論文 한국정밀공학회지 제10권 제4호 1993.12
Journal of the Korean Society of Precision Engineering Vol. 10, No.4 Dec. 1993.

PARAMETRIC DESIGN을 위한 自動設計모듈 生成

李 錫 熙*, 潘 甲 守**

Automated Design Module Generation System for Parametric Design

Seok-Hee Lee*, Kab-Soo Bahn**

ABSTRACT

An davanced method for the automatic generation of parametric models in computer-aided design systems is required for most of two-dimensional model which is represented as a set of geometric elements, and constraining scheme formulas. The development system uses geometric constraints and support of topology parameters from feature recognition and grouping the design entities into optimal ones from pre-designed drawings. The aim of this paper is to present guidelines for the application and development of parametric design modules for the standard parts in mechanical system, the basic constitutional part of mold base, and other 2D features.

1. 序 論

生産體制의 多樣化로, 국제 경쟁력의 향상, 多樣한 消費者의 要求에 따른 多品種少量生産體制 등 生産에 있어서의 당면한 問題들을 해결하며 생산성을 높이기 위해서, 사람의 참여를 줄이고, 발전된 機械가 대신하여 작업을 수행할 수 있게 하는 것이 필요하다. 즉 사람은 자신만의 獨特한 創造的인 思考를 기술의 발전 등에 利用하며 單純反復的인 작업에 消費하는 시간을 줄여야 한다. 이러한 추세는 실질적인 하드웨어의 개발 뿐만 아니라 CAD(Computer Aided Design) 및 CAM(Computer Aided Manufacturing)과 兩者를 연결하는 CAPP(Computer Aided Process Planning) 분야에서도 적용되어 이 分野에서 많은 연구가 진행되어 왔다.

특히 제품의 生産에 있어서 工程計劃 작성, 切削順序 결정, 切削工具 선정, 切削條件 선정, 切削經路 시뮬레이션, NC 프로그램 自動生成 등 生産의 自動化에 필요한 技能 등을 얻기 위해서 CAD/CAM 統合 시스템의 구축이 必要한데, 이러한 一連의 작업의 기본 구조는 데이타베이스(D/B) 및 知識베이스(K/B)의 調和로 이루어진다고 볼 수 있다. CAD/CAM 시스템에서 圖面情報를 加工情報로 變換하는 工程을 自動化하기 위해서는 部品의 設計情報를 CAD 시스템으로부터 이끌어 내기 위한 CAD 인터페이스 問題가 해결되어야 할 것이다. 形狀認識(feature recognition)이란 이런 問題를 解決하기 위해 部品의 모양을 정의하는 點, 線, 面들의 幾何學的인 要素들과 이들 간의 相互 關聯情報들로부터 加工에 必要한 形狀情報들을 認識하는 것이다[1, 2]. 回轉形狀 부품의 境遇 形狀認識에 관한 硏究는 Syntactic Pattern Recognition을 利用한 硏究가 가장 代表的인 方法[3, 4, 5]인데, 이 方法은 입력

* 釜山大學校 生産機械工學科, 기계기술연구소 연구원
** 釜山大學校 大學院

形狀 및 認識方法의 制限으로 인해 複雜한 形狀을 가진 部品認識에는 適用할 수 없는 短點을 가지고 있다[1]. Wang가 Li가 2次元 圖面에서 上部 輪廓線(upper-half profile)을 對象으로 가공의 情報를 얻는 方法을 제시하고 있으나 非回轉形狀에서는 더욱 발전된 方法이 요구된다[6, 7]. 完全回轉形狀 部品(rotational part without deviation)을 對象으로 自動認識에 관해, IGES(Initial Graphic Exchange Specification) 화일을 利用하여 形狀認識力을 높이기 위한 認識 方法論도 제시하고 있다[8]. 設計自動化를 위해서 媒介變數 設計방식(Parametric Design : 이하 PD)의 概念을 利用한 硏究[9-12]는

(1) 이미 設計된 類似形狀에서 원하는 部分을 變數처리한 치수變數(Dimension Parameter)의 概念으로 設計 프로그램을 구현하는 方式

(2) 기본 모델을 畫面상에서 대화식 設計方式으로 設計하고 그 결과에 關聯된 幾何學的 위상과 拘束 條件들을 自動 生成하는 방식을 구분하면서 접근해야 한다고 Roller[9]는 언급하고 있는데 본 硏究에서는 알고리즘에 제시하며 認識한 形狀을 GT(Group Technology)기법[13, 16]으로 類似한 形狀의 分類에 의해 이를 函數化 하여 形象을 처리할 수 있도록 形象函數가 自動으로 生成된다. 이 函數는 規格化된 形狀의 設計變數뿐만 아니라 規格化되지 않은 形狀의 設計變數도 PD 概念으로 設計될 수 있도록 시스템이 構成되어 있다.

機械工業 분야에서 일반적으로 많이 使用되고 기초가 되는 標準機械 部品과 조립도 設計시, 形狀函數를 운용함으로써 CAD 작업에서의 효율성을 높인다. 또한 실제 加工데이타를 生成시킬 때, 設計자가 기입한 치수와 CAD 시스템이 認識하고 있는 실제 情報가 오차가 있으면 정확한 加工情報를 얻을 수 없다. 예를 들어 볼트의 길이가 기입된 치수는 200mm인데 CAD 시스템이 認識하고 있는 길이는 150mm로 되어있다면 어느 치수로 加工情報를 추출해야 할지 판단하기 어렵다. NC 데이타 生成 전문가는 이런 問題를 방지하기 위해 設計자에게 철저한 도면 檢證을 요구하지만 쉽지 않다. 形狀 D/B를 이용하여 自動으로 形狀이 정의되면 이런 問題는 해결된다. 形狀認識 데이타는 AutoCAD 시스템에서 제공되는 데이타베이스(AutoCAD Drawing Database : ADD)를 利用하였으며, 使用 언어는 ADS (Auto CAD Development System)환경에서 使用되는 Metaware High C와 形狀函數는 BORLAND C++을 使用하여 Auto LISP 언어의 모듈화된 技能을 自動生成한다.

2. 파라메타 방식의 設計개념 및 CAD 데이타

既存의 設計된 形狀에서 정의된 變數의 변경으로 새로운 概念의 設計를 할 수 있는 PD 方式의 設計概念은 다음과 같다.

