

STEP을 이용한 통합제품정보모델(IPIM)개발

Development of Integrated Product Information Model Using STEP

서효원*, 유상봉**

Hyowon Suh*, Sangbong Yoo**

Abstract

This research proposes an Integrated Product Information Model (IPIM) using STEP (Standard for the Exchange of Product model data) for Computer Integrated Manufacturing (CIM) or Concurrent Engineering (CE). IPIM is based on Geometry and Topology (STEP Part 42), Form Feature (STEP Part 48), and Tolerance (STEP Part 48) for representing the integrated information of mechanical parts. For the IPIM, 1) new entities are developed for integration of existing entities, and 2) the existing entities are restructured and modified for a special application protocol. In CIM or CE, the advantages of using IPIM having integrated form of geometry, feature and tolerance are 1) integration of product design, process design and manufacturing sequentially or concurrently. 2) keep the product data consistency, modified by different domain, and 3) automatic data exchange between different application software and different hardware. The prototype system is composed of CAD, Data Probe, DBMS and SDAI (Standard Data Access Interface), and the generated STEP data is stored in a STEP file or DBMS for other applications.

1. 서론

CIM (Computer Integrated Manufacturing) 또는 CE (Concurrent Engineering)의 필요성

이 널리 인식되고, 그에 필요한 요소 기술로서 CAD/CAM, 네트워크, 정보 시스템, 소프트웨어 인터페이스, 그래픽스 등에 대한 연구가 활발히 진행되고 있다. 이러한 정보 및 컴

* 생산기술연구원 생산시스템개발센터

** 인하대학교 자동화공학과

퓨터 기술의 도약에도 불구하고 시스템 통합은 아직 실용화 되지 않고 있다. 이것은 생산 시스템에서 사용되고 있는 요소 기술들의 표준이 정립되어 있지 않기 때문이다. 데이터 교환의 측면에서 본다면, 하드웨어나 소프트웨어의 인터페이스를 통하여 서로 다른 시스템이 정보를 교환하여도 교환된 정보가 다른 시스템에서 이용할 수 없는 형태이거나 불완전한 경우가 대부분이다. 이러한 문제점은 관련된 자동화 시스템들이 공통의 통합 정보 모델을 사용하여 다른 시스템으로부터의 데이터를 별도의 변환이나 해석 없이 바로 이용함으로써 가장 효과적으로 해결할 수 있다.

기존의 CAD 시스템간의 데이터 교환은 제품의 형상을 중심으로 정의한 IGES, DXF, PDDI, SET 등의 양식을 이용해 왔으나, 제품의 해석 및 생산에 필요한 형상특징, 공차, 제품관리 등에 관한 정보는 표준이 정립되지 않아 이기종 시스템간의 정보교환이 원활히 이루어지지 않고 있다 [1]. ISO (International Standard Organization)에서는 이러한 문제점을 인식하고 산업 전반에 걸친 다양한 제품의 설계, 해석, 생산, 그리고 관리에 필요한 모든 제품정보를 정의하는 STEP (Standard for the Exchange of Product model data) 표준을 1983년부터 추진하고 있다. STEP은 광범위한 응용 분야에 필요한 정보모델을 체계적으로 개발하기 위하여 이를 공통 자원 (General Resource : Part 41 - 99)과 응용 자원 (Application Resource : Part 101 - 199)으로 분리하여 개발하고 있고 현재 Geometric and topological representation [2], Representation Structure [3] 등 많은 부분의 공통 자원

이 개발되어 있다. 이러한 공통 자원을 이용하여 구성된 특정한 응용 분야에서 사용될 정보모델이 응용 자원이며, Draughting [4], Finite element analysis [5] 등이 개발되어 있다.

공통자원과 응용자원으로 개발된 정보 모델은 범위, 필요조건, 적합성 테스트 조건 등과 함께 응용프로토콜이 되며, 산업의 각 응용 분야에서 사용되는 응용프로그램들은 해당되는 응용프로토콜을 이용하여 데이터 교환을 실현한다. 응용프로토콜의 정보모델은 이미 개발된 공통자원과 응용자원 중 필요한 엔티티 들을 연결하여 구성하며, 새로운 엔티티, 제약조건, 속성 등을 첨가할 수 있다. 그러나 다수의 공통자원과 응용자원을 연결할 때 발생하는 엔티티의 중복, 제약조건 사이의 충돌, 그리고 적합성 테스트 등에 관한 일관된 해결 방안이 마련되지 않아 응용프로토콜의 개발은 아직 초기 단계이다. 본 연구는 STEP 공통자원을 이용하여 CIM 및 CE 시스템에서 필요한 통합제품정보모델(IPIM)을 개발하고 실제 구현을 위한 프로토타입을 개발하였다. 본 연구에서 구성한 IPIM은 기계 공작물의 설계, 프로세스 계획, 그리고 가공을 대상으로 하여 이에 필요한 형상, 형상특징, 그리고 공차를 포함하고 있다. 생성된 STEP 데이터는 데이터베이스 시스템 또는 STEP 화일에 저장되어 다른 응용프로그램이 사용할 수 있다.

2. 관련 연구

서로 다른 응용프로그램이 데이터를 공유하는 방법으로 화일 교환이나 메세지 전송

등의 방법이 주로 이용되어 왔으나, 정보 시스템의 발달과 교환 데이터의 양적 증가에 따라 공통 데이터베이스의 구축이 증가하고 있다. NIAM [6], SQL [7], IDEF1X [8] 등은 이러한 공통 데이터베이스 구축에 많이 쓰이고 있는 정보 모델링 언어이다. 이러한 기존 언어는 E-R (Entity-Relation) 모델 [9]에 기초를 둔 방법으로 공학 분야에 많이 사용되는 복합 객체나 제약 조건의 표현 등에 제한이 있다. EXPRESS [10]는 객체 지향 모델링 언어로서 새로운 타입의 정의와 캡슐화가 가능하고 다양한 제약조건을 부가할 수 있다.

EXPRESS는 STEP에서 개발하였고 STEP의 모든 작업에서 사용되고 있으며, 표준화 작업 외에도 다양한 분야의 연구 또는 산업 분야에서 제품모델과 시스템 통합을 위하여 활용되고 있다. 스위스의 CERN에서는 물리학 시뮬레이션 패키지의 형상 데이터를 STEP 파일로 변환하여 CAD 시스템과의 인터페이스를 구현하였다 [11]. 프랑스의 Nancy 대학에서는 EXPRESS로 정의된 제품 모델을 OSI (Open System Interconnection)에서 정의한 생산 메시지 표현 방식인 MMS (Manufacturing Message Specification)로 변환하는 변환기를 개발하여 STEP과 MAP의 결합을 시도하였다 [12]. 독일의 DEC에서는 미국의 NCBI (National Center for Bio-technology Information)에서 세계의 각 지역 연구소로 제공되는 생체 데이터를 변환하는 프로그램을 개발하였다. 이 프로그램의 역할은 OSI의 ASN.1 (Abstract Syntax Notation)으로 저장된 NCBI 데이터를 STEP 파일로 변환하고 각 연구소에서 사용하는 데이터베이스 시스템에 저장

하는 것이다 [13]. 독일의 CIMTOFI (CIM system with improved capabilities for furniture industry) 프로젝트에서는 가구의 설계와 생산의 통합을 위한 제품 모델을 EXPRESS로 구현 하였다 [14].

STEP에서는 파트번호 40번대의 공통자원들을 결합하여 특정분야의 응용 프로그램들이 공통으로 사용할 수 있는 응용자원 또는 응용프로토콜을 개발하고 있으나 CIM 또는 CE에서 필요한 형상, 형상특징 및 공차 정보를 결합한 통합제품정보모델은 포함되지 않고 있다. 한편 이러한 세가지 정보의 결합은 주로 솔리드모델 데이터표현 [15] 및 형상특징 [16] [17] 연구 분야에서 논의되었으나 구체적인 정보 구조는 발표되지 않았다. 본 연구에서는 STEP을 이용하여 이들 세가지 정보가 통합된 제품정보의 구조 (IPIM)를 개발하고, 이를 바탕으로 CAD 시스템, STEP 파일 및 데이터베이스 시스템을 통합하는 프로토타입을 개발하였다.

3. IPIM 을 위한 STEP의 공통 자원 (General Resources)

본 연구에서 구성한 IPIM은 STEP에서 정의된 표준을 이용하여 EXPRESS로 정의되었다. 이용된 주요한 STEP 파트들은 Geometric and topological representation(Part42) [2], Shape variational tolerances(Part47) [18], 그리고 Form features(Part48) [19]이다. 그 밖에 메타정보들의 표현을 위하여 Fundamentals of product description and support(Part41) [20]과 Representation structures(Part43) [3]을 포함하고 있다.

Part42는 제품의 형상을 명확하게 표현하기 위한 것으로 다음의 3가지 스키마를 포함하고 있다.

