec이 비워져 있는 채로 정의되는 경우에는 3.2.절의 global_layer_spec의 정보를 상속받아 정의된다. 2차원 단면 형상 정보는 layers에서 정의되는데 layer_type type에서 layer_loop와 layer_scan중 한가지를 선택하여 형상 정보와 조형 정보를 지정하도록 하였다. 이는 3.2.2절의 Fig. 3에서 보여지듯이 scan_width_change 값에 따라 단면의 형상 정보를 loop나 비드 형상으로 표현하는 까닭이다. 한편, 유도 속성의 변수인 total_layer에서는 layers의 List의 요소 수로써 조형될 part의 전체 층수를 나타내어 조형 시 참고하도록 하였다.

한편 TYPE에서 정의된 하위 클래스인 layer_loop entity는 LOM방식에서 보듯 단면 형상을 loop에 따라 laser로 재단하여 조형하는 경우 2차원 단면의 contour 형상 정보를 하나의 단일한 면 정보인 face로 정의하는 경우에 사용된다. 이 경우 local_layer_spec에서는 해당 loop에 관련된 조형 정보가 정의되는데 local_layer_spec이 공란일 경우에는 상위 entity인 sliced_part의 local_layer_spec정보를 상속받아 정의된다. 이 경우 3.2.1절의 Fig. 3에 나타낸 것과 같이 scan_width_change 값이 'false'이므로 형상 정보인 layer_bound에서도 이 loop를 하나의 face로 정의하게 된다. support는 전술 한대로 해당 loop가 support structure인지의 여부를 정의한다.

한편 layer_scan entity는 3D Printing이나 SLS 방식에서 보듯 2차원 단면의 contour 형상 정보를 단일 스트로크의 scan폭을 가진 여러 개의 비드 막대 형상으로 나누어 정의한 경우에 사용된다 이 경우 3.2.1절의 Fig. 3에서 보듯 scan_width_change 값은 'true'가 되며 이때 형상 정보인 segments에서는 이 비드 형상을 4각형으로 표현한다. support에서 이 비드 형상들이 support structure인지의 여부를 정의하며, 조형 정보 local_layer_spec은 layer_loop의 경우 같이 정의된다. 여기서 세 번째 entity 예시에서 서술한 segment_spec은 Fig. 4에서 보여진 바와 같이 단일 스트로크의 scan 폭을 가진 한 개의 비드에 대한 조형 정보를 정의한다. 즉 position: Cartesian_Point에서는 비드 형상의 시작점을, 그리고 dimension에서는 길이와 폭을 지정한다.

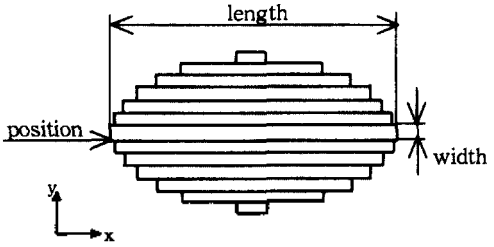


Fig. 4. Illustration of segment_spec.

dimension은 실수형 List로 표현되며 첫 번째 값은 길이, 두 번째 값은 폭을 나타내어 segment bid형상을 구현한다. 이때 position은 원점에서 가장 가까운 점으로 정의한다.

```
ENTITY sliced_part;
  name: STRING;
  local_layer_spec : OPTIONAL layer_spec;
  layers: List[1:?] OF layer_type;
DERIVE
  total_layer: INTEGER := SIZEOF(layers);
END_ENTITY;
```

