A Hierarchical CAPP System Architecture for Die Cavity Machining

K. KO¹, B.H. KIM², D.H. KIM³ and B.K. CHOI¹

¹ Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Seoul, KOREA

² Korea Institute of Industrial Technology, Seoul, KOREA

³ Mold Shop, Wooyoung Co. Ltd., KOREA

Abstract: This paper proposes a hierarchical CAPP (*computer automated process planning*) system architecture for die cavity machining. The CAPP system structure has been developed on the basis of the current practices of die cavity machining and is composed of five modules – *Form EDM, Clean-up, Finish, Semi-finish, and Roughing* planning. The individual CAPP modules are connected in a sequential manner. They are identical in structure and each module consists of four sub-functions for *feature extraction, process planning, machined surface update, and NC-code generation*. Input to the CAPP system are 1) a CAD model of the die surface geometry, 2) the geometry of the raw stock, and 3) a set of available cutting tools with machinability data. Output from the CAPP system is a set of NC-codes. A process plan example, which is generated by a prototype CAPP system developed on the proposed structure, shows that the proposed CAPP architecture is practical and valid in the die cavity machining.

Key words: CAPP, machining stage, machining features, unit machining operation (UMO), STOPP, Form-EDM, Clean-up, Finish, Semi-finish, Roughing

1. INTRODUCTION

Injection molding dies for home-electronics products as well as stamping dies for automobile panels have free-formed surface geometry with complex shapes. It usually takes tens of, sometimes hundreds of, hours to machine a die cavity from a solid block. Planning for die cavity machining needs a high

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35637-2_11](https://doi.org/10.1007/978-0-387-35637-2_11)

R. Soenen et al. (eds.), *Feature Based Product Life-Cycle Modelling*

© IFIP International Federation for Information Processing 2003

level of skill and experience. Often, the time required to prepare NC tool paths for the die cavity machining takes as long as the machining time itself. There is an increasing demand for a CAPP system for the die cavity machining because of the increased pressure for shorter lead-times and higher precision, coupled with the shortage of skilled workers.

CAPP has been a hot research topic (Pratt 1984 and Kanumury 1991), and more than 150 CAPP systems have reportedly been developed (Alting 1989 and Shin 1998). However, very little is published about CAPP for the die cavity machining (Yoo 1992). (Lee *et al* 1992) investigated a rough cutting strategy for free-formed cavity machining, and (Yoo and Choi 1992) proposed a recursive CAPP logic for the machining of purely free-formed die cavities. However, neither of the papers considered the *machining features* inherent in the die cavity machining.

Presented in the paper are a hierarchical CAPP system architecture for the die cavity machining and a methodology for developing such a recursive CAPP logic in (Yoo and Choi 1992). It is based on the STOPP (sequential and tool-oriented process planning) approach proposed by (Choi *et al* 1985) and is structured according to the current practices of the die cavity machining in Korea.

To develop a hierarchical CAPP system, we have to analyse detailed machining practices and then identify rules for applying specific machining operations. In this process, the concept of machining feature plays a key role. The second section reviews the recursive CAPP logic, and the third section introduces the basic elements of the CAPP system for the die cavity machining. Based on the industrial practices of the die cavity machining, this study proposes a CAPP system structure. A process plan example, which is generated by a prototype CAPP system developed on the *Z-map* based CAM software, shows that the proposed CAPP architecture is practical and valid in the die cavity machining.

2. RECURSIVE CAPP LOGIC FOR DIE CAVITY MACHINING

A *generative* CAPP for machining involves an automatic generation of a sequence of machining operations. A purely generative CAPP employs a *backward recursive approach* (Choi 1985). In the generative CAPP, a *recursive process* automatically identifies a sequence of machining operations while recursively updating the *states* of the part. The geometry and tolerance of a part specify its state. The initial raw stock for die cavity machining is either a solid stock called *moldbase* or a *preform* that is made by rough machining, forging, or casting.