1) geometry_schema : point, vector, parametric curve, surface 등을 정의하고 있다.

2) topology_schema : vertex, edge, face, path, loop, shell 등을 정의하고 있다.

3) geometric_model_schema : CSG와 B-rep 등의 solid model 및 surface model, wireframe model 등을 정의하고 있다.

파트 47에 정의된 공차는 크게 geometric_tolerance와 plus-minus-tolerance 두가지이다. Geometric tolerance는 가공된 제품의 기하학적 특성에 대한 허용 범위를 정의하며, 적용 대상을 나타내는 toleranced_shape_aspect와 허용영역의 경계를 나타내는 zone_elements 및 크기를 속성으로 갖는다. 또한 datum을 참조하는 공차, 이미 정의된 단위를 사용하는 공차, 존재하는 공차를 수정하는 공차 등의 서브타입들을 통해서 다양한 공차를 정의할 수 있다. Plus-minus-tolerance는 부여된 치수의 허용범위를 정의하는 공차로서, 대상 치수를 나타내는 toleranced_dimension과 범위를 나타내는 range 등을 속성으로 갖는다.

파트 48은 형상 특징의 특성을 다룬 form_feature_schema, 특징 형상을 표현하기 위한 form_feature_representation_schema로 구성된다. Form_feature_schema는 실제 산업 현장에서 널리 관심을 가지는 형상을 특성화한다. 즉, 형상의 정형들이나 시각을 확립하고, 특징의 형상을 하나 또는 그 이상의 시각과 연관지우는 방법을 제공한다. Form_feature_representation_schema는 형상 모델링에서 사

용할 다양한 표현들을 제공한다. 구체적인 표현방법으로 기하모델 요소들을 사용하는 명시적인 방법과 직경, 중심선 등의 매개변수 데이터를 사용하는 암시적인 방법을 지원한다.

파트 41은 제품의 표현에 사용되는 기본적인 데이터 타입을 포함한다. 이러한 타입에는 제품 구성, 승인, 계약 (contract), 시간, 인적 구성, 치수 등이 포함된다. 파트 43은 표현에 관한 전체적인 구조를 정의한다. 예를 들어, 파트 42의 기하학적인 표현은 여러 방법 중의 특수한 경우이다. 또한, 서로 다른 좌표계 간의 관계 또는 변환식을 표현하는 정보구조를 포함한다.

4. 통합 제품 정보 모델(IPIM)

IPIM에서 한 기계 가공물은 part로 표현되며, part는 id, name, geometric_model, part_features, 그리고 part-features를 속성으로 갖는다. 이 정보모델 중 part와 관련된 상위 레벨 부분을 EXPRESS-G로 표시하면 그림 1과 같다. EXPRESS-G 도표에서 실선은 엔터티의 속성 (attribute)을 나타내며, 속성 쪽에 작은 원을 그린다. 예를 들어, 엔터티 part는 5개의 속성을 가지고 있다. 그리고 굵은 실선은 서브타입과 슈퍼타입 관계를 나타내며, 서브타입 쪽에 작은 원을 그린다.

Part의 형상정보(Geometry와 Topology)는 STEP의 solid model 중에서 manifold_solid_brep을 이용하였다. 이는 여러가지 solid model 중에서 B-rep(Boundary representation)을 사용하는 것이 대상물인 part의 형상과 그것에 연관된 형상특징과 공차를 표현하는데

있어서 가장 적절하기 때문이다. 여기서 정의된 `manifold_solid_brep`은 `outer` 라는 속성을 가지며 이것은 `solid`의 외부 경계를 정의하는 엔티티인 `closed_shell`을 참조한다. `Manifold_solid_brep`은 서브타입으로 `faceted_brep`와 `brep_with_voids`를 가지며, `brep_with_voids`는 속성으로 하나 이상의 `closed_shell`의 집합으로 정의된 `voids`를 갖고 있어 내부의 빈공간을 표현할 수 있다. `Closed_shell`은 연결된 `face`들에 의해 구성되므로 이 엔티티는 속성으로 `cfs_faces`를 가지며 `face`들의 집합으로 표현된다.

하여 정의되는 형상특징은 `part`의 설계, 가공, 또는 조립 등에 필요한 정보를 제공하게 된다. `Form_feature_model`은 형상특징의 타입을 나타내기 위하여 속성 `ff`, 그 형상특징을 구성하는 요소들을 나타내기 위한 속성 `ff_elements`(예를 들어 `slot`을 구성하는 `face`들), 그리고 형상특징을 구체적으로 표현하기 위한 속성 `ff_rep`(예를 들어 `slot`의 위치, 길이, 방향 등)을 갖고있다. `Ff` 속성은 `subtype`으로 `protrusion`, `passage`, `depression` 등을 갖는 `form_feature` 엔티티로 정의됨으로써 형상특징의 타입을 나타내고 `ff_elements` 속성은 `shape_aspect`

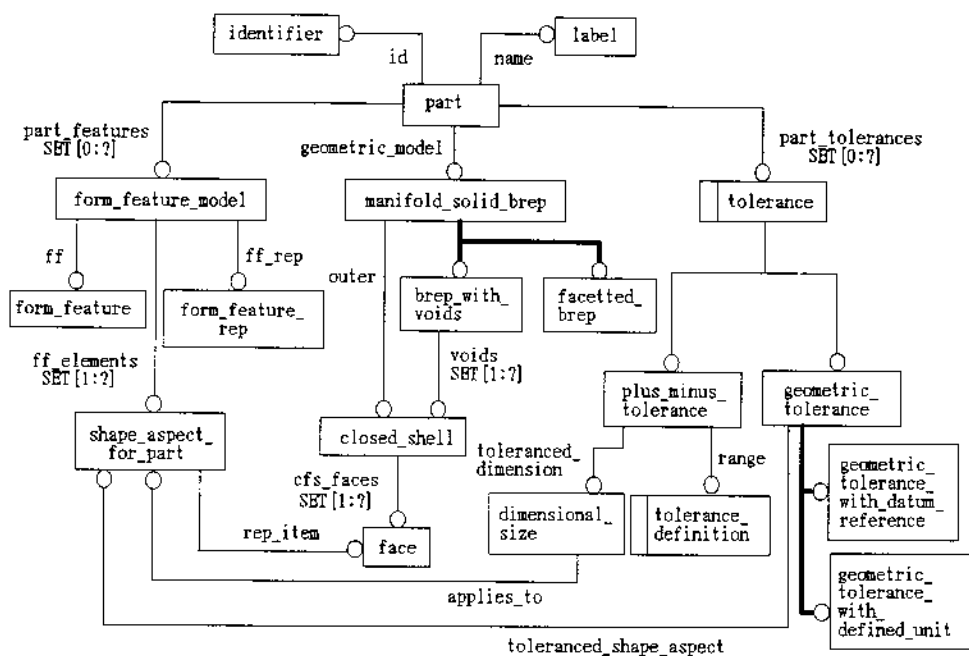


그림 1. 통합정보모델의 상위 레벨

`Part`는 여러개의 형상특징을 가질 수 있으며, 이러한 형상특징들을 효율적으로 표현할 수 있도록 본 IPIM에서는 엔티티 `form—feature—model`을 구성하였다. 이 엔티티에 의

`_for_part` 엔티티에 의해 형상정보모델인 `manifold_solid_brep`의 `face` 엔티티를 참조함으로써 형상특징정보와 형상정보를 연결하는 역할을 한다.

Part는 다수의 공차들을 가질 수 있는데, 각 공차는 STEP의 `geometric_tolerance`와 `plus_minus_tolerance` 중 하나가 된다. STEP의 파트 47에서는 공차의 대상을 `shape_aspect`로 규정하고 있으나 여기서는 엔티티 `shape_aspect_for_part`가 형상정보모델인 `manifold_solid_brep`의 `face` 엔티티를 참조하여 공차정보와 형상정보가 연결되도록 하였다.

이상에서 설명한 IPIM의 형상, 형상특징, 그리고 공차의 상세한 정보 모델은 후속 절에서 설명된다. 본 IPIM은 STEP의 관련 파트에서 필요한 엔티티들을 추출하여 구성하였으며 다음과 같은 몇가지 측면에서 수정이 이루어졌다.

1) 관련 정보의 연결을 위한 엔티티 추가: STEP에서 정의된 두개 이상의 엔티티를 결합하기 위하여 새로운 엔티티를 추가하였다. 예를 들어, 그림 1의 `part`는 5개의 엔티티를 결합시키고 있고, `form_feature_model`은 파트 48에서 정의된 `form_feature`와 `form_feature_representation`을 포함하기 위하여 추가되었다. 그리고 엔티티 `tolerance`는 파트 47에 정의된 `plus_minus_tolerance`와 `geometric_tolerance` 중 하나가 되는 `Select` 타입으로 추가되었다.