4. 전산 프로그램의 구축

본 논문에서는 앞장에서 서술한 조형 정보 모델을 운용, 처리하기 위하여 Fig. 5와 같은 시스템 운용 구조를 지닌 전산 프로그램을 개발하였다. 프로그램은 크게 STL 파일 및 물리적 STEP 파일을 인식/변환하여 가시화 하는 부분, 설계자나 생산자가 파일 내용을 수정하는 부분, 그리고 이 파일을 조형 정보로 변환하기 위한 2차원 단면화 작업을 수행하는 부분 이렇게 3가지 모듈을 기본 틀로 하여 구축되었다. 우선, 가시화 모듈에서는 STEP파일로부터 형상 정보를 인식하여 list를 구성한 후 OpenGL 루틴으로 넘겨서 형상의 시각적 디스플레이가 이루어진다. 다음, 수정 작업 모듈에서는 인식된 형상 개체의 방향, 위치, 크기, 개수 등의 정보나 2차원 단면화를 위한 조형 정보 수정, 확인하는 작업이 가능하도록 하였다. 마지막으로 인식된 형상 자료를 기반으로 2차원 단면화 작업을 수행하는 모듈에서는 조형 작업에 필요한 단면 정보가 생성되어 RP 조형 장비로 전달되도록 하였다. 전산 프로그램의 운영 체계의 구축은 Fig. 6에서 보는 바와 같이 Pentium II processor와 MS-Windows 98 환경 하에 Visual C++6.0을 이용하여 수행하였고, 전산 그래픽스 tool로는 OpenGL을 이용하였다. STEP 파일의 자동 변환을 위한 작업 tool로서는 STEP Tools 사의 ST-Developer 7.0을 사용하였다. 각 모듈에 대한 상세한 설명은 다음과 같다.

```
TYPE layer_type = SELECT
  (layer_loop,
  layer_scan);
END_TYPE;
```

```
ENTITY layer_loop
SUBTYPE OF (edge_loop);
  layer_num: INTEGER;
  local_layer_spec : OPTIONAL layer_spec;
  layer_bound : face_bound;
  support : BOOLEAN;
WHERE
 WH1: layer_num >= 0;
END_ENTITY;
```

```
ENTITY layer_scan;
  layer_num: INTEGER;
  local_layer_spec : OPTIONAL layer_spec;
  segments: List[1:?] OF segment_spec;
  support: BOOLEAN;
DERIVE
  total_segment: INTEGER := SIZEOF(segments);
WHERE
 WH1: layer_num >= 0;
END_ENTITY;
```

```
ENTITY segment_spec;
  segment_num: INTEGER;
  local_layer_spec : OPTIONAL layer_spec;
  position: Cartesian_Point;
  dimension: List[2:2] OF REAL;
WHERE
 WH1: segment_num >= 0;
END_ENTITY;
```

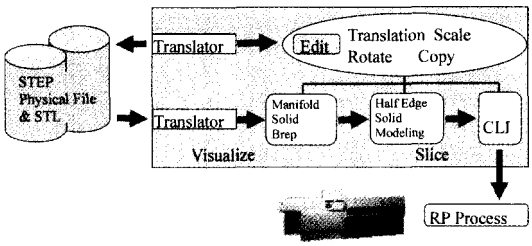


Fig. 5. System configuration of developed program.

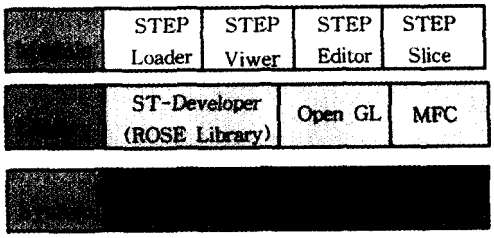


Fig. 6. System environment of developed program.

4.1 STEP 파일 인식 및 Visualization

이 모듈은 입력받아 인식된 솔리드 형상을 그래픽 모듈을 이용하여 시각적 디스플레이 화하는 기능을 담당한다. 전술한 바와 같이 이 모듈은 ST-Developer의 ROSE compiler를 이용하여 part42의 entity에 근거한 C++ class로 구현하였고, 생성된 class들 및 Rose library를 이용하여 STEP 파일에서 서술하는 entity들의 data를 각 entity들 사이의 포함 관계에 따라 list로 구축하였다. STL을 입력받아 이를 인식, AP203의 솔리드 모델의 B-Rep. 형상으로 변환하는 경우에는 형상을 구성하는 각각의 면(face) 정보가 OpenGL의 gluNurbsSurface() 함수를 이용하여 가시화 하게 된다. 한편 AP203의 솔리드 모델의 B-Rep. 형상으로 직접 입력받은 경우에는 Plane, Cylindrical surface, Conical surface등의 기본 형태를 각각 NURBS 형태로 통일하여 구현한 후 외부 loop와 내부 loop의 정보를 변경하여 형상이 가시화된다.