In the generative CAPP, a “planning operation” is regarded as a “metal addition” operation that is the reverse of a machining (metal removal) operation. From tool *database* (DB), each planning operation selects the cutting tools that can successfully add metal to the current machined surface to obtain a feasible set of process plans. It then selects a “best tool” according to a performance measure. Finally, it updates the *current state of the part* (that is, the machined surface is offset and the tolerance relaxed) to complete a planning step. This individual planning step is repeated until the current state of the part becomes the *final state* of the part. This planning procedure is summarized as follows (Yoo 1992):

Recursive CAPP Algorithm:

0. *Input: Initial (current) state and tolerance, final state, tool DB.*

1. *Select all the tools that can “plan” the current state of the part.*

2. *Choose a tool (or a set of tools) that gives minimum cutting time.*

3. *Update the current state by offsetting the surface and relaxing the tolerance.*

4. *If the current state equals to the final state then stop, else go to 1.*

When planning for sculptured surface machining, the relationships among the following three items play an important role:

- *Machining allowance*, α : the amount of intentional under-cut thickness.
- *Machining tolerance*, τ : maximum deviation of the actual machined surface from the ideal machined surface to be obtained.
- *Machining depth*, δ : the difference between the machined surface and the preform surface.

The required machining depth δ at a (machining) stage is determined as a function of the tolerance τ . Similarly, τ is also defined as a function of α . These relations are expressed as

$$\tau = f(\alpha) \text{ or } \alpha \Rightarrow \tau$$

$$\delta = f(\tau) \text{ or } \tau \Rightarrow \delta$$

Obviously, the machining allowance at a stage becomes the machining depth at the next stage of machining. There usually are three stages of machining in the die cavity machining: *roughing*, *semi-finishing*, and *finishing*. This paper assumes the following technical relationships:

$$\tau_f = 0.05 \Rightarrow \delta_f = 0.3, \text{ for finishing;}$$

$$\alpha_s (\equiv \delta_f) = 0.3 \Rightarrow \tau_s = \pm 0.15 \Rightarrow \delta_s = 2.0, \text{ for semi-finishing;}$$

$$\alpha_r (\equiv \delta_s) = 2.0 \Rightarrow \tau_r = \pm 1.0 \Rightarrow \delta_r = \infty, \text{ for roughing;}$$

These data have no theoretical basis, but die makers use them as “technical standards”. We will also use these rules throughout the paper.

The die cavity machining mainly uses *ball endmill (BEM)* and *flat endmill (FEM)*. The following abbreviations will be used throughout the paper:

BEM20 ϕ : ball endmill of 20mm in diameter.

FEM20 ϕ : flat endmill of 20mm in diameter.

Drill50 ϕ : drill of 50mm in diameter.

Figure 1 shows a schematic description of the recursive CAPP logic. As depicted in Figure 1-a, the final geometric shape of the die cavity and its tolerance value ($\tau_f=0.05\text{mm}$) are specified as an *initial state* of the part. The first step *plans* a finishing operation:

- 1) Select a finishing tool (*BEM20 ϕ*)
- 2) Offset the part by δ_f ($=0.3\text{mm}$) and relax its tolerance to τ_s ($=0.15\text{mm}$).

The result is a *finishing preform* as depicted in Figure 1-b. The next step plans a semi-finishing operation:

- 1) Select a cutter (*BEM30 ϕ*)
- 2) Offset the part by δ_s and relax the tolerance to τ_r ($=1\text{mm}$).

Now we have a *semi-finishing preform* to be obtained by a roughing operation as depicted in Figure 1-c. The roughing (i.e., pocketing) operation is planned as:

- 1) Select a pocketing tool *FEM50 ϕ* .
- 2) Fill the die cavity up to the initial solid block ($\delta_r = \infty$).
- 3) Plan for a drilling operation.

Figure 1. Recursive CAPP Logic

3. BASIC ELEMENTS OF CAPP SYSTEM

The basic elements of CAPP system for the die cavity machining are machining stages (*MS*), machining features (*MF*), and unit machining operations (*UMO*).

3.1 Machining stages

Diverse definition of machining stages has been proposed in the literature (Fallbohmer 1995, Yamazaki 1995, and Yoo 1992), with the most popular scheme being the “three-stage” scheme consisting of roughing(*R*), semi-finishing (*S*), and finishing (*F*). However, this study proposes a “five-stage” decomposition scheme based on a close examination of the die cavity machining practices employed by die-makers. *Figure 2* shows the five machining stages: roughing stage, semi-finish stage, finish stage, clean-up stage, and form-EDM stage.