2) 필요 없는 엔티티의 삭제 또는 기존 엔티티의 수정: STEP 파트는 광범위한 분야를 대상으로 개발되어 많은 종류의 정보모델을 포함하고 있다. 응용프로토콜은 제한된 응용 분야를 대상으로 하기 때문에 STEP 파트 중에서 필요한 엔티티들만을 뽑아 재구성하여야 한다. 또한 서로 다른 파트 간의 일관성을 위하여 기존 엔티티의 수정도 필요하다. 예를 들어, 파트 42에서 정의한 `Geometric`

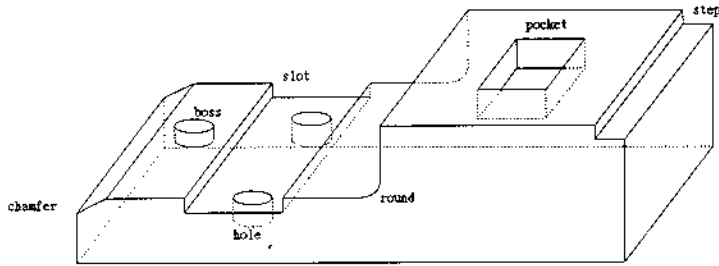
`Model` 중 `B-rep` 부분은 포함시키고 `CSG` (`Constructive Solid Geometry`)에 관련된 엔티티는 삭제하였다. 형상정보, 형상특징정보, 그리고 공차정보를 연결하는 엔티티 `shape_aspect_for_part`는 기존의 엔티티 `shape_aspect`에 형상정보 중의 `face`를 참조하는 속성 `rep_item`을 추가하여 사용하였다.

3) STEP 응용 프로그램 (`fedex`와 `data probe`)의 기능 제한에 의한 수정: 다중 계승되는 엔티티들을 하나의 `supertype`에 대해서만 계승되도록하고 나머지 `supertype`의 속성들은 첨가하였다. 속성의 `subtype`이 실제 인스턴스의 값으로 입력되었을 경우의 처리를 위한 문장이 포함되어 있는 `WHERE`, `FUNCTION`, `RULE` 등을 수정하거나 삭제하였다.

그림 2.a의 기계공작물을 이상에서 설명한 정보모델을 적용하여 `EXPRESS-I`로 표현하면 그림 2.b와 같다. `EXPRESS-I` 언어는 `EXPRESS` 모델에 따른 인스턴스 (`instance`)의 속성 값을 표시하는 언어로, 여기서 파트의 `id`는 'part1'이고 `name`은 'Example part'이다. 속성값 `geometric_model`, `part_features`, 그리고 `part_tolerances`는 각각 하나의 인스턴스 또는 인스턴스의 집합으로 정의되었다. 이들 속성값으로 사용된 인스턴스들은 후속절에서 해당 정보 모델과 함께 설명된다.

5. 형상 정보 모델

그림 3은 IPIM 중 형상정보모델에 사용되는 엔티티들과 그 엔티티들에 속해 있는 속성들 사이의 상호 연관성을 `EXPRESS-G`로 나타낸 것이다. `Manifold_solid_brep`은 닫힌 2-manifold 곡면에 의해 경계지워지는 유한적



(a) 공작물의 형상

```

part1 = part {
  id --> 'part1';
  name --> 'Example part';
  geometric_model --> @msbl;
  part_features --> { @boss, @slot, ... };
  part_tolerances --> { @pt1, @pt2, ... }; };
  
```

(b) 공작물의 EXPRESS-G

그림 2. 기계공작물의 예

이며 연결된 volume의 표현이다. 이 모델에 의하면 표현 가능한 volume의 종류에 대하여 전혀 제약을 받지 않는다.

그림 3에 도시된 바와 같이 manifold_solid_brep의 outer와 voids 속성에 참조되는 closed_shell은 solid의 경계를 정의하며 연결된 face들에 의해 구성된다. 이러한 face는 face_bound 엔티티로 정의되는 루프들의 집합에 의해 경계 지워진다. 경계루프들 중에서 적어도 하나는 face의 outer boundary 정보이어야만 한다. 본 정보모델에서는 face_bound 엔티티의 bound 속성이 참조하는 경계루프 정보를 표현하기 위하여 edge_loop 엔티티를 사용하고 있다. Edge_loop는 edge들의 리스트로 표현되며 edge는 edge_curve 엔티티를 참

조하게 된다. Edge_curve 엔티티는 속성으로 edge_start와 edge_end 및 edge_geometry 등을 가지고 있다. Edge_start와 edge_end 속성은 vertex 엔티티를 참조하고 vertex는 cartesian_point 엔티티에 의해 실제적인 3차원의 점으로 표현된다. 그리고 edge_geometry 속성은 curve 엔티티를 참조하게 되는데 curve는 다양한 subtype들을 가짐으로써 여러가지 edge의 표현을 가능하게 한다. Curve의 subtype들로 line, conic, bounded_curve, pcurve, surface_curve 등이 정의되어 있다.

경계루프들의 집합에 의한 표현방법 외에 face의 내부 형상을 정의하기 위해 geometry 정보인 곡면(surface)을 사용할 수가 있다. 이를 위하여 face의 subtype으로 face—surface 엔

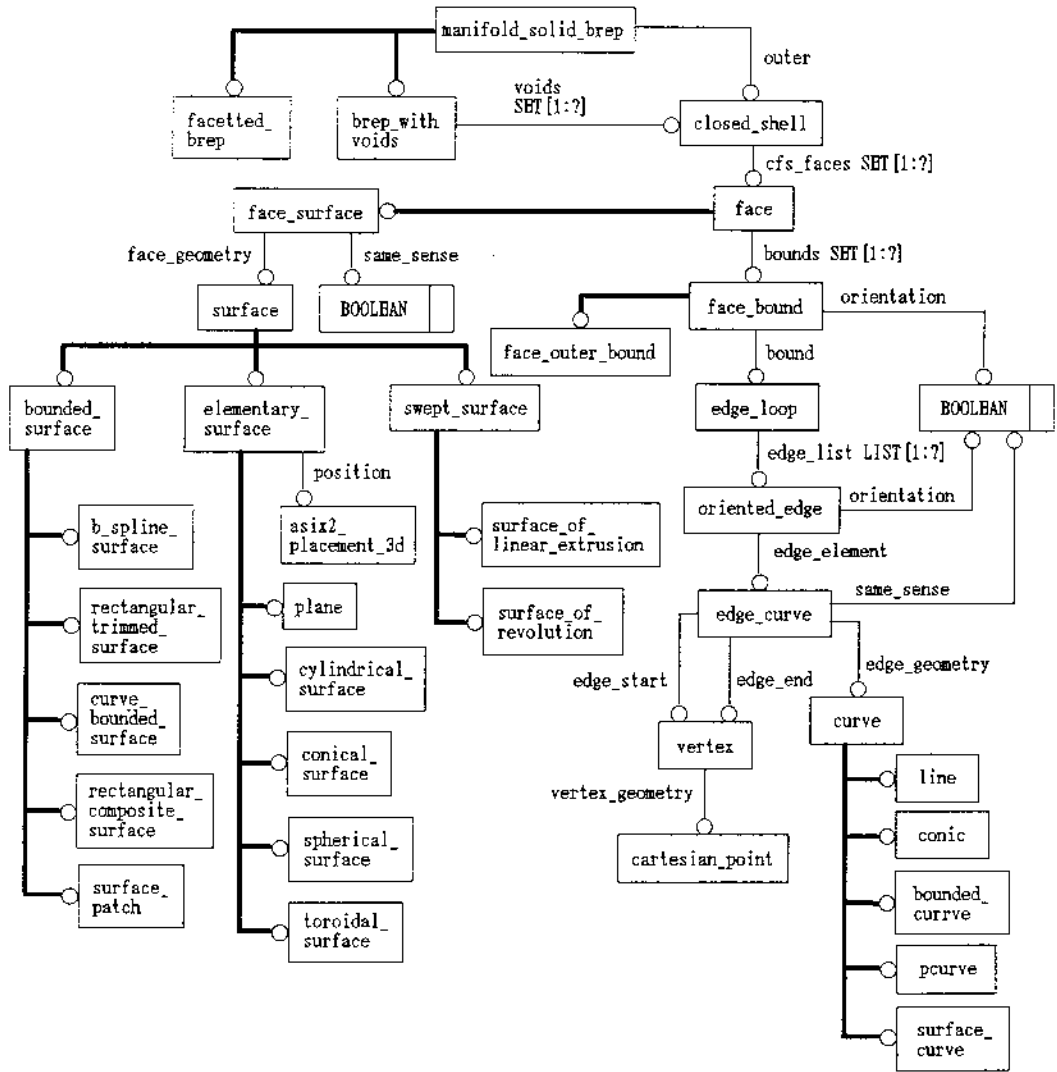
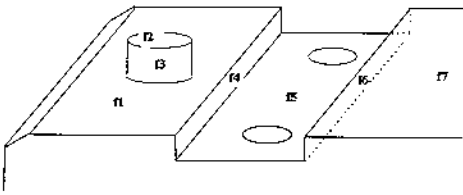


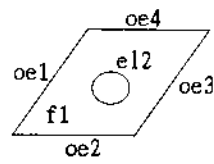
그림 3. 형상정보모델의 EXPRESS-G 표현

타티가 정의되어 있다. 이 엔티티는 속성으로 face_geometry를 가지고 있는데 이 속성에 의해 다양한 곡면 정보가 face의 내부 형상을 정의하기 위하여 사용된다. 본 정보모델에서 지원하는 곡면의 종류는 그림에서 도시된 바와 같이 크게 elementary_surface, bounded_surface, 그리고 swept_surface가 있다.