2.1 PD 方式의 設計 概念

파라메트릭 모델링(Parametric Modeling)은 사용자가 製品모델의 치수를 어떤 숫자로 고정시키지 않고 유동적으로 정의할 수 있도록 하는 것이다. 이 境遇에 치수는 수식의 형태로 주어진다. 임의의 치수를 수정하였을 境遇에, 幾何學的 모델과 치수결정 계획

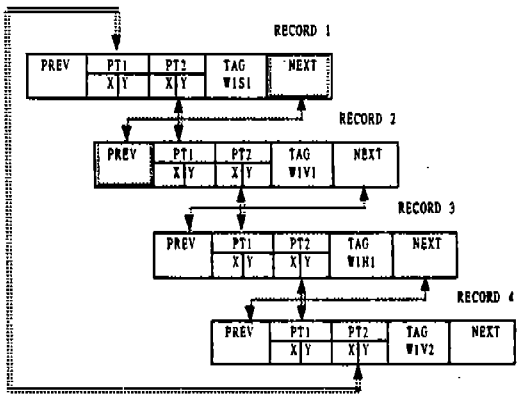


Fig. 1 Pointer list structure of lines

(dimensioning scheme)에 대해서 양 방향의 데이터 연결고리에 의하여 그와 關聯된 부위의 치수가 自動的으로 수정되고 모델 또한 自動的으로 변경된다. George[17]은 직사각형의 形狀에서 직선간의 위상학적인 관계를 규명하기 위하여 그림 1과 같이 Record1의 終點이 Record2의 始作點이 되고 Record2의 終點이 Record3의 始作點이 되는 식의 데이타 구조를 이용하였다. 이러한 方式은 幾何學的으로 形狀이 類似하고 치수만 다른 部品群의 設計에 있어서 상당한 효율의 증대를 가져온다. 또한 設計에서 加工에 이르는 싸이클에서 전형적으로 要求되는 設計의 性能向上에 신속하고 쉽게 適用된다. 그리고 概念設計 단계에서는 치수가

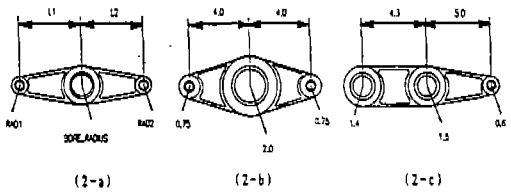


Fig. 2 Example of parametric design

결정되지 않는 境遇도 많이 있으므로 이점이 많다[9]. 그림 2-a는 Rocker Arm의 設計시 L1, L2, RAD1, RAD2 및 BORE-RADIUS를 設計變數로 사용하여 設計한 모양을 나타내고 있다. 여기서 設計變數들의 변경으로 2-b와 2-c와 같은 形狀을 자유롭게 再設計할 수 있다.

2.2 形狀要素(Entity)의 屬性分類

CAD로서 행하는 設計작업은 어떤 좌표 위치에 形狀要素를 위치 시키는 것이 대부분이다. 여기서 形狀要素라는 것은 圖面상에 삽입되어지는 미리 결정된 것으로써 線, 圓弧, 圓 등은 形狀要素의 구성要素이다. 문자 및 치수도 使用되는 圖面要素들이다. 각각의 圖面要素에 대하여 색상 및 線의 형태를 지정할 수 있다. 形狀을 分類하여 표현하면 다음과 같다.

Table 1. Entity Classification of CAD Database

Group code.	Variable name	Meaning	Entity type	Variable type
0	e-type	Entity type	L, A, C, B	String
2	b-name	Block name	B	String
8	l-name	Layer name	L, A, C, B	String
10	s-point	SP of Entity	L, B	Point
11	e-point	EP of Entity	L	Point
40	radius	Radius	A, C	Real
x	a-sp	SP of Arc	A	Point
x	a-ep	EP of Arc	A	Point
50	s-ang	SA of Arc	A	Real
51	e-ang	EA of Arc	A	Real

(NOTE)

1) Entity : L=Line, A=Arc, C=Circle, B=Block
2) Data type : SP=Start point, EP=End point, SA=Start angle, EA=End angle, x=not defined

2.3 層(Layer)의 활용

設計 圖面상의 部分에 대하여 層을 부여 할 수 있고 필요한 만큼의 層으로 정의할 수 있다.
이와 같은 多層化 槪念은 많은 設計圖面 작업에 있어서 투명 겹침(Transparent Overlay) 현상과 類似하다. 多層化는 圖面의 연관된 部分들을 분리하거나 또는 서로의 조합으로 보거나 그릴 수 있게 해준다. 예를 들면 첫번째 層에는 마루 바닥에 관한 圖面을, 두번째 層에는 전기 배선에 관한 圖面을 겹쳐서 볼 수 있고 또한 마루 圖面에 수도관의 圖面을 겹쳐서도 볼 수가 있다.

Table 2. Layer name, Line type and it's Color

Layer Name	Line Type	Color
"H"	HIDDEN	Yellow
"S"	CONTINUOUS	Green
"C"	CENTER	Red
"P"	PHANTOM	Blue
"0"	CONTINUOUS	White

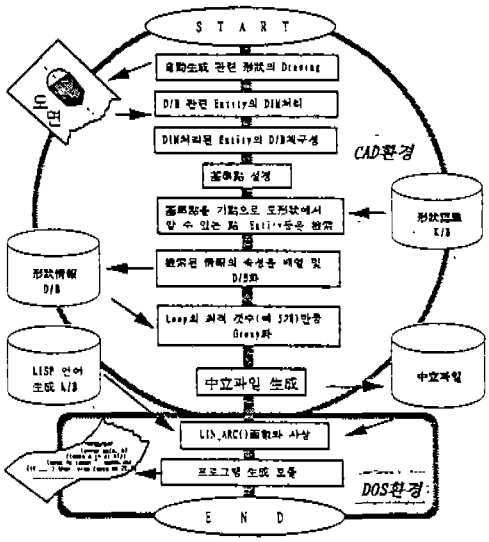
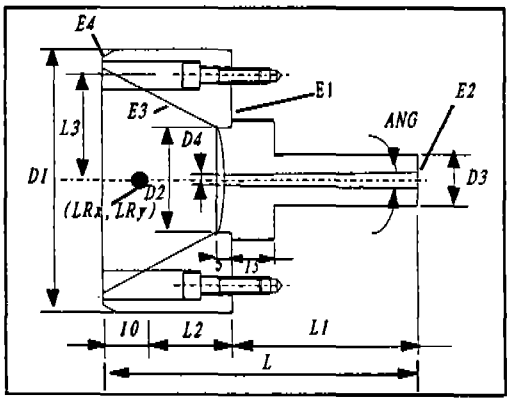