4.2 STEP FILE Editing

설계자에 의하여 생성된 CAD 파일 정보가 STEP 파일로 전환되어 RP 조형 정보가 추가된 채로 다른 설계자나 생산자에게 일방적으로 전달되기보다는, 이 파일을 전달받는 객체인 다른 설계자나 생산자에 의하여 파일 내용이 편집(editing)될 수 있다면 이들 끼리의 쌍방향 설계 및 조형 정보 교환이 가능하다. 예를 들면 최적의 조형 작업 조건을 설정하기 위하여 형상 파트의 재배치 작업과 같은 편집 작업 기능이 이에 해당한다. 이에 따라 본 논문의 STEP file editing 기능은 대개 형상 정보에서 인식된 형상 개체의 방향, 위치, 크기, 개수 등의 정보를 작업자의 의도대로 다시 재정의 할 수 있도록 하였다. 이외에도 2차원 단면화를 위한 조형 층의 두께, 조형 재료, x-y 단면에서의 scan 정밀도와 같은 조형 정보들도 생산자의 입장에서 편집할 수 있는 기능을 지원하므로 새로이 정의된 조형 정보는 설계자끼리 혹은 설계자와 생산자끼리 상호 정보 교환을 가능하게 한다.

4.3 STL to STEP conversion

현재 RP 업계에서 일반적으로 통용되는 STL을 입력받는 경우에는 인식된 STL의 면(face)정보로부터 point list와 edge list, face list를 새롭게 구성하고 이를 STEP 형상 정보로 변환한다. 변환 시에는 part42에서 정의한 plane entity를 참조하여 세 개의 꼭지점을 한 loop로 연결하여 면 정보를 정의하였다. 형상 정보의 변환 결과로 생성된 파일 의 크기는 원래 STL ASCII 포맷의 그것보다 증가하게 되는데 이는 하나의 삼각형 면 정보를 표현하기 위해서 세 개의 점 data외에 세 개의 edge data와 하나의 plane data가 각각 정의되기 때문이다.

4.4 STEP Slicing

형상 정보의 가시화 및 편집 작업이 완료되어 RP 조형 작업 필요한 층 단면 정보를 추출하기 위해서는 조형 파트 형상을 기준으로 2차원 단면화 작업을 행하게 하게 된다. 이를 위해서는 형상 정보의 geometry와 topology를 단면화 작업에 용이하도록 별도의 자료 구조로 전환하게 되는데 이때 STEP 파일 내에 포함되어 함께 전달된 조형 층 두께 정보가 직접 단면화에 필요한 작업 변수로 이용될 수 있다. 이 결과로 생성된 단면 정보는 2차원 단면 정보 교환 체계의 일종인 CLI(Common Layer Interface) 파일로 저장되어 이종의 RP장비를 이용한 조형 작업에서도 이용될 수 있다.

4.5 Virtual prototyping

조형 정보와 2차원 층 단면 정보가 생성되면 이를 기반으로 전산 그래픽스 기능을 적용, 조형 장비의 특성에 맞는 가상의 조형 파트의 형상을 가시화 할 수 있다. 이 기능을 이용하면 형상을 직접 조형하지 않고도 예상될 수 있는 문제점을 시각적으로 파악, 설계 및 조형 정보 수정하는 것이 가능하므로 경제적인 조형이 가능하다.