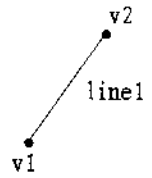
이상에서 설명한 정보모델을 이용하여 그림 2의 기계 공작물의 일부를 도시한 부분(그림 4.a, 4.b)에 대한 데이터를 EXPRESS-I로 정의하면 그림 4.c와 같다.



(a) 기계공작물의 face 구성



(b) face f1의 형상정보



```

msb1 = manifold_solid_brep {
  outer --> @cs1; };
cs1 = closed_shell {
  cfs_faces --> { @f1, @f2, @f3, @f4, @f5, @f6, @f7, ... }; };

f1 = face {
  bounds --> { @fb1, @fb2 }; };
fb1 = face_outer_bound {
  bound --> @e1;
  orientation --> TRUE;
  SUBOF face_bound; };
fb2 = face_bound {
  bound --> @e2;
  orientation --> TRUE; };
e1 = edge_loop {
  edge_list --> ( @oe1, @oe2, @oe3, @oe4 ); };
oe1 = oriented_edge {
  edge_element --> @ec1;
  orientation --> TRUE; };
ec1 = edge_curve {
  edge_start --> @v1;
  edge_end --> @v2;
  edge_geometry --> @line1;
  same_sense --> TRUE; };
v1 = vertex {
  vertex_geometry --> @cp1; };
line1 = line {
  pnt --> @cp1;
  dir --> @vector1; };

```

(c) face 의 EXPRESS-G

그림 4. 기계공작물의 face 및 형상정보

6. 형상 특징 정보 모델

그림 5는 IPIM 모델중 `ff`, `ff_elements`, 그리고 `ff_rep`을 속성으로 갖는 `form_feature_model`을 구성하고 있는 엔티티들과 그 엔티티들이 가지고 있는 속성들의 상호관계를 EXPRESS-G로 표현한 것이다. 속성 `ff`는 정의되는 형상특징의 타입을 표시하는 속성으로 `form_feature` 엔티티를 참조한다. `form_feature` 엔티티는 속성으로 형상특징의 타입을 나타내는 `ff_type`을 가지며 `subtype`으로 `volume feature`가 있다. `Volume_feature`는 기존의 `volume`에 추가되거나 혹은 제거되는 `volume`으로 해석되는 형상특징이며 이것은 다시 `depression`, `passage`, `void` 등이 속하는 `subtractive_volume_feature`와 `protrusion`, `connector`, `standalone` 등이 속하는 `additive_volume_feature`로 분류된다. 이외에 `form_feature`의 서브타입으로 정의된 `transition`과 `feature-pattern`은 본 정보모델에서는 생략되었다. 속성 `ff_elements` 형상특징의 요소가 실제로 무엇인지를 나타내는 속성이며 본 정보모델에서 추가된 엔티티인 `shape_aspect_for_part`를 참조한다. `Shape_aspect_for_part` 엔티티는 속성으로 `rep_item`을 갖는데 이 속성은 `face` 엔티티를 참조함으로써 형상특징 정보와 형상정보 (Topology와 Geometry)를 상호 연결하는 역할을 한다.

한편 `form_feature_model` 엔티티의 또다른 속성 `ff_rep`는 그림 5에 도시된 바와 같이 `form-feature-rep` 엔티티를 참조한다. `Form_feature_rep`는 공간상에서 평면곡선의 스윕핑에 의하여 추가되거나 혹은 제거되는 `volume`으로 형상특징을 기술하는 엔티티의 일

종인 `sweep_ff_rep` 엔티티를 서브타입으로 갖는다. `Sweep_ff_rep`는 속성으로 `sweep_sense`를 갖는데 이 속성은 기존의 `volume`에 스윕핑된 `volume`이 추가되는지 또는 제거되는지를 나타내며 선택적으로 지정된다. 즉, 이 정보는 특별히 기술하지 않더라도 다른 정보로부터 유추할 수 있기 때문에 반드시 지정될 필요는 없다. 이 속성의 지정될 필요성의 유무는 이 엔티티를 사용하는 응용에 달려있다. 예를 들어 응용의 영역이 기계가공에 의해 제거되는 부분을 `volume`으로 표현한다면, 모든 `sweep_ff_rep`는 `subtractive`일 것이며, 따라서 `sweep_sense` 속성은 따로 지정할 필요가 없다. 또한 `sweep_ff_rep`는 `subtype`으로 `profile_sweep` 엔티티를 가지며 이 엔티티는 `profile`이라는 속성을 가진다. 이 속성은 스윕핑되는 `profile`을 나타내며 `sweep_profile` 엔티티를 참조한다. 본 정보모델은 STEP에서 제공하고 있는 `sweep profile`들 중 `rectangle_profile`, `circle_profile`, `triangle_profile`, `circular_arc_profile`, `line_profile`, `line_plus_arc_profile` 등을 포함하고 있다. `Profile_sweep` 엔티티는 스윕핑 경로가 직선인 `linear_sweep`과 스윕핑 경로가 원호인 `circular_sweep`으로 분류된다. `Linear_sweep`은 일정한 크기의 `profile`을 가지고 있는 `constant_profile_linear_sweep`과 `profile`의 크기가 스윕핑하는 동안에 비선형적으로 변하는 `contoured_linear_sweep`으로 나누어진다. 또한, `circular_sweep`은 스윕핑 경로가 완전한 원인 `complete_circular_sweep`과 스윕핑 경로가 360도 보다 작은 원호인 `partial_circular_sweep`으로 나누어진다.