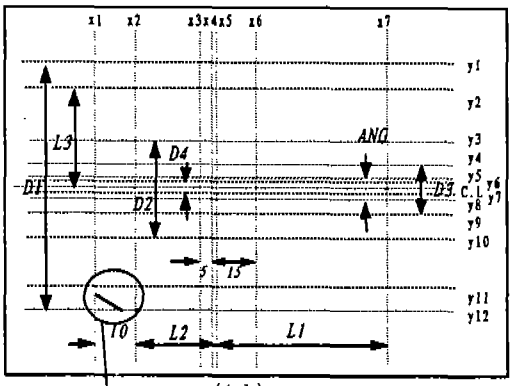
Fig. 3 System flow chart

3. 硏究의 내용 및 方法

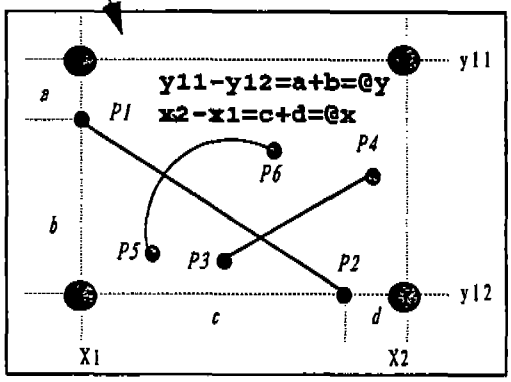
旣存의 設計된 形狀을 利用하여 같은 部品 및 類似形狀 部品을 設計할 때 유용한 해결책을 제시할 본



(4-a)



(4-b)



(4-c)

Fig. 4 A Sample Drawing and It's Known Points

研究의 전체적인 시스템 흐름도는 그림 3과 같다. 여기서 입력사항은 形狀 파일이며 形狀이 입력되면 形狀에서 變數화 시키고 싶은 形狀요소는 變數化하기 위해 形狀認識이 가능하도록 치수(Dimension)로 처리한다. 또한 치수로 처리된 變數는 데이타 파일에서 管理되도록 D/B화 한다. 그림 4-a는 본 연구의 對象形狀이다. 對象形狀의 基準點은 形狀의 基準이 되는 點을 설정한다. 基準點에서 치수 D/B를 이용하여 도형 상에서 기준점에서 치수변수의 상대위치로 정의된 알수 있는 點(Known Point : KPS)을 찾아내어 D/B로 管理한다. 그림 4-b는 치수變數를 이용하여 x-y 평면상에 KPS를 모두 나타낸 것이다. 여기서 알 수 있는 點은 모두 12×7개가 존재하며 KPS 사이에 존재하는 모르는 點들은 PD 기법에 의해 비율로 정의된다. 中心선과 KPS를 基準으로 모든 形狀을 複合라인 群으로 분류한 다음, 中立파일을 生成한다. 中立파일을 LIN-ARC( )函數와 연관시킨다. 그리고 CAD 환경에 맞는 시스템 變數를 설정하여 LISP 原始파일을 최종적으로 生成한다.

3.1 研究의 내용

(1) 既存의 設計된 形狀에서 既처리된 變數와 타 形狀要素와의 幾何學的 拘束관계

(2) 形狀要素의 분석 및 形狀群(cluster)으로 分類하여 특정갯수의 形狀要素 단위로 그룹화

(3) 既存의 形狀分類 函數와의 연관化

3.2 研究의 方法

研究의 내용에서 정의한 3가지에 대한 方法을 설명하면 다음과 같다.

3.2.1 정의된 變數의 다른 形狀要素와의 幾何學的 拘束條件

既存의 設計된 變數는 形狀의 주요한 인자를 결정하는 것으로서 形狀을 유지하는데 중요한 역할을 한다. 이들 變數들은 치수를 대표하는 역할을 하고 設計者의 設計方法에 대한 基準과 檢證의 중요한 역할을 한다.

그림 5는 設計形狀의 예를 나타낸 것으로 分岐가 되는 形狀 및 단일 形狀要素의 연결된 형태인 分岐가 없는 形狀으로 구분할 수 있다. 使用되는 形狀要素는 線, 圓弧, 曲線 등으로 구분하여 分類할 수 있으며 일반적인 形狀에 使用되는 曲線은 圓弧의 연속된 形狀이라 가정한다. DXF나 IGES 變換 파일을 利用하여 檢索할 때에는 해당되는 形狀要素를 檢索하여 연결된 것을 찾아낸다. 形狀이 복잡할수록 이 方法의 使用은

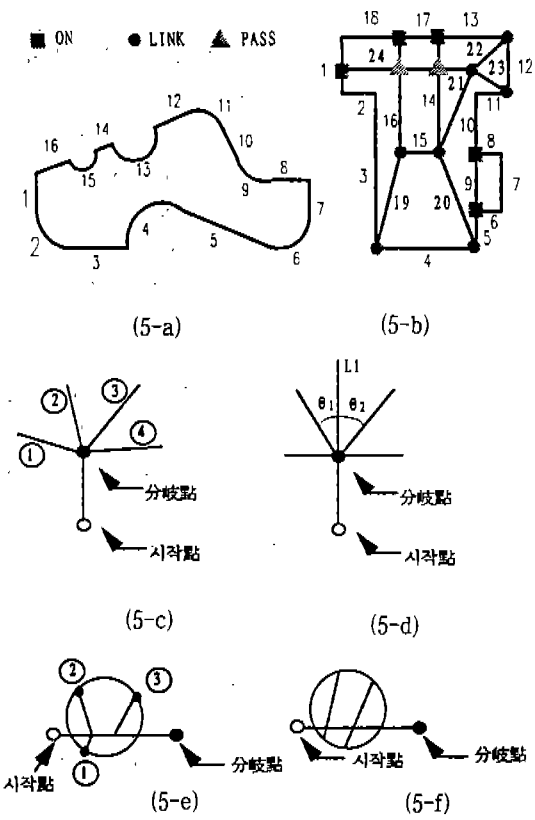


Fig. 5 Example of design feature

檢索에 소요되는 시간과 分岐 여부의 판단에 혼란의 소지가 있다. 본 연구에서 적용한 認識方法은 畵面에서 직접 시각적인 認識의 도움을 받아서 관심의 對象이 되는 영역을 미리 설정해서 그 영역안에 존재하는 形狀要素 만을 檢索함으로써 檢索에 소요되는 시간을 최소화 할 수 있다. 또한 난해한 分岐의 여부를 檢索하는 알고리즘의 정립이 용이하다.