Figure 2. Die cavity machining stages

- *Roughing machining stage*: Removes metal from a raw stock as fast as possible (using heavy-duty cutters) to obtain a machined preform for the next machining stage (i.e., semi-finishing).
- *Semi-finish machining stage*: Obtains a “finishing” preform having a uniform thickness of the finishing allowance.
- *Finish machining stage*: Obtains the final shape of the die surface with the required surface-finishes. To have a shortened machining time, this stage usually uses relatively large cutters (ball endmills of 20-30mm diameter).

- Clean-up machining stage*: Using smaller ball endmills (and flat endmills), removes uncut volumes at the concave regions where the “finishing” ball endmills could not reach.
- Form-EDM machining stage*: Using special form-cutters or EDM-electrodes, cuts the sharp concave edges and the special depression that can not be produced with a ball endmill (The EDM-electrodes are usually machined by ball endmills).

3.2 Machining features

Most of the free-form features introduced in the literature (Park 2000, Shin 1998, and Storr 1995) are concerned with machining applications. This study assumes that the die surface is visible from above (i.e., no “negative draft” in the die surface) and that NC machines can perform all 3-axis machining operations. Based on a close observation of the current die cavity machining, this study takes into account five types of machining features: wall type machining features, floor type machining features, concave-strip type machining features, volume type machining features, and prismatic type machining features (Choi 1996).

1) Wall type machining features

- Wall*: a near vertical surface with positive draft angles (say, less than 30 degree) (*Figure 3-a*).
- Vertical wall*: a wall with a zero draft-angle (*Figure 3-b*).
- Cavity wall*: the entire wall of a die cavity forming a closed strip of walls (*Figure 3-e*)
- Core wall*: the entire wall of a core or a protrusion (*Figure 3-f*).

2) Floor type machining features

- Floor*: a near horizontal surface with an inclination angle (say, less than 30 degree) (*Figure 3-a, b, and c*)
- Compound floor*: a “compound” area in the die surface containing a number of floors and other features (*Figure 3-d*).

3) Concave-strip type machining features

- Vertical concave fillet*: a concave blending surface between two draft(or vertical) surfaces, or a concave edge(a round with a very small fillet radius or a sharp concave edge)(*Figure 3-a*).
- Sloped concave fillet*: a concave blending surface between a slant and a draft surface, or a sloped concave edge (*Figure 3-c*).
- Horizontal concave fillet*: a concave blending surface between floor and draft surfaces, concave edge, or a round groove (*Figure 3-b*).

4) Volume type machining features (*Figure 3-h*)

- Cavity volume*: a volume in the die cavity that is defined as a Boolean difference between a preform and a machined surface.

▪ *Shoulder volume*: an “open” volume of metal to be removed.

5) Prismatic type machining features

▪ *Step*: a stepped feature consisting of a vertical plane and a horizontal plane.

▪ *Slot*: a rectangular slot.

▪ And more types: there are many papers dealing with prismatic-type features (Han 1998, Lee 1998, and Xu 1998).

Figure 3. Die cavity machining features

3.3 Unit machining operations

A unit machining operation (UMO) is defined as a specific pattern of the die cavity machining operation performed using a single cutter (Choi 1998). Introduced in the study are a total of 17 UMOs that are widely employed in the die cavity machining. Based on the mapping relationship between the machining features and UMOs, this study groups the UMOs into the following five types: wall type UMOs, floor type UMOs, strip type UMOs, volume type UMOs, and special UMOs (*Figure 4* shows some UMOs usually used in the die cavity machining).

- 1) *Wall type UMOs*: z-constant contour-cut, BC(boundary curve)-parallel contour-cut, helical contour-cut.
- 2) *Floor type UMOs*: XY-parallel area-cut, BC-parallel area-cut, BC-normal area-cut, CPO(contour parallel offset) area-cut, and spiral area-cut.
- 3) *Strip type UMOs*: pencil-cut, strip-parallel fillet-cut, and strip-normal fillet-cut.
- 4) *Volume type UMOs*: drilling, pocketing, and shouldering.
- 5) *Special UMOs*: form-cutting (slotting and 2D-contouring) and EDM.