이상에서 설명한 그림 5의 정보모델은 `volume feature`와 그것에 연관된 형상의 표현

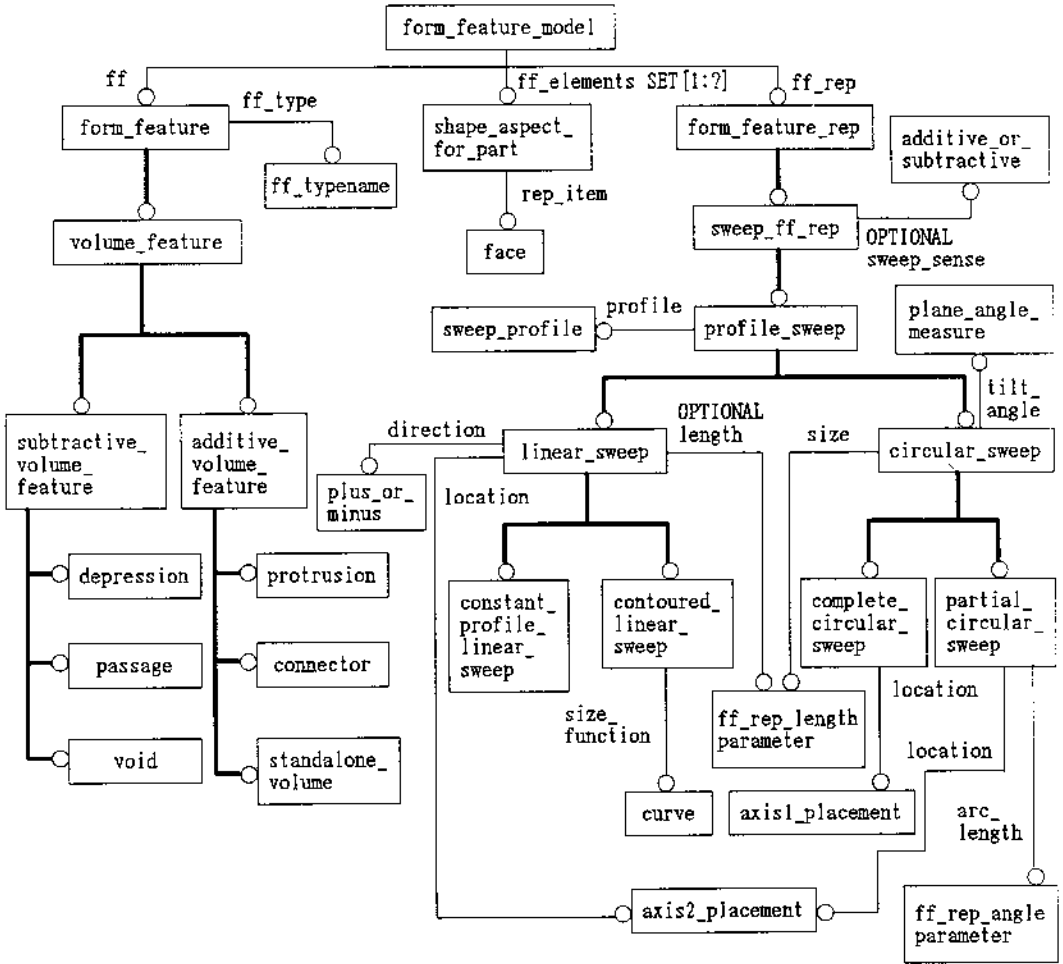


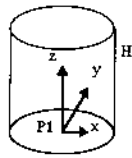
그림 5. form_feature_model의 EXPRESS-G 표현

방법을 포함하고 있다. 본 정보모델에서는 enumerative, volumetric, edge_blend 등의 다양한 표현 방법중 volumetric 표현 방법만을 사용하며, 이 방법은 다양한 sweep profile을 제공 함으로서 산업에서 필요한 많은 형상들을 표현할 수 있다. 그림 4의 기계 공작물에 포함된 형상특징 중에서 slot과 boss를 예로 들어 위에서 설명한 정보모델에 적용 시키면 다음과 같이 데이터를 정의할 수 있다.여기

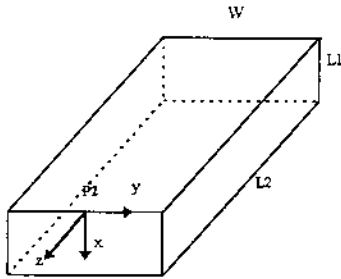
서 엔티티 form_feature_model의 인스턴스인 ffm1과 ffm2는 각각 그림 6의 boss와 slot을 나타낸다.

7. 공차 정보 모델

공차정보는 STEP의 Part47에서 정의되어 있는 plus_minus_tolerance와 geometric_tolerance를 포함한다. 그림 7은 공차정보모델에서



(a) Boss



(b) Slot

```

boss = form_feature_model {
    ff_elements --> { @s2, @s3 };
    ff_rep --> @ffr1;
    ff --> @protrusion1; };
slot = form_feature_model {
    ff_elements --> { @s4, @s5, @s6 };
    ff_rep --> @ffr2;
    ff --> @passagel };
s2 = shape_aspect_for_part {
    rep_item --> @f2; };
s3 = shape_aspect_for_part {
    rep_item --> @f3; };
s4 = shape_aspect_for_part {
    rep_item --> @f4; };
s5 = shape_aspect_for_part {
    rep_item --> @f5; };
s6 = shape_aspect_for_part {
    rep_item --> @f6; };
protrusion1 = protrusion {
    ff_type --> 'Boss';
    SUBOF additive_volume_feature; };
passagel = passage {
    ff_type --> 'Slot';
    SUBOF
        SUBOF
            subtractive_volume_feature; };
    ffr1 = constant_profile_linear_sweep {
        sweep_sense --> ADDITIVE;
        profile --> @circle_profile1;
        location --> @P1;
        length --> @length1;
        direction --> PLUS;
        SUBOF linear_sweep; };
    ffr2 = constant_profile_linear_sweep {
        sweep_sense --> SUBTRACTIVE;
        profile --> @rectangle_profile1;
        location --> @P2;
        length --> @length2;
        direction --> MINUS;
        SUBOF linear_sweep; };
    circle_profile1 = circle_profile {
        diameter --> @ff_rep_length_parameter1;
        SUBOF sweep_profile; };
    rectangle_profile1 = rectangle_profile {
        length --> @ff_rep_length_parameter2;
        width --> @ff_rep_length_parameter3;
        SUBOF sweep_profile; };

```

(c) Express-I를 이용한 인스턴스의 표현

그림 6. 기계공작물의 boss 와 slot

사용되는 엔티티들 및 그 속성들 사이의 연관성을 EXPRESS-G로 표현한 것이다. Plus_minus_tolerance는 부여되는 치수의 허용오차 범위를 정의하는 공차이며, 적용 대상과 치수를 나타내는 toleranced_dimension과 허용오차 범위를 나타내는 range를 속성으로 갖는다. 또한 속성 range로 인하여 plus_minus_tolerance는 그림 7에 도시된 바와 같이 다음의 두 가지 방법으로 기술될 수 있다.

1) 최대 허용치인 upper_bound와 최소 허용치인 lower_bound에 의한 기술

2) limits와 fits에 의한 기술(예를들어, hole이나 shaft에 적용됨)

Geometric_tolerance는 가공되는 제품형상의 기하학적 특성에 대한 허용오차 범위를 정의하는 공차이며, 적용 대상을 나타내는 toleranced_shape_aspect, 공차역(tolerance zone)의 경계를 나타내는 zone_elements, 공차의 크기를 나타내는 magnitude, 그리고 공차의 타입을 나타내는 name등을 속성으로 갖는다. 또한 geometric_tolerance는 다음과 같은 두 종류의 subtype들을 가지며 이들을 통해서 다양한 공차들을 정의할 수 있다.

1) geometric_tolerance_with_datum_reference: 하나 이상의 datum을 참조한다. 대칭도(symmetry), 흔들림공차(runout), 직각도(perpendicularity), 평행도(parallelism), 동심도(concentricity), 경사도(angularity), 위치도(position) 등이 여기에 속한다.

2) geometric_tolerance_with_defined_unit: 허용오차 범위를 나타낼 때 단위 길이(unit length)나 단위 면적(square unit area)에 대하여 정의한다. 곧음도(straightness), 윤곽도(profile), 평면도(flatness), 원통도(cylindri-

city), 진원도(circularity) 등이 여기에 속한다.

그림 8.a의 boss에 정의된 plus-minus-tolerance를 그림 7의 정보모델을 이용하여 표현하면 그림 8.c와 같다.

Manifold_solid_brep의 face를 tolerance가 참조할 수 있도록 하기 위해서 앞에서 정의된 shape_aspect_for_part를 이용한다. 위 예에서 ②는 plus_minus_tolerance의 속성인 toleranced_dimension에 의해 참조됨으로써 공차정보와 형상정보를 연결하는 역할을 한다. 그림 8.b의 parallelism 공차는 엔티티 geometric_tolerance_with_datum_reference를 이용하여 다음과 같이 정의된다.

```
pt2 = geometric_tolerance_with_datum_
reference {
  name --> 'parallelism';
  datum_system --> { @dr1 };
  toleranced_shape_aspect --> @s6;
  zone_elements --> { @tze1, @tze2 };
  magnitude --> 0.01; };
dr1 = datum_reference {
  referenced_datum --> @d1;
  precedence --> 1; };
d1 = datum {
  relating_shape_aspect --> { @s4 }; };
tze1 = tolerance_zone_element {
  relating_shape_aspect --> @s8; };
tze2 = tolerance_zone_element {
  relating_shape_aspect --> @s9; };
s4 = shape_aspect_for_part {
  rep_item --> @f4; };
s6 = shape_aspect_for_part {
  rep_item --> @f6; };
```