(1) 分岐가 없는 形狀의 認識

그림 5-a는 形狀要素의 연결이 分岐되지 않는 境遇를 나타내고 있다. 始作點과 終點의 좌표값을 고려하여 마치 自動車가 미지의 외길을 따라가는 方法으로 연결된 形狀을 찾아간다. 하나의 形狀要素에서 始作하여 終點에 도달하면 다음 形狀要素를 檢索하게 된다. 檢索하는 方法은 終點에서 적당한 크기의 창(Window)을 설정하여 선택세트(Selection Set : SSET)를 生成한다. 이 境遇에 SSET에는 2개의 形狀要素가 존재한다. 여기서 현재의 形狀要素를 제외하면 나머지 하나가 연결되는 形狀要素이다. 본 연구에서 分岐가 없는 形狀은 分岐가 있는 形狀의 認識方法에 포함된다.

(2) 分岐가 있는 形狀의 認識

그림 5-b는 終點에서 여러 갈래로 分岐되는 形狀을 나타내고 있다. 5-c와 같이 分岐點에 도착하면 다음 연결이 가능한 形狀要素의 후보는 1, 2, 3, 4 네개가 존재한다. 이때 한개의 形狀要素만 선택되므로 5-d에서 θ_1, θ_2가 L1을 基準으로 최소가 되는 形狀要素를 선택한다. 나머지 形狀要素는 배열상에 形狀要素의 이름과 分岐點을 저장하고 다음 形狀要素를 檢索한다. 分岐의 또 다른 형태로는 5-e와 같은 형태를 생각할 수 있는데 始作點에서 分岐點에 도달하기 전에 이미 남겨진 形狀要素가 존재하는 형태이다. 이때에도 始作點과 分岐點을 線分으로 하고 分岐點을 中心으로 시계방향으로 回轉시킬 때 만나는 순서대로 배열번호를 부여한다. 5-f와 같이 形狀要素가 교차되게 지나가는 境遇에는 認識할 필요가 없으므로 認識對象에서 제외한다.

(3) 나머지 情報의 認識

치수 및 문자 등의 認識은 關聯이 되는 形狀要素의 보조情報로 취급하여 처리한다. 표 3은 텍스트의 情報를 나타낸 것이며 Code 1번에 들어있는 텍스트의 내용을 認識하여 形狀情報를 추출해 낸다.

Table 3. The information of TEXT from AutoCAD Database.

code	attribute	variable type	remark
-1	〈Entity name : 6000053C〉	Name	entity name
0	TEXT	String	Entity type
1	Thanks for all the fish !	String	Content
50	0.523599	Real	Rotation angle (radians)
40	1.0	Real	Width factor
51	0.0	Real	Obliquing angle
7	STANDARD	String	Text style

3.2.2 形狀要素의 분석 및 形狀群(cluster)으로 處理

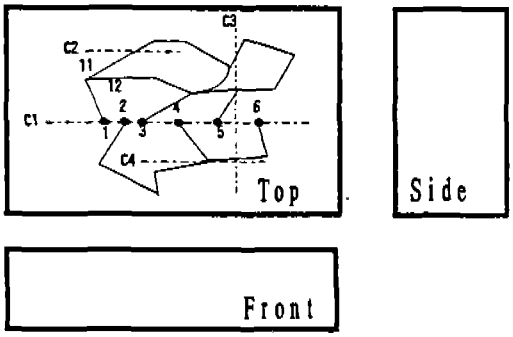


Fig. 6 Sample Drawing of Views

하나의 形狀要素가 다른 形狀要素에 연결되어 있으면 形狀群으로 취급한다. 또한 다른 形狀要素에 연결되어 있지 않으면 연결될 수 있는 최단거리에 있는 것을 찾아서 연결시킨다. 이때 경계를 구분할 수 있는 다른 投影圖들은 각각의 形狀群으로 처리한다. 그림 6에서 Top, Front, Side view는 별도로 管理된다. 形狀을 그룹화하기 위한 알고리즘을 다음과 같이 집합과 C언어의 문법의 조합으로 표현하였다.

E : 形狀要素

G_v : 投影圖(view)의 集合

V_s : 하나의 投影圖 혹은 斷面圖에 소속된 形狀要素의 集合

V_s-rem : V_s에서 처리되고 남은 形狀要素의 集合

E_s : 연결된 形狀要素의 集合(1-5개)

C_s : 中心선 上에서 출발하는 形狀要素의 集合

$C_s(j)$: C_s의 배열번호

$E_s(j)$: E_s에 최대값이 채워지면 부여되는 形狀群의 번호

CURRENT-s-p(i), CURRENT-e-p(i) :
현재 진행중인 形狀要素의 시작 및 終點

NEXT-s-p(i), NEXT-e-p(i) :
다음에 연결될 形狀要素의 시작 및 終點

C-node : 中心선의 집합

C-node(i) : 中心선의 배열번호

Ctrl-num : 形狀要素의 최적 조정 개수

E-node : 中心선을 제외한 形狀要素

E-nm-pt(k) :
分岐가 있는 形狀에서 남겨진 E-node의 이름과 시작點을 중간버퍼에 기억시킬 때 부여되는 形狀情報로써 데이타 구조는 다음과 같다.

여기서 i, j, k는 정수 :

E-nm-pt(k) ⇨Struct (int k, ads-name Entity-name, ads-point point)

LINK-E-node : 현재의 E-node에 연결되는 새로운 E-node :

1. G_v에서 V_s를 구분하여 영역을 분리한다. 그림 6에서 Top, Front, Side로 분리시키는 공정에 해당한다. ($V_s \in G_v$)

2. 모든 中心선을 檢索하여 C-node에 저장한다. 그림 6에서는 C1-C4가 對象이 된다.