5. 전산 프로그램의 실행 예

본 절에서는 구축된 전산 프로그램을 이용하여 CAD 파일을 처리하는 과정을 설명하고자 한다. 먼저 Fig. 7은 fan.stl이라는 시험 형상 파일을 part42으로 변환하고 있는 과정을 보여준다. Fig. 8은 STEP 물리 파일 중 anc101.step 이라는 시험 형상 파일을 기준으로 조형 정보를 입력하는 과정을 보여주는데, 이때 그림의 좌측 하단에 있는 slice dialog box에서

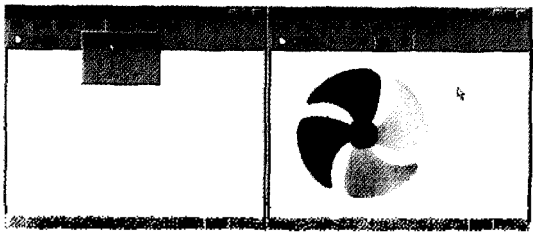


Fig. 7. Process of STL conversion to STEP file.

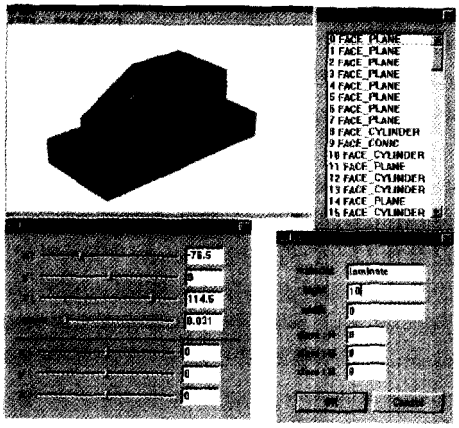


Fig. 8. Process of importing RP information.

조형 정보를 입력받아 STEP 물리 파일로 저장된다. Fig. 9는 이처럼 조형 정보와 같이 정의된 STEP 물

```
ISO-10303-21;
HEADER;
............
FILE_SCHEMA (('AP_4_RP'));
ENDSEC;
DATA;
#10=MANIFOLD_SOLID_BREP_WITH_RP('',0,#12,#11,#1921);
#11=RP_CLOSED_SHELLS($,$,(#701),$,'FALSE');
#12=DESIGN_INF(0.01,#13);
#13=COLOR_TABLE((255.,0.,0.));
...
#701=CLOSED_SHELL('',(#679, ...));
...
#1921=SLICED_SOLID_WITH_RP('',#1922,#1923,(#1924));
#1922=PROCESS_SPEC(0.1,0.001,'FALSE',$,$,'FALSE','FAL
SE',$,1,$,$,$);
#1923=LAYER_SPEC(0.01,$,'Laminated',#13);
#1924=SLICED_PART('',$,(#1925, ...));
#1925=LAYER_LOOP(0,$,#2465,'FALSE');
...
#2465=FACE_BOUND('',#2357,.T.);
...
ENDSEC;
END-ISO-10303-21;
```

Fig. 9. A STEP physical file based on AP_4_RP.

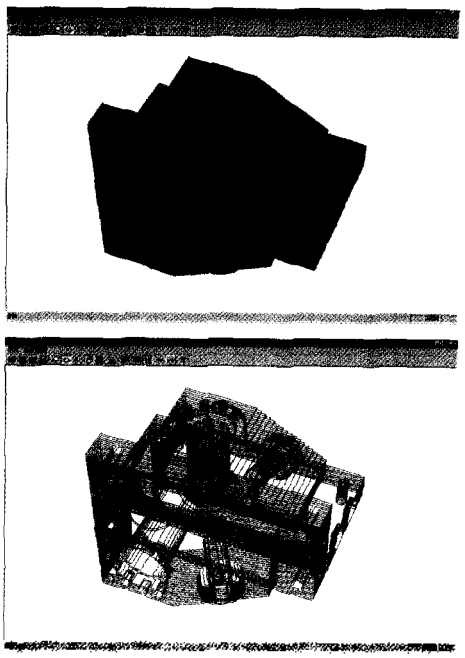


Fig. 10. An example of edited anc101.step parts.