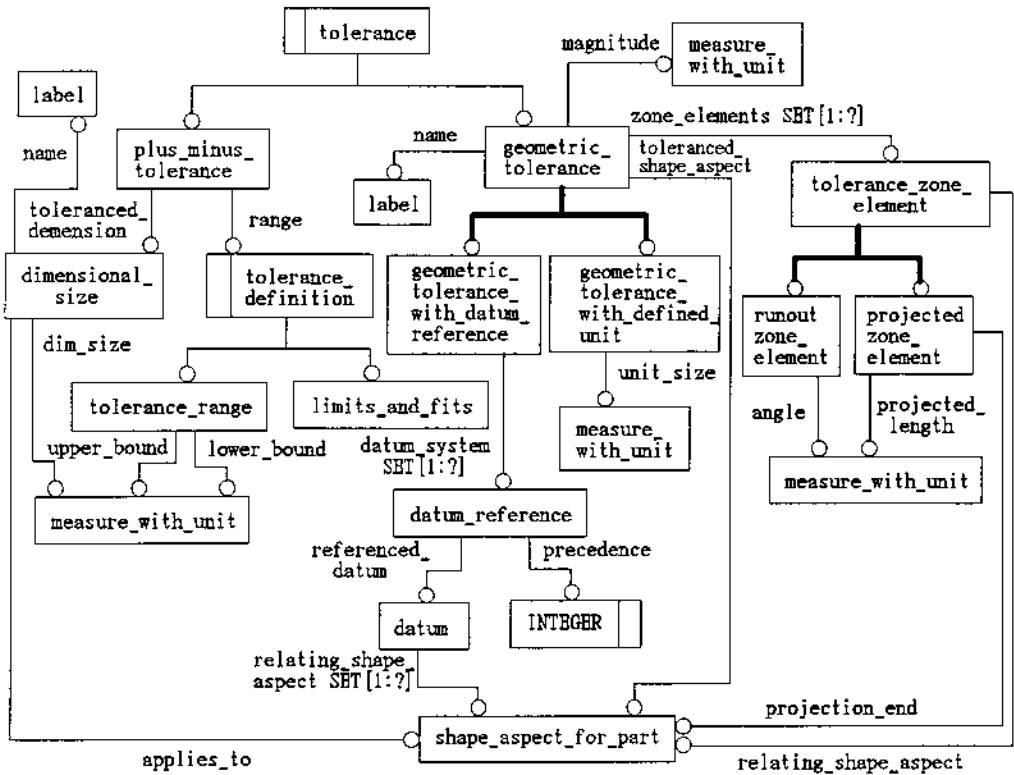


그림 7. Shape tolerance를 나타내는 정보모델의 EXPRESS-G 표현

```
s8 = shape_aspect_for_part {
    rep_item --> @f8; };
s9 = shape_aspect_for_part {
    rep_item --> @f9; };
```

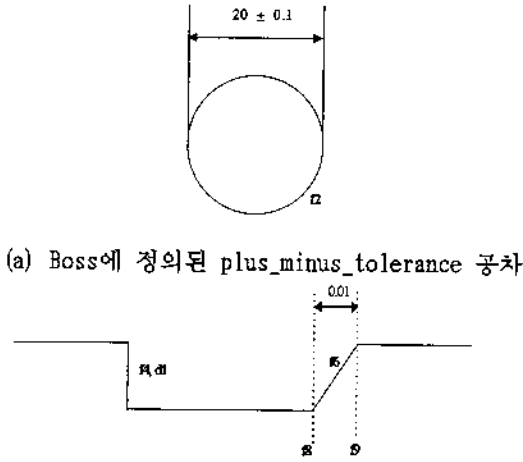
8. 시스템 통합 프로토타입 구현

8.1 프로토타입의 구조

본 연구에서 구현한 프로토타입은 표준 데이터 인터페이스 (SDAI, Standard Data Access Interface) [21]를 중심으로 구성되어 있다 (그림 9 참조). 표준 데이터 인터페이스는 응용 프로그램과 데이터베이스 또는 파일 시스템을 연결하는 기능을 수행한다. 본 프

로토타입에서는 I-DEAS를 이용하여 설계된 기계 공작물에 관한 형상 정보가 IGES 파일로부터 추출되어 STEP 파일 [22] 또는 데이터베이스 시스템 (ORACLE)에 저장된다. 형상특징과 공차 데이터는 NIST (National Institute of Standards and Technology, USA)에서 개발한 Data Probe를 수정하여 개발한 Graphic Data Probe를 통하여 형상정보에 추가된다. 이렇게 STEP 파일이나 데이터베이스 시스템에 저장된 제품 정보는 표준 데이터 인터페이스를 통하여 다양한 응용 프로그램 (예를들어, CAD, CAM, CAE, CAPP, MRP 시스템 등)이 이용할 수 있다.

그림 9에 도시된 시스템 통합 프로토타입



```

pt1 = plus_minus_tolerance {
    toleranced_dimension --> @td1;
    range --> @tr1; };
td1 = dimensional_size {
    applies_to --> @s2;
    name --> 'diameter';
    dim_size --> 20; };
tr1 = toleranced_range {
    upper_bound --> 0.1;
    lower_bound --> -0.1; };
s2 = shape_aspect_for_part {
    rep_item --> @f2; };
    
```

(c) Express-I를 이용한 공차의 표현

그림 8. 형상특징의 공차

의 구조에서 점선으로 표시된 CAD, CAE, CAM, CAPP, 그리고 MRP 시스템은 현재 프로토타입에 포함되어 있지 않지만 향후 제품 데이터를 공유할 통합 대상 시스템들이다. 이러한 시스템들을 추가로 포함하기 위하여는

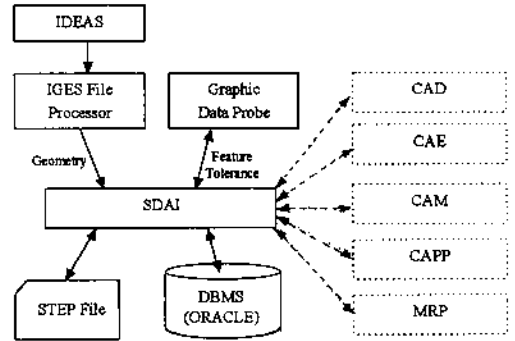


그림 9. 시스템 통합 프로토타입의 구조

우선 통합 정보 모델이 확장 되어야 한다. 프로토타입은 SUN OS와 X 윈도우 상에서 C와 C++로 개발 되었고, fedex, fedex+, Data Probe, SCL (STEP Class Library)등 NIST에서 개발한 프로그램들을 이용하였다. 프로토타입에 포함된 각 모듈의 기능은 다음과 같다.

- 1) STEP 파일 또는 데이터베이스 시스템: 프로토타입에서 정의된 제품 정보 (Product Data)를 저장하는 장소로 사용된다. STEP 파일은 파트 21에 정의되어 있는 구문 (syntax)을 이용하여 경제적으로 데이터를 저장 또는 운반한다. 데이터베이스 시스템으로는 현재 Oracle 버전 6.0을 사용하고 있으며, 다른 관계형 시스템이나 객체지향형 시스템도 추가 될 수 있다. 데이터베이스 시스템을 사용 함으로서 질의어 처리, 동시 접근 제어, 데이터 복구, 보안 등의 데이터베이스 시스템의 기능을 이용할 수 있다.
- 2) IGES File Processor: I-DEAS에서 출력된 IGES 파일을 형상 정보 모델에서 정

의된 Brep 표현으로 변환한다. 변환된 데이터는 표준 데이터 인터페이스를 통하여 STEP 화일이나 데이터베이스 시스템에 저장된다.

- 3) Graphic Data Probe: 저장된 제품 데이터를 읽어 형상 정보는 그래픽 디스플레이 하고, 제품 정보 모델에 EXPRESS로 표현된 데이터 무결성 제약조건(integrity constraints)을 확인하며, 데이터를 삭제, 추가, 또는 수정 한다. 여기서서는 제품 정보에 형상특징과 공차 정보를 입력 하는데 사용된다. 수정된 데이터는 표준 데이터 인터페이스를 통하여 다시 STEP 화일이나 데이터베이스 시스템에 저장된다.
- 4) SDAI(Standard Data Access Interface): 특정 저장 시스템 (예를 들어, 화일, RDB, 또는 OODB)에 종속되지 않는 프로그램 인터페이스를 제공한다. 인터페이스는 프로그램의 함수(function) 형태로 제공되며 응용프로그램에 의하여 이용된다.

8.2 표준 데이터 인터페이스의 이용

표준 데이터 인터페이스를 이용하여 엔티티를 생성 또는 조작하기 위해 엔티티의 모임인 모델을 이용한다. 모델은 보통 STEP 화일에 포함된 모든 엔티티 인스턴스로 구성되며 프로그램에서는 SdaiModel 클래스의 멤버가 된다. 이때 모델이 포함하는 모든 인스턴스는 SdaiEntity 클래스의 멤버가 된다. 모델의 조작은 엔티티의 리스트를 대상으로 이루어지며 화일 또는 Oracle과의 인터페이스를 위하여 SdaiModel 클래스의 멤버 함수를 이

용한다. SdaiModel 클래스의 멤버 함수의 종류와 기능은 표 1과 같다.