3. C-node에 檢索하여 C-node(i)별로 $C_s(j)$를 구조체에 저장한다. (그림 6의 경우에는 C-node(0) (=C1에 해당) $C_s(j)$는 1-6까지 E를 의미)

4. $C_s(j)$를 檢索하여 C-node를 基準한 대칭形狀을 소거한다. (그림 4의 境遇에는 상반부만 남기고 하반부는 소거한다. 또한 상반부에서도 체결볼트의 경우는 상반부만 남기고 하반부를 소거)

```
for(j=0; Cs[j]==∅;j++) {
if (sign = symmetry_ent(Cs[j], ent1) == 1)
    erase(ent1);
```

5. C-node중에서 基準이 되는 것을 MAIN(C-node)라고 하면 이것은 식(1)과 같이 정의된다. MAIN(C-node)에서부터 시작되는 E-node 中에서 E-node가 많이 분포하는 방향에서 비교적 적게 분포하는 방향으로 $C_s(j)$를 부여한다.

$$MAIN(C\_node) = \{MAX[length[C\_node[i]]], \quad i=1,n\} \qquad \ldots(1)$$

6. $C_s(j)$의 j가 적은 값부터 먼저 檢索을 한다. E-node의 새로운 버퍼(buffer)에 최대값이 채워지면 $E_s(i)$에 저장하고 끝난 點을 시작點으로 하여 새로운 檢索을 반복한다. 만일 E_s의 최대값이 채워지기 전에 해당 $C_s(j)$의 작업이 종료되면 채워진 만큼의 갯수로써 $E_s(i)$를 결정하고 새로운 $C_s(j+1)$로 옮겨서 같은 순서로 檢索한다. 해당 알고리즘을 요약하여 표현하면 다음과 같다. (그림 9의 形狀분류와 같은 공정)

```
for(j=0; Cs[j]==∅;j++) {
for(i=0; LINK_E_node==∅;i++) {
if(LINK_E_node==∅ ⋂ i==Ctrl_num)
    WRITE(Es[i]);
```

else if(LINK_E_node==∅∩i<Ctrl_num)
WRITE(Es[i]); }}

7. 만일 $E_s(i)$에서 E-nm-pt(k)가 존재하지 않으면 $E_s(i)$를 V_s에서 소거한다. 존재하면 E-nm-pt(k)에 등록한 후 소거한다. V_s를 식으로 표현하면 식(2)와 같다.

$$V_s=\{V_s-E_s\mid E\_nm\_pt[k]\notin E_s[i]\,?\,(V_s-E_s) : E\_nm\_pt[k+1]\cap(V_s-E_s)\} \quad \ldots(2)$$

8. 만일 중간에 E-node의 연결이 끊어져 있으면 종點에서 사각형의 창(window)을 설정하여 통과하지 않은 E-node 중에서 선택하여 계속하고, 끊어진 부분에 대해서는 Null로 처리한 후 연결된 것으로 가정한다. 이 條件을 표현하면 다음과 같다.

IF{LINK_E_node $\notin V_s \cap V_s \neq \varnothing$}
THEN{NEXT_s_p[i++]=∃Pt[MIN(distance
(CURRENT_e_p[i], rem(∀Pt)))]}

9. k를 증가시켜 단계 6-9을 계속한다. k가 마지막에 도달하면 E-nm-pt(k)의 데이타를 이용하여 단계 6-9을 계속한다.

10. $V_s=\phi$이면 中立파일 生成작업을 종료한다.

3.2.3 그룹화된 形狀의 形狀生成 函數와의 連結

形狀의 구성要素를 조사하여 보면 直線과 圓弧의 연결로 구분될 수 있으며 直線과 圓弧가 조합된 形狀群의 형태를 그림 7에 나타내었고 그룹코드는 L(ine), A(rc), N(ull)의 조합으로 표기된다. 여기서 N(ull)은 중간에 形狀要素가 생략된 境遇를 의미하며, Fillet이나 Chamfer의 개념은 認識의 용이함을 위해서 圓弧나 直線의 개념으로 통합한다. LIN-ARC()函數는 그림 8-b와 같이 연결되는 形狀要素가 直線이나 圓弧이며, 形狀要素의 집합인 E_s의 시작點 S.P에서 끝點 E.P사이

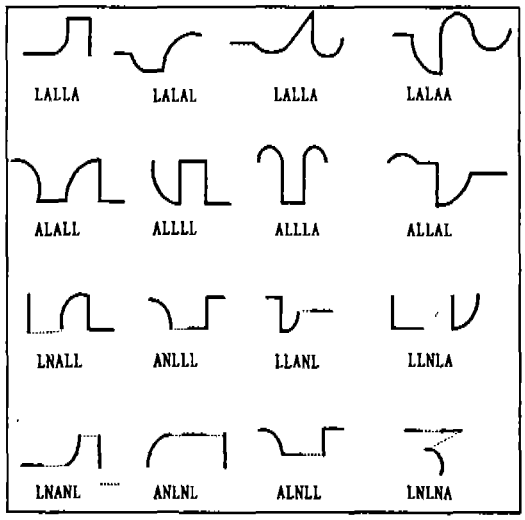
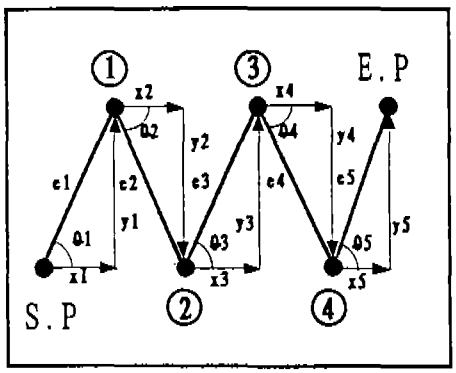


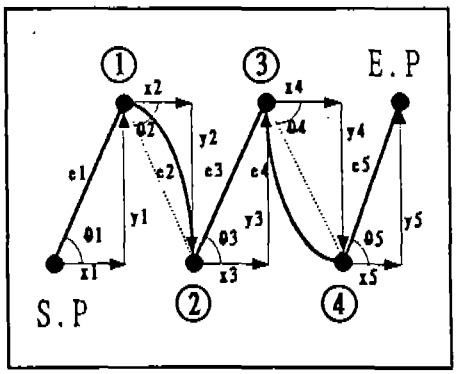
Fig. 7 Example of feature group by line and arc combination

Table 4. Illustration of function LIN-ARC()