리 파일의 한 부분이다. 그림에서 header부를 살펴보면 FILE_SCHEMA(('AP_4_RP'));에서 'AP_4_RP'스키마를 사용하였음을 보여주며, data부를 살펴보면 AP_4_RP에서 정의한 manifold_solid_brep_with_rp를 사용하여 형상 정보 설계 정보를 추가하였다. 이 예제에서는 형상을 하나의 closed_shell로 정의하였으며, 형상이 support structure가 아니므로 support는 'false' 값으로 정의되었으며, 2차원 단면 정보를 sliced_solid_with_rp에 저장할 경우 process_spec은 LOM 장비로 가정하여 정의하였다. Fig. 10, 11, 12, 13은 편집 작업의 예를 보여주는 그림들로서 먼저 Fig. 10은 두 개의 anc101.step 파일을 서로 rotation, translation 변환을 한 후 결합하여 단일한 파일로 정의하고 이를 2차원 단면화 한 결과이다. 또한 Fig. 11은 anc101.step의 앞면 실린더 부를 삭제하고 평면을 NURBS surface로 수정/편집한 후 이를 단면화한 결과이며 Fig. 12는 서로 다른 크기의 두 개의 part10.step 파일을 결합하여 단일한 파일로 정의하고 이를 2차원 단면화 한 결과이다. 마지막으로 Fig. 13은 서로 다른 파일인 lower_carriage.step파일과 anc101.step파일을 여러 개 서로 혼합 배치하여 단일한 파일로 정의하고 이를 2차원 단면화 한 결과이다. virtual prototyping의 수행 예를 보여주는 Fig. 14는 anc101.step 파일에 대하여 LOM장비를 기준으로 한 가장

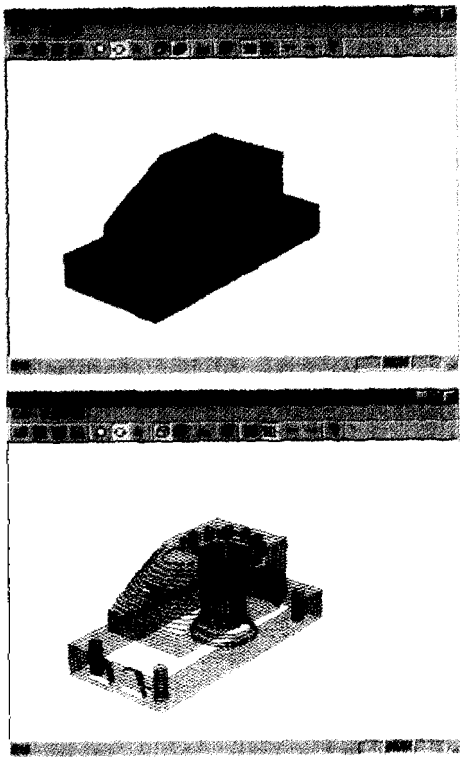


Fig. 11. An example of edited anc101.step parts.

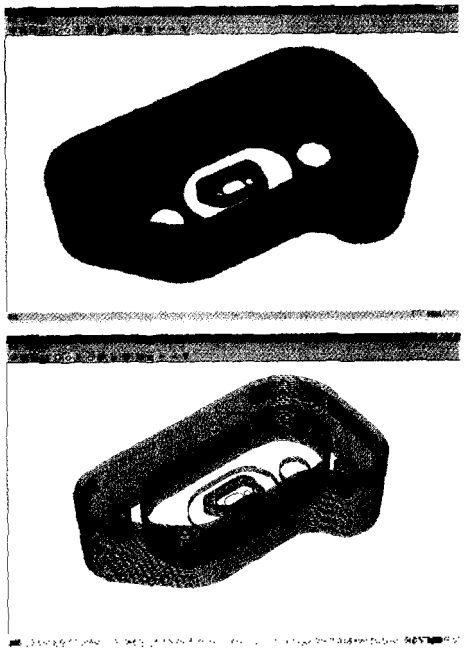


Fig. 12. An example of edited part10.step parts.