Figure 4. Some UMOs used in the die cavity machining.

4. CAPP SYSTEM ARCHITECTURE FOR DIE CAVITY MACHINING

The proposed CAPP system architecture is built around the machining practices. By reversing the sequence of the machining stages introduced in the previous section, this study constructs a five-stage CAPP logic as follows:

Five-Stage CAPP Logic

0. Input: Raw Stock; Die Cavity; Tool List;
1. Plan for the Form-EDM machining;
2. Plan for the Clean-up machining;
3. Plan for the Finish machining;
4. Plan for the Semi-finish machining;
5. Plan for the Roughing machining;
6. Output: NC-codes.

Figure 5 shows the overall structure of the proposed CAPP system, which is based on the five-stage CAPP logic. The individual CAPP modules are connected in a sequential manner and the planning operations are driven by available cutting tools. However, we characterize the proposed CAPP structure as “sequential” and “tool-oriented” (instead of “recursive” and “generative”). Specifically, it is an adaptation of the *STOPP* (Sequential and Tool-Oriented Process Planning) approach proposed by (Choi *et al* 1985).

Figure 5. Overall Structure of the CAPP System

Figure 6 shows the internal structure of each CAPP module that includes the following objects:

- $MS[j]$: machined surface at the planning stage j .
- $TL[j]$: tool-list at the planning stage j .
- $MR[j]$: machining rules to be used at the planning stage j .
- $FL[j]$: feature-list for the features extracted at the planning stage j .
- $PP[j]$: process plans obtained at the planning stage j .
- $NC[j]$: NC-codes generated at the planning stage j .

Figure 6. Structure of the local CAPP Module

Each CAPP module has its own set of built-in machining rules, $MR[j]$ (Machining rules indicate how to cut for machining features extracted from the postform at each machining stage), and its input and output are as follows:

- **Input:** $MS[j]$, $TL[j]$, $TL[j+1]$.
- **Output:** $NC[j]$, $MS[j+1]$.

Note that

$$MS[1] \equiv \text{Die Cavity}; \quad (MS[6] \equiv \text{Raw Stock})$$

Further, to make all the modules identical in structure, this study uses the following convention:

$$TL[6] \equiv \text{Raw Stock}.$$

Internally, the planning function of a CAPP module j consists of the following four sub-functions:

- 1) Extract a set of features $FL[j]$ from $MS[j]$ using $MR[j]$, $TL[j]$ and $TL[j+1]$.
- 2) Plan for machining processes $PP[j]$ from $FL[j]$ using $MR[j]$ and $TL[j]$.
- 3) Update the machined surface from $MS[j]$ to $MS[j+1]$ using $PP[j]$.
- 4) Generate NC-codes $NC[j]$ from $MS[j]$ using $PP[j]$, $TL[j]$, and $MR[j]$.

Thus, this study formalizes the “Five-stage CAPP logic” in terms of the following “sequential and tool-oriented process planning” (STOPP) algorithm:

STOPP Algorithm:

0. Input : $MS[1] = \text{Die Cavity}; \quad TL[6] = \text{Raw-Stock};$
 $\{ TL[j], j = 1, \dots, 5 \} = \text{Tool-Lists};$
1. For $j = 1$ to 5 do {
 $FL[j] \leftarrow \text{extract_feature}(MS[j] \mid MR[j], TL[j], TL[j+1]);$
 $PP[j] \leftarrow \text{plan_processes}(FL[j] \mid MR[j], TL[j]);$
 $MS[j+1] \leftarrow \text{update_MS}(MS[j] \mid PP[j]);$
 $NC[j] \leftarrow \text{generate_NC}(MS[j] \mid PP[j], TL[j], MR[j]);$
 $\}$
2. Output $\{ NC[j], j = 1, \dots, 5 \};$

The machined surface $MS[j]$ denotes the geometry of the die cavity at stage j (The technological data such as uncut allowance, tolerance, and cutting depth are treated as machining rules $MR[j]$). The feature list $FL[j]$ is a set of machining features F_i , where F_i is defined as follows:

$$F_i = \text{feature_name}[\text{parameter} \mid \text{update}];$$

where, $\text{parameter} = \{ \alpha \text{ (draft-angle)}, \rho \text{ (radius)}, \delta \text{ (depth)}, \omega \text{ (width)}, \text{etc.} \};$