SdaiModel 은 모델의 고유한 이름을 기준으로 작업하게 되며 이름은 사용자에게 부여된다. 화일로부터의 입력인 경우 기본 값으로 화일의 이름을 가지며 Oracle인 경우는 모델 이름을 인자로 주어야 한다. 예를 들어 "test.spf" 라는 이름의 화일로부터 데이터를 입력받아 모델을 생성하는 경우에는 다음과 같은 코드가 필요하다.

```
SdaiModel model;
model.ReadFromFile("test.spf");
```

또는 Oracle로부터 "test"라는 이름의 모델을 추출하고자 할 경우에는 다음과 같은 형태를 가진다.

```
SdaiModel model;
model.ReadFromOracle("test");
```

이 때 모델은 고유 이름으로 "test"를 가지게 되며 변경이 필요한 경우에는 Name()을 호출 하면 된다. 예를 들어 "test"를 "Test Model"로 바꾸고자 할 경우에 다음과 같이 변경한다.

```
model.Name("Test Model");
```

모델은 이 때부터 "Test Model"이라는 이름을 통해 다른 모델들과 구별되며 같은 데이터의 모임을 가지고 있더라도 이름이 다르면 서로 구분된다. 생성이 된 모델은 엔티티의 리스트를 포함하고 있으며 GetEntityList

표 1. SdaiModel 클래스의 멤버 함수

멤버 함수명	기능
ReadFromFile	STEP 화일로부터 데이터를 읽고 SdaiEntity 리스트를 형성한다. 인자로 화일명이나 SdaiFile 클래스를 지정할 수 있으며 실제적인 처리는 SdaiFile 클래스가 담당한다.
ReadFromOracle	Oracle로부터 지정된 모델명을 가지는 데이터를 추출한다. 인자로 모델 이름을 지정한다.
WriteToFile	저장된 데이터를 STEP 화일의 형태로 출력한다.
WriteToOracle	저장된 데이터를 Oracle로 저장한다.
UpdateOracle	Oracle내의 데이터를 갱신한다. 이때 인자로 모델 이름을 지정하며 동일 이름의 모델이 Oracle내에 존재하지 않을 경우 WriteToOracle과 같은 효과를 가진다.
Remove	저장된 데이터를 모두 삭제한다.
GetEntityListH	저장된 SdaiEntity의 리스트를 반환한다.
GetEntity	저장된 데이터로부터 일치하는 데이터를 반환한다. 이 때 인자는 엔티티의 ID나 이름을 지정할 수 있으며 이름으로 지정된 경우는 리스트를 반환한다.
Name, Description	모델의 고유 이름 과 설명에 관여하는함수로 Oracle과의 인터페이스에서 인자로 작용한다.

() 함수를 호출하여 얻을 수 있다. 사용자는 얻어진 엔티티 리스트로부터 엔티티 인스턴스를 조작할 수 있다. 엔티티 리스트를 이용한 작업의 일반적인 형태는 다음과 같다.

```
SdaiEntityListH entList =
    model.GetEntityList();
SdaiEntityH ent = entList->Rewind();
while(ent) {
    // 단일 엔티티에 대한 조작에 필요한 프
    // 로그램 삽입
    ent = entList->Next();
}
```

생성 및 조작이 완료된 모델은 STEP 화일이나 Oracle로 저장이 가능하며 이 때는 WriteToFile(), WriteToOracle() 함수를 이용하게 된다. Oracle의 경우는 모델의 이름을 기본 인자로 포함하고 있으며 화일로의 출력인 경우는 출력될 화일의 이름을 지정해야 한다. 이미 같은 이름의 모델이나 화일이 존재할 경우 화일출력은 기존의 화일을 덮어쓰게 되며 Oracle의 경우는 에러를 발생한다. Oracle로 같은 이름의 모델을 출력하고자 하는 경우는 UpdateOracle() 을 이용하면 된다. UpdateOracle()은 Oracle내에 존재하는 모델을 검색하여 해당하는 모델이 존재하지 않는 경우는 새로운 모델을 생성하며 해당 모델이 존

재하는 경우는 모델내의 데이터를 갱신하고 자 하는 데이터로 바꾸어 준다. 추가적인 기능으로, 작업의 대상이 되고 있는 모델에 대한 설명문을 함수 description()을 이용하여 첨가 할 수 있는데 이는 Oracle로의 저장이나 입력인 경우에만 제공된다.

8.3 제품 데이터의 예

앞에서 정의한 IPIM을 설명하기 위하여 사용한 기계 공작물의 예를 모아 그림 10에 도시하였다. 이 그림에서 엔티티의 인스턴스들

은 괄호안에 표시 하였고 인스턴스가 집합인 경우 대괄호 ([])로 표시하였다. 속성 정보가 추가되는 경우 인스턴스의 ID에 밑줄을 그어 표시하였고, 그림을 알아보기 쉽게 하기 위하여 많은 데이터가 생략되었다. 이 그림을 통하여 형상 특징과 공차 정보가 형상 정보 엔티티인 face를 통하여 서로 연결되는 것을 알 수 있다.

표 2 는 앞에서 설명한 제품 데이터의 예를 Graphic Data Probe를 통하여 입력한 후 STEP 파일로 저장한 예이다.

