THE ENGINEERING ECONOMIST • FALL 1994 • VOLUME 40, NO. 1

# A COST BASED SYSTEM FOR CONCURRENT PART AND PROCESS DESIGN

BEHROKH KHOSHNEVIS
University of Southern California

JOO Y. PARK
Southwest Research Institute

DUSAN SORMAZ
University of Southern California

---

## ABSTRACT

This paper presents a Real Time Computer Aided Process Planning (RTCAPP) system which can serve as a support tool for economic decision making during design activities. The system is capable of providing the information about manufacturing implication of each added design feature to the designer during the design process of prismatic parts. A knowledge-based system using a hierarchical planning scheme and a multi-bank rule base is developed to generate near-optimal process plans in real time. The planning process is supported by an optimization module which uses dynamic programming to minimize manufacturing costs. The incremental planning mechanism utilizes as much of the existing plan information as is necessary to generate a new process plan, whenever the design is updated. The system serves as an effective concurrent engineering tool that can be used to provide real time feedback on the manufacturing cost consequences to the designer.

---

## INTRODUCTION

Increasing recognition of cost competition has spurred the development of manufacturing technology, planning processes, and computer-based controls, among other systems and procedures, all intended to reduce costs and/or improve product quality. Of particular interest there are two recent developments, computer aided design (CAD) and automated process planning (APP). Although the results of these developments are impressive, the question of directing these developments as major means of creating designs with lower manufacturing costs has not been emphasized.

If manufacturing cost estimating could be integrated into the design process, the cost impact of alternative designs could be evaluated. This would result in the selection of a design which would represent a desirable trade-off between product performance and cost. However, due to the time and effort required to evaluate each design alternative and its cost consequence, the above integration task has been an intractable problem to date. With the advent of modern computer science and industrial engineering techniques, it is now more feasible than ever to consider a cost based design strategy that would place the tools of a manufacturing cost estimator at the disposal of the design engineer.

A number of prior studies indicate that significant opportunities exist for substantial cost savings if value analysis could be integrated in the design process. For example, in a survey carried out in West Germany, value analysis for 135 products from 42 companies were surveyed [7]. These analyses were carried out approximately in the mid point of the product life cycles. The results: "Manufacturing costs were reduced by 35% for vehicles, 40% for precision products, and 30% for machine tools, giving a mean of 33%. Of the 33% mean saving, 65% was derived from the design sector, 19% from production planning sector and 15% from purchasing." Clearly, if relevant cost information is provided at the design stage rather than later in the product life cycle, substantial benefits are to be expected.

## CURRENT PRACTICE

The above observations notwithstanding, the problem has not been totally ignored. Based on the authors' observations many manufacturing firms at least utilize a liaison function between manufacturing and design engineering to resolve problems. But the liaison function is often more concerned with manufacturability than cost. The insensitivity to cost may have resulted from lack of serious competition in many areas. But in todays competitive world market the manufacturers must become more rigorous in their approaches to cost estimating during design. Although the recent developments in CAD and APP show a great promise, the resulting systems have been used in isolation from one another, that is, the product design and manufacturing process plans are developed independently, with the product design representing constraints on the optimal manufacturing process design.

A principle feature of CAD is the ability to analyze design configurations quickly and accurately. However, CAD systems currently in use limit these analyses to technological considerations: weight, surface area, moments of inertia, stresses, deflections, and the like. Product performance is the primary consideration. Although CAD permits the designer to consider a relatively large number of alternative configurations and materials, the economic consequences of intermediate iterations are unspecified. It is only when the final design has been completed that the economic analysis may be effected. If the designer is not influenced by economic consequences during the iterative stages of design, then it is quite probable that lower cost alternatives will be overlooked.

On the other hand, a considerable amount of research has been done to create various automated process planning systems during the last decade. The motivation in part has been due to the increasing need for the already scarce expert human process planners and for bridging the gap between CAD and CAM. The current automated process planning systems, which all operate in a batch mode, try to generate a detailed instruction on manufacturing steps for the designed part. Recently, there has been a turning point in process planning system methodology in accordance with the advent of artificial intelligence. Artificial intelligence can perform certain functions such as reasoning, planning and problem solving which are associated with human intelligence and expertise. Especially, expert system and AI planning techniques have been used to develop generative process planning systems. GARI [4], TOM [11], PROPLAN [14], HI-MAPP [2], Turbo-CAPP [19], and DYNACAPP [10] are some of the AI based systems that have been reported. These newer systems like the previous ones are essentially concerned with the automation of manufacturing process planning function rather than serving as a vehicle for improving the design itself.

## THE NEW APPROACH

The fundamental here is that the engineering designer be provided timely estimates of the economic consequences of possible design alternatives. Means must be developed to estimate rapidly the manufacturing cost consequence of each design feature as it is being added during the design process. This necessarily implies that, for any candidate design feature, the associated manufacturing process plan can be described on a real time basis. Creation and evaluation of economic based process plans constitutes the essence of the developments discussed in this article.