Arg NO.	Arg name	Variable Type	Remark
1	E-code	String	그룹화 分類코드
2	x	Real	基準點의 x 좌표값
3	y	Real	基準點의 y 좌표값
4	lay	Character	線의 타입을 구분하기 위한 層(Layer)이름
5	ang	Real	中心線의 基準線에서의 각도(Degree)
6	ulf	Character	中心線을 基準으로 形狀의 生成위치
7	pd(0)-pd(5)	Real	각 直線을 연결하는 點의 直徑(X-Y평面상의 y값)
13	pl(0)-pl(5)	Real	각 直線을 연결하는 點 사이의 간격(x값)
18	r(0)-r(4)	Real	形狀要素가 Arc일 때 半徑(0 0R 1의 값) (None or Line →0)
23	C-code	String	形狀要素가 Arc일 때 방향 (C : 시계 방향, K : 반시계 방향) (None of Line →N)
28	ecp	Character	中心線이 존재여부



(8-a)



(8-b)

Fig. 8 Attribute of Feature Group by Line and Arc Combination

에 形狀要素가 최대 5개까지 존재하고 각각의 形狀要素를 연결하는 구조는 그림 8과 같이 e1과 e2와 e3과 e4 그리고 e4와 e5를 연결한다. 표 4는 그림 8-b를 모형으로 LIN-ARC()函數의 매개變數로 설명한 것이다. LIN-ARC()函數의 구조는 다음과 같다.

```
(LIN_ARC E_code x y ang ulf pd[0] pd[1]
pd[2] pd[3] pd[4] pd[5] p1[0] p1[1] p1[2]
p1[3] p1[4] r[0] r[1] r[2] r[3] r[4]
C_code ecp)
```

여기서 E-code는 中立파일의 形狀要素의 집합인 "$$$$$"와 같으며 x1, y1은 中立파일의 x1, y1과 같고, layer는 中立파일의 대표 layer와 같다. ang는 p1과 p2가 이루는 값인데, ADS에서 두點이 이루는 각을 구하는 函數인 ads-angle(p1, p2)를 이용한다. ulf는 p1, p2가 이루는 직선을 基準으로 상반부, 하반부, 상하 모두에 위치시킬 것인가를 결정한다. p1(i), pd(j)는 中立파일의 p1-p6에 해당된다. p1과 p2를 @x를 c : d, @y를 a:b로 하는 비율을 계산하여 p1, p2를 拘束시킨다. 만일 4-c의 영역에 p3, p4과 같이 KPS가 없는 境遇에는 p3, p4를 @x, @y의 비로 정의해서 拘束시킨다.

예를들어 그림 4-a에서 形狀要素 E3은 4-b에서 〈x1, y1〉와 〈x3, y3〉이 이루는 영역안에 존재하므로 그림 4-c에서 p3, p4의 境遇에 해당한다. 만일 ARC가 존재할 때 P5, P6사이의 거리가 달라지면 PD기법에 의해서 ARC가 연속적으로 변하는 곡선으로 처리되어야 하지만 境遇에 따라서는 너무 많은 ARC의 집합으로 된다. 이 境遇에 다음 공정에서(예 가공情報 추출) 認識이 난해하므로 반경을 고려한 하나의 ARC로 처리하는 것을 원칙으로 한다. 대부분의 形狀要素는 KPS 상에 끝점이 존재하므로 직접영향(그림 4-a의 E1, E2 등)을 받으며 간접영향(E1, E2 등)을 받는 境遇도 있다. 만일 그림 4-a에서 D1과 L만이 존재한다면 形狀要素들은 전부 간접 영향을 받으며 PD개념의 적당한 예가 될 것이다. 形狀그룹화가 끝나면 개별 形狀要素는 치수變數의 函數로 표현된다.

LIN-ARC函數의 매개變數 중 r(i)는 形狀要素가 ARC일 때 각각의 반경값을 나타내고 C-code는 ARC의 시계, 반시계를 구한다. ecp는 中心선 삽입 여부를 결정하는 것인데 항상 "y"값을 기본으로 갖는다. 이상과 같이 표 4의 매개變數들이 결정되면 基準點(x, y)을 基準으로 해당形狀에 대해 LIN-ARC 函數가 生成된다.

3.2.4 認識된 데이타의 中立파일로의 變換

AutoCAD Drawing Editor(이하 ADE)상에서 認識된 形狀情報들을 形狀처리 函數인 LIN-ARC()에 適用하려면 認識된 결과의 D/B와 K/B를 직접 形狀函數에 適用하는 方法보다 중간과정을 管理하는 中立파일을 利用하는 것이 용이하다. 中立화일을 利用하면 ADD환경이 아닌 다른 환경에서도 프로그램을 自動으로 生成할 수 있는 장點도 있다. 그림 4의 變數들 중 E-node에서 시작點과 分岐點의 x값은 L1, L2의 函數관계이고, y값은 D1, D2, D3, D4, L3의 函數관계이다. 즉 x와 y는 다음식과 같이 표현된다.

$$x[i] = f(L1, L2) \qquad \ldots\ldots(3)$$

y[i] = g(D1, D2, D3, D4, L3) .....(4)

식(3), (4)에서 임의의 點 P(i)는 P(i) = (x(i), y(i), z(i))로 표현할 수 있으므로 모든 E-node는 (*, /, +, -)의 사칙으로 표현되는 매개變數 문자열로 구성된다. 표 5는 中立파일의 구조를 나타낸 것이다.

3.2.5 生成 프로그램의 모듈별 分類

形狀에 대한 하나의 完全한 프로그램을 분석하여 보면 아래와 같이 分類된다.

- ○ 관련函數 Loading부
  形狀과 關聯된 필요한 函數를 Loading
- ○ 프로그램 설명부
  날짜 및 작성자, 技能에 대한 설명
- ○ 函數명 設定
  形狀과 關聯된 函數명 設定
- ○ 환경變數 設定
  Auto CAD의 시스템變數 設定
- ○ Slide file 生成
  Slide file show 여부
- ○ 原點지점
  Drawing : 圖形始作의 基準이 되는 點
  Block : 圖形전체의 基準이 되는 點
- ○ D/B file 읽기 및 割當
  치수變數에 데이타 割當
- ○ 形狀生成
  中立파일 利用
- ○ Block 指定 및 이름 設定
  形狀特徵에 關聯
- ○ Block 挿入
  원하는 位置