$\text{update} = \{ \rho \text{ (preform fillet radius)}, o \text{ (offset distance)}, \text{etc.} \}$

The “update” field contains methods of updating machining features (or the postform-surface). If an argument is omitted, its value is assumed to be zero. Actually, each planning stage takes into account only a limited number of types of feature elements like the following (finishing stage) example:

$$FL[3] = \{ \text{concave_fillet}[\mid o = \delta_f, \rho_p = R_s], \text{wall}[\mid o = \delta_f], \text{floor}[\mid o = \delta_f] \}$$

where, Subscripts f, s denote finish, semi-finish, respectively;

R : minimum cutter radius;
 δ_f : machining depths at finish stage.

For the sake of simplicity, this study excludes *convex features* (Specifically, the *convex-strip features* are treated as a part of *floor* or *wall*).

A process-plan $PP[j]$ contains a number of machining-plans $\{P_i\}$. A machining-plan consists of a machining-features(MF) and a UMO, and is expressed as follows:

$$P_i \equiv (MF_i, UMO_i).$$

Finally, the update operation is expressed as

$MS[j+1] \Leftarrow \text{update_MS}(MS[j] \mid \text{type}(MF), v); \quad v: \text{real number.}$

where, $\text{type} = (\text{offset}, \text{fillet}, \text{fill})$.

The real number v denotes the offset distance, fillet radius, or the amount of fill. If the cavity is filled up to the rim, it is expressed as

$$MS[j+1] \Leftarrow \text{update_MS}(MS[j] \mid \text{fill}(F1), \infty);$$

5. ILLUSTRATIVE EXAMPLE

The STOPP algorithm for the die cavity machining requires five CAPP modules (*Figure 5*), and each module consists of four sub-functions (*Figure 6*). We define the mapping relationships between machining features and UMOs under the proposed CAPP structure, and develop a prototype CAPP system on the *Z-map* based CAM software called *Soft-Master®*.

The prototype CAPP system generates a sequence of machining operations for an example of the die cavity machining. The input data of the example is a CAD model of a (17 inches) monitor front-mask consisting of 83 trimmed surfaces and a raw stock having the size of 545mm length, 504mm width, and 100mm height.

▪ $MS[1]$ = Die cavity (*Figure 7-a* shows the CAD and the Z-map model of the example).

▪ $TL[6]$ = Raw stock (moldbase, *Figure 7-b*).

▪ Tool lists:

$$TL[1] = \{ FEM20\phi \}$$

$$TL[2] = \{ BEM4\phi \}, BEM8\phi, BEM12\phi, BEM16\phi \}$$

$$TL[3] = \{ BEM20\phi \}$$

$$TL[4] = \{ BEM30\phi \}$$

$$TL[5] = \{ FEM30\phi, \text{Drill}25\phi, \text{Drill}50\phi \}$$

Figure 7. Input data of an example process planning

Stage 1: Form-EDM Plan

0) Machining rule, $MR[1] = \{R1\}$: machining rules for *Form-EDM* planning.

▪ $R1$ = Form features (step) are machined with FEM20 ϕ .

1) Feature extraction, $FL[1] = \{F1\} \Leftarrow extract_features (MS[1]|MR[1], TL[1], TL[2])$: Form features with fillet radius ρ less than 2mm (BEM4 ϕ) are extracted.

▪ $F1 = step[\mid \rho_p = 2.0]$.

2) Process plan, $PP[1] = \{P1\} \Leftarrow plan_process (FL[1]|MR[1], TL[1])$

▪ $P1 \equiv (F1, UMO1) = (step, 2D-contouring[FEM20\phi])$.

3) Machined surface, $MS[2] \Leftarrow update_MS (MS[1]|PP[1])$

▪ Call “ $MS[2] = update_MS(MS[1] \mid fillet(F1, \rho_p = 2.0))$ ” to remove $F1$ by filleting it with a fillet-radius of 2mm (making it a new fillet of radius 2mm) (Figure 8-b).