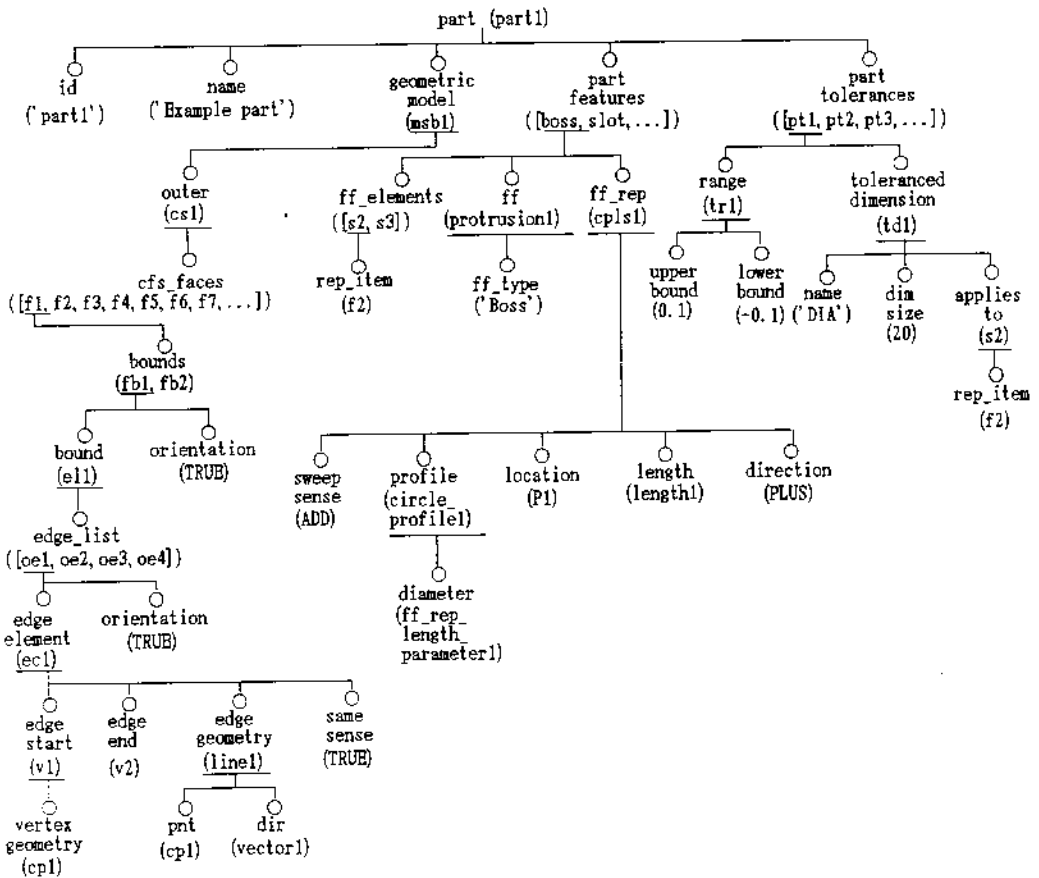


그림 10. 정보 모델에 기계 공작물의 일부에 대한 데이터를 입력한 예

표 2. STEP 파일로 저장된 제품 정보

TEP_WORKING_SESSIONHEADER;	#31=VECTOR(\$,\$,\$);
FILE_DESCRIPTION(\$,\$);	#32=CONSTANT_PROFILE_LINEAR_SWEEP(ADD.,#41,#
FILE_NAME(\$,'1995-04-02 19:24:55',\$,\$,\$,\$);	42, #43,PLUS.);
FILE_SCHEMA(\$);	#33=SHAPE_ASPECT_FOR_PART(#9);
ENDSEC;	#34=SHAPE_ASPECT_FOR_PART(#10);
DATA;	#35=PROTRUSION('Boss');
#1=PART('part1', 'Example part', #2, (#3, #4), (#5, #6);	#36=CONSTANT_PROFILE_LINEAR_SWEEP
#2=MANIFOLD_SOLID_BREP(#7);	(SUBTRACT., #45, #48, #49, MINUS.);
#3=FORM_FEATURE_MODEL((#33, #34), #32, #35);	#37=SHAPE_ASPECT_FOR_PART(#11);
#4=FORM_FEATURE_MODEL((#37, #38, #39), #36, #40);	#38=SHAPE_ASPECT_FOR_PART(#12);
#5=PLUS_MINUS_TOLERANCE(#50, #51);	#39=SHAPE_ASPECT_FOR_PART(#13);
#6=GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE	#40=PASSAGE('Slot');
(#39, 'Parallelism' (#56, #57), #58,\$);	#41=CIRCLE_PROFILE(#42);
#7=CLOSED_SHELL((#8, #9, #10, #11, #12, #13, #14));	#42=FF_REP_LENGTH_PARAMETER(\$, FULL.);
#8=FACE((#17,\$));	#43=AXIS2_PLACEMENT_D(\$,\$,\$);
#9=FACE(\$);	#44=FF_REP_LENGTH_PARAMETER(\$, HALF.);
#10=FACE(\$);	#45=RECTANGLE_PROFILE(#46, #47);
#11=FACE(\$);	#46=FF_REP_LENGTH_PARAMETER(\$, FULL.);
#12=FACE(\$);	#47=FF_REP_LENGTH_PARAMETER(\$, FULL.);
#13=FACE(\$);	#48=AXIS2_PLACEMENT_3D(\$,\$,\$);
#14=FACE(\$);	#49=FF_REP_LENGTH_PARAMETER(\$, FULL.);
#15=FACE(\$);	#50=DIMENSIONAL_SIZE(#33, 'Diameter', #52);
#16=FACE(\$);	#51=TOLERANCE_RANGE(#53, #54);
#17=FACE_OUTER_BOUND(#20, T.);	#52=MEASURE_WITH_UNIT(20, #55);
#18=FACE_BOUND(#19, T.);	#53=MEASURE_WITH_UNIT(0.1, #55);
#19=EDGE_LOOP(\$);	#54=MEASURE_WITH_UNIT(-0.1, #55);
#20=EDGE_LOOP((#21, #22, #23, #24));	#55=SI_UNIT(.MILLI., METRE.);
#21=ORIENTED_EDGE(#25, T.);	#56=TOLERANCE_ZONE_ELEMENT(#60);
#22=ORIENTED_EDGE(\$,\$);	#57=TOLERANCE_ZONE_ELEMENT(#61);
#23=ORIENTED_EDGE(\$,\$);	#58=MEASURE_WITH_UNIT(0.01, #55);
#24=ORIENTED_EDGE(\$,\$);	#59=DATUM_REFERENCE(#62, 1);
#25=EDGE_CURVE(#26, #27, #28, T.);	#60=SHAPE_ASPECT_FOR_PART(#15);
#26=VERTEX(#29);	#61=SHAPE_ASPECT_FOR_PART(#16);
#27=VERTEX(#30);	#62=DATUM((#37,\$));
#28=LINE(#29, #31);	ENDSEC;
#29=CARTESIAN_POINT(\$);	END-STEP_WORKING_SESSION;
#30=CARTESIAN_POINT(\$);	

9. 결론

본 연구에서는 STEP에서 개발된 공동 자원인 파트 42 (Geometric and topological

representaiton), 파트 47 (Shape variational tolerances), 그리고 파트 48 (Form features) 등을 이용하여 기계공작물의 설계과정에서 생성되는 Geometry, Feature, 그리고 Tolerance

를 관리하는 통합정보모델(IPIM)을 개발하였다. STEP 화일 또는 데이터베이스 시스템에 저장된 기계공작물에 대한 제품정보는 다른 응용프로그램 (예를들어, 공정계획 시스템, 조립 시스템, 또는 해석 시스템)에 의하여 이용되고 추가되는 설계정보 또는 생산정보는 다시 데이터베이스 시스템에 저장된다. 이러한 과정을 거쳐서 CAD/CAM/CAE를 연결하는 통합생산시스템(CIM)을 구축할 수 있다.

본 프로토타입에서 사용한 IPIM은 STEP에서 개발중인 응용프로토콜 (예를들어, Explicit draughting, Configuration controlled design, Mechanical design using boundary representation, Electronic printed circuit assembly, design and manufacture 등)과 유사한 기능을 수행한다. 조선, 자동차, 기계설계 등과 같은 각각의 응용분야에서 사용할 수 있는 응용프로토콜이 ISO에서 개발되고 있고 이것은 향후 해당 산업분야에 적용되어 자동화 시스템의 정보교환에 사용될 것이다. 본 연구에서는 현재 정의된 STEP의 표준을 이용하여 프로토타입을 개발함으로써 STEP을 이용한 통합시스템의 모델을 제시하였다. 프로토타입에서 사용한 EXPRESS 모델을 적절히 수정하여 다양한 분야에 적용할 수 있으며, 이러한 EXPRESS 정보모델은 사용자의 소프트웨어 환경에 따라 NIAM, SQL, IDEFIX 등의 다른 정보모델로 변환하여 사용할 수 있다.

STEP을 이용한 시스템 통합은 향후 STEP의 개발이 완료되고 주요 CAD/CAM 또는 소프트웨어 벤더들이 이를 지원하여 다양한 응용프로그램들과 데이터베이스 시스템이 수정

없이 직접 제품 데이터를 교환할 때 그 궁극적인 목적을 달성 할 수 있다. 이와 같이 STEP이 산업 표준으로 자리 잡기 전에는 각 기업체에서 부분적으로 판매되는 STEP 프로그램들을 구입하고 기존 응용프로그램을 수정하여 통합 시스템을 구축할 수 있다. 이러한 경우 IGES 변환기, 표준 데이터 인터페이스, 그리고 데이터베이스 시스템 등이 공통적으로 필요한 모듈이 된다. 그리고 통합 정보 모델은 각 기업에서 사용하는 응용 프로그램들을 고려하여 독자적으로 개발 하여야 한다.

참 고 문 헌

- [1] D. N. Chorafas and S. J. Legg, The Engineering Database, Butterworths, London, 1988.
- [2] ISO TC184/SC4/N141 (WG3/N169), STEP Part 42: Integrated Generic Resources: Geometric and topological representation, August 15, 1992.
- [3] ISO TC184/SC4/WG4 N60, STEP Part 43: Integrated Generic Resources: Representation structures, August 15, 1992.
- [4] ISO TC184/SC4, STEP Part 101: Application Resources: Draughting, December, 1993.
- [5] ISO TC184/SC4, STEP Part 104: Application Resources: Finite element analysis, March, 1993.
- [6] G.M. Nijssen and T.A. Halpin, Conceptual Schema and Relational Database Design: A Fact Oriented Approach, Prentice Hall,

- Englewood Cliffs, NJ, 1989.
- [7] C.J. Date, A Guide to the SQL Standard, Addison-Wesley, 5th edition, 1990.
- [8] AFWAL/MLTC, Wright-Patterson AFB, OH., Integrated Information Support System (IISS), Vol. V: Common Data Model Subsystem, Part 4: Information Modeling Manual - IDEF1X, 1985, Report Number: AFWAL-TR-86-4006, Volume V.
- [9] P. Chen, "The Entity-Relationship Model - Towards a Unified View of Data", ACM TODS, 1(1), March 1976.
- [10] ISO TC184/SC4/WG5, STEP Part 11: EXPRESS Language Reference Manual, July 1, 1994.
- [11] J. Vuoskoski, et. al., "Using EXPRESS in High Energy Physics Research Environment", in P. Wilson, editor, EUG'94 - 4th International EXPRESS User Group Conference, Greenville, South Carolina, October 1994.
- [12] T. Divoux, et. al., "Using EXPRESS: to Define MMS Manufacturing Communication", in P. Wilson, editor, EUG'94 - 4th International EXPRESS User Group Conference, Greenville, South Carolina, October 1994.
- [13] F. Schoenefeld and C. Bohm, "Using EXPRESS Database Technology for Accessing NCBI Genomic Data", in P. Wilson, editor, EUG'93 - Third International EXPRESS User Group Conference, Berlin, Germany, October 1993.
- [14] M. Eberl, "Integrated system for furniture manufacturing", Proceedings of International Conference on Data and Knowledge Systems for Manufacturing and Engineering, Hong Kong, May, 1994, pp. 383-388.
- [15] H. Mantyla, Solid Modeling, Computer Science Press, 1988.
- [16] J. Shah, P. Sreevalsan, and A. Mathew, "Survey of CAD/feature-based process planning and NC programming techniques", Computer-Aided Engineering Journal, February 1991, pp. 25 - 33.
- [17] L. Qiao, et. al., "A PDES/STEP-based product data preparation procedure for computer-aided processing planning", Computers in Industry, 21, 1993, pp. 11 - 22.
- [18] ISO TC184/SC4/WG3, STEP Part 47: Integrated Generic Resources: Shape variation tolerances, September 10, 1993.
- [19] ISO TC184/SC4, STEP Part 48: Integrated Generic Resources: Form features, January, 1992.
- [20] ISO TC184/SC4, STEP Part 41: Integrated Generic Resources: Fundamentals of product description and support, August 15, 1992.
- [21] ISO TC184/SC4/WG7, STEP Part 22: Standard Data Access Interface, 1993.
- [22] ISO TC184/SC4, STEP Part 21: Clear text encoding of the exchange structure, 1993.