4. 適用 사례

구축된 시스템을 利用하여 그림 4의 예제圖面을 수행하였고 사출금형의 주요부품인 로케이트링과 스프루부시

Table 5. The structure of neutral file

@ ! ! !	//形狀群 번호(맨앞에 항상 @를 붙인다.)
$$$$$	//形狀要素 Structure (L : Line, A : Arc, N : None)
###, ###, ###, ###	//基準線(X1, Y2, X2, Y2)
###	//대표 layer
1. ###, ###	//P1 ###, ###
2. ###, ###	//P2 X, Y
3. ###, ###	//P3 (X, Y는 基準線을 X축으로 하고)
4. ###, ###	//P4 (基準線의 P1(X1, Y1)을 原點으로 한다.)
5. ###, ###	//P6
#. ###, !	//# : Ent No., ### : 반경, ! : 방향(0 : CW, 1 : CCW)
.....	//DATA는 엔티티 구조의 R갯수만큼 한다.
!, LP	//대칭여부(0 : No, 1 : Yes), LP : 線 or 點 대칭
{ ###, ###	//대칭시에만 P1(X1, Y1), P2(X2, Y2)가 들어간다.
###, ###}	//대칭이 아닐 시에는 없음.
	Line Type이 바뀔 시에는 아래의 명령어를 使用한다.
@ ! ! !	//다음 形狀코드 始作
$$$$$	
.	
.	
.	
%	//File이 끝났음을 표시

(Locating Ring and Sprue Bush)의 形狀을 對象으로 하였다. 分岐가 없는 形狀은 비교적 認識이 용이한 반면, 分岐가 있는 形狀은 남겨진 形狀要素의 管理 및 처리에 실질적으로 많은 시간이 소요되며, 境遇에 따라서 매우 애매한 部分은 認識의 혼란을 초래할 수도 있다. 대부분의 形狀들은 分岐를 가지고 있으며, 신발의 윤곽을 設計한 形狀은 分岐가 없는 形狀의 예라고 볼 수 있다. 데이타 구조는 幾何學的인 Constraints와 위상학적 Parameters를 만족하면서 새로운 형태로 정의한다.

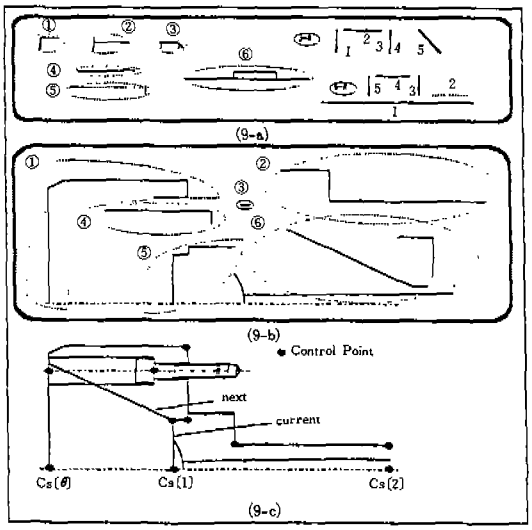


Fig. 9 Detailed Entity Groups for function LINE-ARC( )

그림 9는 그림 9-a 形狀을 Es집합 단위로 그룹화한 모형을 나타내고 있다. 그림 9-a는 체결볼트의 境遇이고, 그림 9-b는 로케이트링과 스프로부시의 境遇를 나타낸다. 그림 10은 그림 9에서 하나의 Es을 나타낸 것인데 ①-⑤까지의 5개가 이루는 외곽形狀을 나타내는 中立파일은 Table 6와 같이 표현된다. 그림 9-a와 9-b에서 5개 이하로 그룹화된 Es들은 각각 그림 10과 같은 方法으로 中立파일에 등록되고 등록된 Es의 情報는 LISP의 원시파일을 生成에 관여한다. 도형生成시 순서는 중요하지 않으므로 파일의 變換은 표 6의 形狀군의 번호에 따라 순서대로 變換된다. 그림 11은 C++을 이용한 시스템 메뉴를 나타낸다. 自動生成에 필요한 입력사항은 中立파일명, 變數처리한 形狀관련 D/B

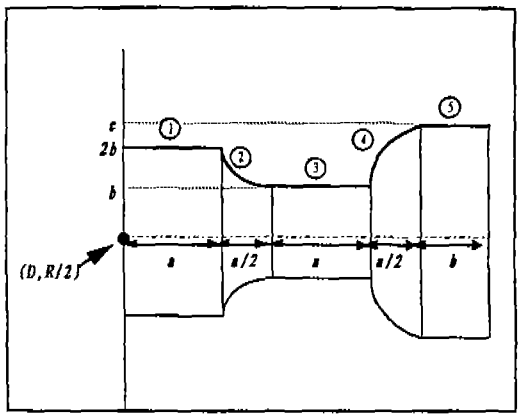


Fig. 10 Example drawing of Es Set

Table 6. Input D/B of Locating Ring and Sprue Bush

L1	L2	L3	D1	D2	D3	D4	ANG.
50	35	80	100	90	15	130	2
50	35	80	100	90	20	130	2
55	40	85	100	90	25	130	2
55	40	85	120	100	15	150	2
60	50	90	120	100	20	150	2
60	50	90	120	100	25	150	2

파일명, 生成될 LISP 원시파일명, LISP 函數명, 形狀을 BLOCK로 처리하기 위한 BLOCK명, 形狀을 Slide로 보여줄 것인가의 여부, 도형전체의 基準點과 BLOCK의 基準點 정한다. 메뉴 우측에서는 CAD 시스템의 設計환경을 설정할 수 있도록 구성되어 있다. 입

Table 7. Example of neutral file of Locating Ring and Sprue Bush

```
@1
LLLLN
S
LRx-10, LRy-10+3-LR12-8, LRy
0, 0
0, LRd1-2
3, LRd1
LR12-8, LRd1
0, 2*LR13+6, 6
0, 0
I, L
LRx-10, LRy, LRx-10+3-LR12-8, LRy
```

력이 완료되면 F10으로 수행한다. 표 7은 그림 4의 로케이트링과 스프루부시의 볼트를 제외한 形狀의 中立파일을 나타내는데 볼트는 다른 부품으로 이미 函數화 되어 있어서 독립적으로 사용된다. 그림 12는 BGS라는 中立파일과 形狀관련 D/B 파일을 이용하여 自動으로 生成된 設計모듈을 나타내고 있다.