4) NC-code, $NC[1] \Leftarrow generate_NC (MS[1]|PP[1], TL[1], MR[1])$

▪ Generate NC-codes for the form cutting operation $P1$ with the FEM20 ϕ (Figure 8-a).

(a) NC-code for form cutting

(b) preform of form-EDM plan ($MS[2]$)

Figure 8. Form-EDM planning example

Stage 2. Clean-up Plan

0) Machining rule, $MR[2] = \{R2, R3\}$: machining rules for *Clean-up* planning.

- R2 = Concave fillets with their radii less than the radius of the finish-cut tool (BEM20φ) are machined with a set of clean-up tools.

- R3 = The clean-up cutting sequence is pencil-cut operations with BEM16 φ, BEM12 φ, and BEM8φ, and strip-parallel fillet-cut with BEM4 φ.

1) Feature extraction, $FL[2] = \{F2\} \Leftarrow extract_features (MS[2]|MR[2], TL[2], TL[3])$

- F2 = *concave fillet* [$\rho_p = 10.0$].

2) Process plan, $PP[2] = \{P2, P3, P4, P5\} \Leftarrow plan_process (FL[2]|MR[2], TL[2])$

- P2 $\equiv (F2, UMO2) = (F2, fillet-cut(BEM4\phi))$.

- P3 $\equiv (F2, UMO3) = (F2, pencil-cut(BEM8\phi))$.

- P4 $\equiv (F2, UMO4) = (F2, pencil-cut(BEM12\phi))$.

- P5 $\equiv (F2, UMO5) = (F2, pencil-cut(BEM16\phi))$.

3) Machined surface, $MS[3] \Leftarrow update_MS (MS[2]|PP[2])$

- MS[3] = *update_MS*(MS[2]| *fillet*(F2), $\rho_p = 10.0$) (Figure 9-b).

4) NC-code, $NC[2] \Leftarrow generate_NC (MS[2]|PP[2], TL[2], MR[2])$

- Generate NC-codes for P2, P3, P4, and P5 (Figure 9-a).

(a) NC-codes for clean-up cutting

(b) preform of Clean-up plan (MS[3])

Figure 9. Clean-up planning example

Stage 3. Finish Plan

0) Machining rule, $MR[3] = \{R4, R5, R6, R7\}$: machining rules for *Finish* planning.

- R4 = The machining tolerance and depth for finish-cut are 0.05mm ($\tau_f = 0.05$) and 0.3mm ($\delta_f = 0.3$), respectively.

▪R5 = Concave fillets with their radii less than $\delta_f + \rho_s = 15.3$ ($\rho_s = 15$ is the radius of the semi-finish tool, BEM30 ϕ) are “pencil-cut” machined before the main finishing.

▪R6 = The machining feature “wall” is machined by a “contour-cut” UMO.

▪R7 = The machining feature “floor” is machined by a “area-cut” UMO.

1) Feature extraction, $FL[3] = \{F3, F4, F5\} \Leftarrow extract_features(MS[3]|MR[3], TL[3], TL[4])$

▪F3 = concave_fillet[$o=0.3, \rho_p=15.0$]. // offsetting followed by filleting

▪F4 = wall[$\alpha=5^\circ, \delta=40 | o=0.3$] : a 5°-draft wall of depth (or height) 40.

▪F5 = floor[$o=0.3$] : offsetting

2) Process plan, $PP[3] = \{P6, P7, P8\} \Leftarrow plan_process(FL[3]|MR[3], TL[3])$

▪P6 $\equiv (F3, UMO6) = (F3, pencil_cut[BEM20\phi])$.

▪P7 $\equiv (F4, UMO7) = (F4, contour_cut[BEM20\phi])$.

▪P8 $\equiv (F5, UMO8) = (F5, area_cut[BEM20\phi])$.

3) Machined surface, $MS[4] \Leftarrow update_MS(MS[3]|PP[3])$

▪ $MS[4] = update_MS(MS[3]|offset, 0.3)$.

▪ $MS[4] = update_MS(MS[4]|fillet(F3), 15.0)$ (Figure 10-d)

4) NC-code, $NC[3] \Leftarrow generate_NC(MS[3]|PP[3], TL[3], MR[3])$

▪Generate NC-codes for P6: BEM20 ϕ pencil-cut (Figure 10-a).