단순한 형태의 가상 조형 파트의 형상을 가시화한 결과이다.

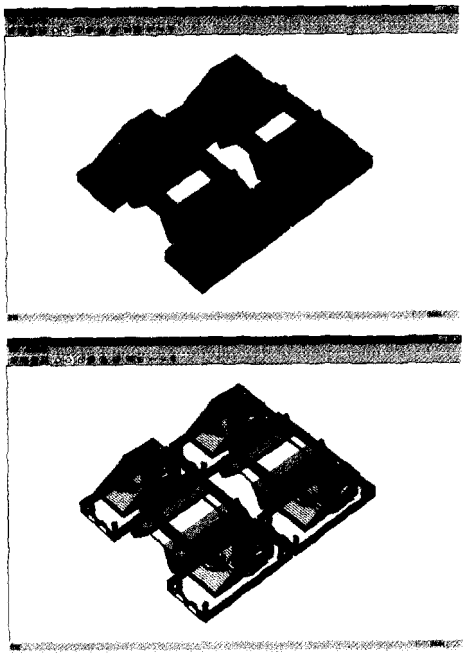


Fig. 13. An example of edited lower_carriage.step parts & anc101.step parts.

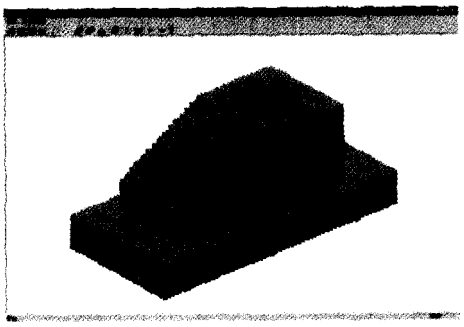


Fig. 14. A virtual prototyping image of anc101.step part.

6. 결 론

본 논문에서는 최근 차세대 정보 교환 표준 체계로서 국제적인 인정을 받고 있는 STEP에 의거하여 RP조형용 조형 정보 모델을 시험적으로 정의하고 이를 제시하였다. 조형 정보 AP_4_RP 스키마는 EXPRESS 언어로 정의하며, AP203 및 Part42의 형상 정보를 상속받아 사용하고 여기에 조형 정보 추가로 정의하는 방식으로 구축되었다. 제시한 조형 정보 모델을 검증하기 위하여 STEP 관련 전산 프로그램을 구축하였으며 이를 현재 통용되고 있는 몇 개의 STEP 불러파일을 대상으로 실행하였다.

본 논문은 STEP을 RP분야에서 적용하기 위한 정보 모델을 처음으로 제시하였다는 데에 그 의의가 있다. 조형 정보가 포함된 형상 정보를 이용하면 설계자와 생산자간의 의사 전달이 가능할 뿐만 아니라 2차원 단면화 작업을 수행 후 조형 정보를 가진 단면 정보를 얻게 되어 보다 정확하고 경제적인 조형 작업이 가능하다. 향후 연구 작업으로서는 functionally graded material과 같이 기하학적 형상 정보와 재질 정보 및 조형 정보가 복잡하게 결합된 형상 파트의 정의의 관한 연구가 요구되며 이와는 별도로 새로운 조형 장비의 특성을 추가하여 정의하는 기준이 마련되어야 한다. 전산 모듈에서는 자유곡면을 포함한 Part 42 Integrated generic resource: geometric and topological representation에서 정의한 모든 형상을 완벽하게 가시화 할 수 있는 기능을 보완하기 위한 연구가 추가로 수행될 것이다.