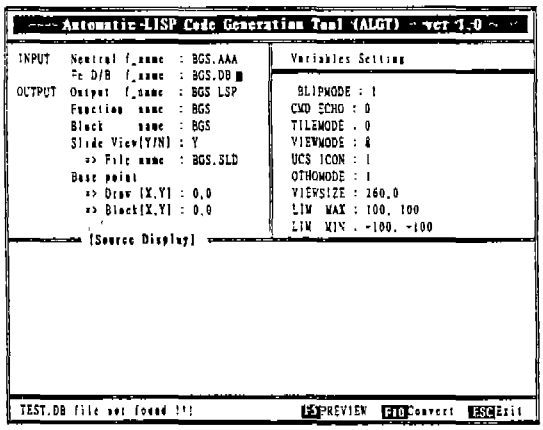


Fig. 11 System menu to generate LISP-language from neutral file

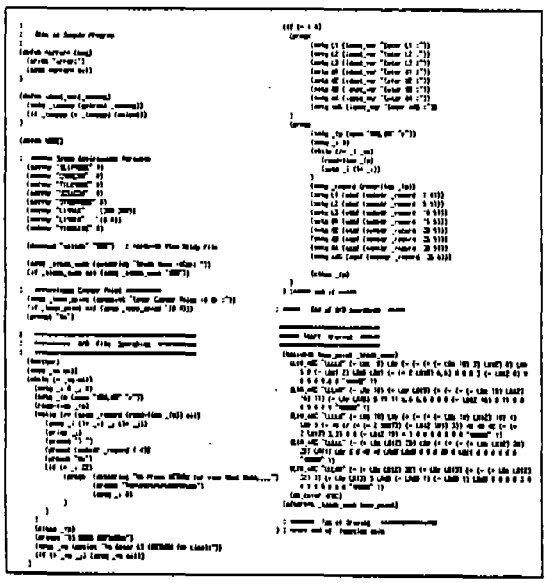


Fig. 12 Automatic Generated LISP code from Neutral File

5. 결론 및 고찰

프로그램 自動生成시스템의 特徵을 몇가지 나열하고 향후 발전방향에 대해 언급한다.

(1) 旣存의 方式은 프로그래머가 직접 形狀을 認識하고 形狀要素를 그룹화 하였으며 形狀關聯 위상학적 관계를 전문가의 認識으로 규명하여 프로그래밍 하였으나 본 硏究에서는 이 工程을 自動化 시켰다.

(2) 自動設計시스템의 개발에 소요되는 시간이 전문가의 認識에 의한 方法보다 개발된 시스템을 적용하면 形狀에 따라 5-10배의 절감효과를 가져온다.

(3) 設計된 形狀의 認識은 形狀分類 및 그룹화를 위해서 필요하며 모든 形狀要素에 대해서 중복을 허용하지 않고 효과적인 形狀궤적을 찾는 일은 대단히 복잡한 工程이다. 變換파일을 利用하지 않고 ADE에서 시각적인 認識方法을 利用함으로써 사람의 認識方法에 매우 근사하여 認識에 關聯된 규칙을 變換파일 利用方法보다 더욱 용이하게 K/B化 했다.

(4) 이미 設計된 形狀을 중요한 部分의 形狀을 치수와 關聯하여 變數化하는 PD기법을 適用하여 치수變數의 조작으로 設計를 변경시킬 수 있다. 또한 기존의 관심이 있는 部分을 變數化함으로써 類似形狀의 設計에 효율적으로 使用할 수 있다.

(5) 標準化된 形狀뿐만 아니라 標準化되지 않은 形狀의 표준화 工程에 유용하게 適用될 수 있으며, 2차원으로 표현되는 大部分의 形狀에 공통적으로 適用되므로 활용범위가 매우 넓다.

차기에는 지금까지의 PC수준에서 구현하는 축적된 기술을 확대 適用하여 設計된 圖面을 CAM의 실질的인 情報로 연결하여 CAD/CAM의 연결시스템을 구축할 것이다.

후 기

본 연구는 1992년도 한국학술진흥재단 연구비 및 ㈜금성사의 데이타 지원으로 수행되었으며, 관계자 여러분께 깊은 감사를 드립니다.

참고문헌

1. Chang, T. C., "Expert Planning for Manu facturing", Addison-Wesley Publishing

Company, 1990.
2. Joshi, S., "CAD Interface for Automated Process Planning", Ph. D. Thesis, Purde University, West Lafayette, Ind. 1987.
3. Jakubowski, R., "Syntatic Characterization of Machined Part Shape", Cybernatics, 13, 1~24, 1982.
4. Staley, S. M., et. al., "Using Syntatic Pattern Recognition to Extract Feature information from Solid Geometric Model Database", Computers in Mechanical Engineering, 2, 61~66, 1983.
5. Srinivasan, R., et. al., "Extraction of Manufacturing Details from Geometric Models", Computer and Industrial Engineering, 9, 125~133, 1985.
6. Li, R. K., "A Part-feature Recognition System for Rotational Parts", Int. J. Prod. Res., Vol. 26, No. 9, 1451~1475, 1988.
7. Wang, H. P. and Wysk, R. A., "AIMSI : a preclude to a new generation of integrated CAD/CAM systems", Int. J. Prod. Res., Vol. 26, No. 1, 119~131, 1987.
8. 이경휘, 정무영 "回轉形狀 部品에서의 加工形狀 特徵의 自動認識에 관한 硏究", 한국自動제어학회, 19~21, 1992.
9. Roller, D., "An approach to computer-aided parametric design", Computer Aided Design, Vol, 23, No. 5, 385~391, 1991.
10. Aldefeld, D., "Variation of geometrices based on a geometric-reasoning method", Computer-Aided Design, Vol. 20, No. 3, 117~126, 1988.
11. Verroust, A., Schonek, F. and Roller, D., "Rule-oriented method for parameterized computer-aided design", Computer-Aided Design, Vol. 24, No. 10, 531~540, 1992.
12. T. McMahon, C. A., Lehane, K., Sims Williams, J. H., and Webber, G., "Observations on the application and development of parametric programming techniques", Computer-Aided Design, Vol. 24, No. 10, 541~546, 1992.
13. M. R. Henderson and S. Musti, "Automated Group Technology Part Coding From a Three-Dimension CAD Database", Transactions of the ASME : J. of Engineering for Industry Volume 110, No. 3, 1988.
14. R. Billo, R. Rucker, D. Shunk, "Intergration of a Group Technology Classification and Coding System with an Engineering Database : J. of Manufacturing System Volume 6 No. 1, 1987.
15. Jami, J. S. and Anant, S. B., "Group Technology Classification from Feature-Based Geometric Models", Manufacturing Review, Vol. 2, No. 3, 204~213, 1989.
16. Andrew Kusiak, "Knowledge-Based Group Technology", Technical Report, University of lowa, 1988.
17. Randy George, "Developing Intelligent Drawing", Technical Report, CADENCE, May, 65~68, 1988.