▪Generate NC-codes for P7: BEM20 ϕ contour-cut (Figure 10-b).

▪Generate NC-codes for P8: BEM20 ϕ area-cut (Figure 10-c).

Figure 10. Finish planning example

Stage 4. Semi-finish Plan

0) Machining rule, $MR[4] = \{R8, R9, R10, R11\}$: machining rules for *Semi-finish* planning.

▪R8 = The machining tolerance and depth for SF-cut are 0.15mm ($\tau_s=0.15$) and 2.0mm ($\delta_s=2.0$), respectively.

▪R9 = Concave fillets with their radii less than $\delta_s + \rho_r = 17.0$ ($\rho_r = 15$ is the radius of the rough tool, FEM30 ϕ) are “pencil-cut” machined before the main semi-finishing.

▪R10 = The machining feature “wall” is machined by a “contour-cut” UMO.

▪R11 = The machining feature “floor” is machined by a “area-cut” UMO.

1) Feature extraction, $FL[4] = \{F6, F7, F8\} \Leftarrow extract_features (MS[4]|MR[4], TL[4], TL[5])$

▪F6 = concave_fillet[$\alpha=2.0$].

▪F7 = wall[$\alpha=5^\circ, \delta=40$ | $\alpha=2.0$]

▪F8 = floor[$\alpha=2.0$]

2) Process plan, $PP[4] = \{P9, P10, P11\} \Leftarrow plan_process (FL[4]|MR[4], TL[4])$

▪P9 $\equiv (F6, UMO9) = (F6, pencil_cut[BEM30\phi])$.

▪P10 $\equiv (F7, UMO10) = (F7, contour_cut[BEM30\phi])$.

▪P11 $\equiv (F8, UMO11) = (F8, area_cut[BEM30\phi])$.

3) Machined surface, $MS[5] \Leftarrow update_MS(MS[4]|PP[4])$

▪MS[5] = update_MS(MS[4] | offset, 2.0).

▪MS[5] = update_MS(MS[4] | fillet(F6), 15.0).

4) NC-code, $NC[4] \Leftarrow generate_NC (MS[4]|PP[4], TL[4], MR[4])$

▪Generate NC-codes for P9: BEM30 ϕ pencil-cut.

▪Generate NC-codes for P10: BEM30 ϕ contour-cut.

▪Generate NC-codes for P11: BEM30 ϕ area-cut.

Stage 5. Roughing plan

0) Machining rule, $MR[5] = \{R12, R13, R14, R15\}$: machining rules for Roughing planning.

▪R12 = Rough cutting is performed by pocketing, and the machining depth for the pocketing is 5mm.

▪R13 = A drill hole is needed for the pocketing, and the drill hole is drilled first by a 25 ϕ drill and then by a 50 ϕ drill.

▪R14 = A FEM30 ϕ is used for the pocketing.

▪R15 = The tolerance of rough cutting is 1mm($\tau_r=1.0$).

1) Feature extraction, $FL[5] = \{F9\} \Leftarrow extract_features (MS[5]|MR[5], TL[5], TL[6])$

▪F9 = cavity volume[$\delta=\infty$].

2) Process plan, $PP[5] = \{P12\} \Leftarrow plan_process (FL[5]|MR[5], TL[5])$

▪P12 $\equiv (F9, UMO12) = (F9, pocketing[FEM30\phi])$.

3) Machined surface, $MS[6] \Leftarrow update_MS(MS[5]|PP[5])$

▪MS[5] = update_MS(MS[5] | fill(F9), ∞)

4) NC-code, $NC[5] \Leftarrow generate_NC (MS[5]|PP[5], TL[5], MR[5])$

▪Generate NC-codes for P12: drilling with Drill25 ϕ and Drill50 ϕ , and pocketing with FEM30 ϕ .

6. CONCLUSIONS AND DISCUSSIONS

This paper presents a sequential and tool-oriented CAPP structure for die cavity machining. The proposed CAPP system architecture is a practical one because it is based on industrial practices in die cavity machining. In other words, if a company prefers different rules, the CAPP system easily incorporates them. We developed a prototype CAPP system on the Z-map based CAM software. Preliminary results showed that the proposed CAPP system architecture is practical and valid in die cavity machining.