감사의 글

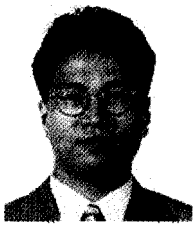
이 논문에서 발표된 연구는 1999년도 한국과학재단의 핵심연구지원사업(981-1007-048-2)에 의하여 지원되었음.

참고문헌

1. Kochan, D., *Solid Freeform Manufacturing*, Manufacturing Res. and Tech. 19, Elsevier Science Publishers B.V. 1993.
2. Bøhn, J. H. and Wozny, M. J., "Automatic CAD-Model Repair: Shell-Closure", in: H. L. Marcus *et al.* (eds.), Proc., *Proceedings of Solid Freeform Fabrication Symposium*, The University of Texas at Austin, August 3-5, pp. 86-94, 1992.
3. Barequet, G. and Kaplan, Y., "A data front-end for layered manufacturing", *Computer-Aided Design*, Vol. 30, No. 4, pp. 231-43, 1998.
4. 이영준, 고광욱, 유상봉, "STEP을 이용한 CAD 데이터 변환 시스템의 구현", *한국 CAD/CAM 학회 논문집*, 제1권 제2호 pp. 87-96, 1996.
5. 최 영, 조성욱, 권기억, "STEP을 이용한 유한요소해석 정보모델 구축", *한국 CAD/CAM 학회 논문집*, 제3권 제1호, pp. 48-56, 1998.
6. 함진호, 김준환, 한순흥, "인터넷에서 STEP 모델의 스테레오 가시화", *한국 CAD/CAM 학회 학술발표회 논문집*, pp. 123-127, 2000.
7. 강석호, 김민수, 김상호, "STEP표준과 Web을 이용한 RPD환경 구축", *한국 CAD/CAM 학회 논문집*, 제5권 제1호, pp. 23-32, 2000.
8. Pratt, Mike and Sriram, R. D., "The STEP Standard as a Geometry Transfer Medium for SFF", *NSF Workshop on Design Method. for Solid Freeform Fabrication*, CH. 4.6.11., June 5-6, 1995.
9. Gilman, C.R. and Stephen J. Rock, "The Use of STEP to integrate Design and Solid Freeform Fabrication", *Proceedings of SFF Symposium*, U. of Texas, August, pp. 213-20, 1995.
10. Jee, H, "STEP-based SIF", *Solid Interchange Format Workshop*, National Institute of Standards and Technology, Gaithersburg, MD, USA, December 1996.
11. Mak, Helium, "Minutes for the STEP for RP meeting", *ISO/STEP Meeting*, Germany, June 7. 1998.
12. Dutta, D., Kumar, V., Pratt, M. J. and Sriram, R. D., "Towards STEP-based Data Transfer in Layered Manufacturing", in CD-ROM *Proceedingss. of IFIP WG5.2/5.3 Conf.* PROLAMAT '98, Kluwer Academic Publishers Trento, Italy, September, 1998.
13. Marsan, A. L., Kumar, V., Dutta, D. and Pratt, M. J. *An Assessment of Data Requirements and Data Transfer Formats for Layered Manufacturing*, NISTIR 6216, September, 1998.
14. STEP 연구회, *제품 모델 정보 교환을 위한 국제 표준 (ISO 10303) STEP*, 성안당, 1996.

이 병 열

1999년 홍익대학교 기계공학과 학사
2001년 홍익대학교 기계공학과 석사
2001년~현재 서울대학교 제어계측신기술연구센터 연구원
관심분야: Geometric modeling, STEP



지 해 성

1986년 서울대학교 공과대학 기계설계학과 학사
1988년 서울대학교 공과대학원 기계설계학과 석사
1996년 미국 MIT대학원 기계공학과 박사
1988년~1992년 한국과학기술연구원(KIST) CAD/CAM 연구실 연구원
1996년~1997년 미국 상무성산하 국제표준연구소(NIST) 연구원
1997년~현재 홍익대학교 기계공학과 조교수
관심분야: Rapid Protuping, 최적설계, Geometric Modeling