The novel contribution of the study is more in the formulation of CAPP problem (than in solving the problem). The authors believe that the proposed CAPP system architecture needs more testing and improvement. We then hope that the current contribution will encourage more researchers in the CAPP area to work on the process planning problem in die cavity machining.

The proposed CAPP architecture can be extended to the machining of negative draft areas, which are not machined by 3-axis machining operations. Specifically, if the negative draft areas are extracted from the die surface and a feasible tool orientation is determined, the proposed architecture is applied to the machining of the negative draft areas in the same manner.

ACKNOWLEDGEMENT

The Ministry of Science and Technology of Korea supported this research through a “*National Research Lab Project*” granted to Prof. Choi.

REFERENCES

- Alting, L. and Zhang H. (1989), Computer Aided Process Planning: the state-of-art survey, *Int. Jr. Prod. Res.*, Vol. 27, No. 4, pp. 553-585
- Choi, B. K. and Barash, M. M.(1985), STOPP: an approach to CAD/CAM integration, *CAD*, Vol. 17, No. 4, pp.162-168
- Choi, B. K. (1991), *Surface Modeling for CAD/CAM*, Elsevier
- Choi, B. K., Kim, B. H., and Kim, D. H. (May 1996), Feature Extraction for CAPP of Die Cavity Machining, *IFIP WG5.2 workshop on Geometric Modeling in CAD*, Airlic Center, VA, U.S.A.
- Choi B. K. and Jerard R. B. (1998), *Sculptured Surface Machining*, Kluwer Academic Publishers
- Fallbohrer, P, Akgerman, N and Altan, T, Survey of the U.S. Die and Mold Manufacturing Industry, *ERC/NSM-D-95-41 (1995)*, *ERC for Net Shape Manufacturing*, The Ohio State University, U.S.A.

- Han, J. H. and Rosen, D. (1998), Special panel session for feature recognition at the 1997 ASME computers in engineering conference, *CAD*, Vol. 30, No. 13, pp.979-982
- Kanumury, M. and Chang, T. C. (1991), Process planning in an automated manufacturing environment, *Jr. of Manufacturing System*, Vol. 10, No. 1, pp.67-78
- Lee, J. Y. and Kim, K. (1998), A feature-based approach to extracting machining features, *CAD*, Vol. 30, No. 13, pp.1019-1035
- Lee, Y. S. Choi B. K. and Chang T. C. (1992), Cut distribution and cutter selection for sculptured surface cavity machining, *Int. Jr. Prod. Res.*, Vol. 30, No. 6, pp.1447-1470
- Park S. C. and Choi B. K. (2000), Boundary extraction algorithm for cutting area detection, accepted for publication in *CAD*.
- Pratt, M. J. (1984), Solid modelling and the interface between design and manufacture, *IEEE CG&A*, pp.52-59
- Shin H., Trecapelli R.M., Shudra Jr. A., Chung Y.C. (1998), A generative approach for die pattern machining, *Proc. of An International IFIP TC5/WG53 Conference*, Machining Impossible Shapes, Chrysler Technology Center, Auburn Hills, Michigan, USA, November 9-11 1998, 24-31
- Storr, A., Handel, D., Itterheim, C., Rommel, B. and Strohle, H. (1995), Manufacturing Objects as Uniform NC-Interface for Machining and Measuring, *Proc. of the IFIP WG5.3 PROLAMAT Conference*, Berlin, Germany, pp.356-368
- Xu, X. and Hinduja, S. (1998), Recognition of rough machining features in 21/2D components, *CAD*, Vol. 30, No. 7, pp.503-516
- Yamazaki, K *et al.* (1995), Autonomous process planning with real-time machining for productive sculptured surface manufacturing based on automatic recognition of geometric features, *Annals of CIRP*, Vol. 44 No. 1, pp.439-444
- Yoo Y. S. and Choi, B. K. (1992), CAPP for Die Cavity Machining, *Proc. of the IFIP WG5.3 PROLAMAT Conference*, Tokyo, Japan, pp.437-447