This paper describes a new process planning system called Real Time Computer Aided Process Planner (RTCAPP). The system is capable of generating a detail process plan for an incrementally updated part design described in terms of a collection of machining features usually found on prismatic parts. RTCAPP dynamically interacts with the designer. In order to accomplish this work several knowledge bases are created and a hierarchical planning technique is utilized. The system characteristics and structure will be explained in the following sections and an example will be included for further clarification. Several theoretical concepts, which are applied in the organization of the planning system will be addressed.

### REAL TIME INTERACTION SYSTEM

During a product design process the design engineer typically starts with a simple representation of the part boundary and subsequently adds new features to this representation. The designer may also alter certain feature elements of the part design. The alterations are especially expected if the designer intends to evaluate various alternative features that meet the same functionality requirements. Clearly, if an entire process plan is to be developed for each state of the design, the process will be very lengthy and time consuming even on a fast computer. This is true, because automated process planning involves a search through a vast space of possibilities.

Given that a process plan is developed for the current state of the design, one of the two possibilities are expected after each addition or alteration of the design: First, the change in design may not violate the nature of the process plan to date, in which case just an attachment will be made to the end of the existing plan to include the processes needed for the added feature. The process in this case is high speed and straight forward. Second, the change may violate the current plan by imposing a need to backtrack and re-sequence and/or change a portion of the recommended processes for the features added at the earlier design stages. In most instances such a procedure would rearrange and modify only a small portion of the existing process plan, and again would be possible to perform in real time.

One interesting observation in creating a real time interaction between the human designer and the computer automated process planning system is that not only the design process can benefit from the process planning module, but the process planning module itself can greatly benefit from the information about

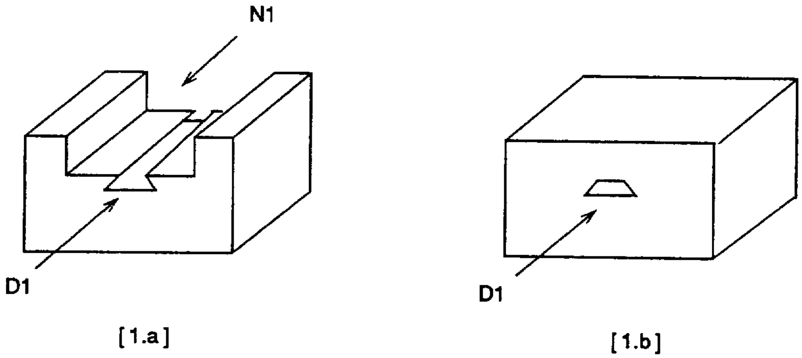


**FIGURE 1.** Feature addition sequence in design.

the logical sequence of the added features inherent in the action of the human designer. For example, consider in Figure 1.a that a design engineer would most likely add the straight notch feature prior to adding the dove-tail notch. This is because the addition of the former feature generates a new surface that provides a convenient reference for the dove-tail notch feature. It is very unlikely that a designer would choose to start with the dove-tail notch, as shown in Figure 1.b, and then add the straight notch feature.

In the conventional batch mode process planning systems, the input to the process planning environment is the finalized design only, and the valuable information about the human designer's sequence of feature addition is nonexistent. This can create a huge search space, as the process planning module has to try various sequence possibilities, many of which take a long time to evaluate and which end with failure. There is a strong correlation between the designer's choice of sequencing the feature addition and the actual manufacturing process sequences. RTCAPP effectively uses this correlation to its advantage, and as a result it can generate process plans many times faster than the conventional systems.

### HIERARCHICAL PLANNING

The process planning function can be viewed as a decision making activity in which input to the system is all the design descriptions about a part and output is a detailed specification of process type, process sequence, proper machines, resting faces, cutting tools and cutting parameters to be used to manufacture a given part.

In order to implement process planning more efficiently, it is important to be able to clarify the process planning output and eliminate the details until the solution addressing the main issues is found. A hierarchical planning methodology has been developed in the generic planning system, ABSTRIPS [16], which plans in a hierarchy of abstraction spaces, in each of which preconditions at a lower level of abstraction were ignored. The general problem solving system, NOAH [17], also operates hierarchically in that it constructs first an abstract skeleton of a plan and then, at successive steps, fill in more and more detail. The hierarchical planning approach was applied in a previous process planning system. Berenji and Khoshnevis [2] embedded this approach implicitly into HI-MAPP's AI planning mechanism called DEVISER. The conceptual structure of a hierarchical process planning scheme was also suggested by Subramanyan and Lu [18].

In a manufacturing process planning task, the major issues in planning are the generation of sequence of features to cut, and a feasible process selection procedure for each feature. The proper machine or tool selection and cutting parameters may be assigned in later stages of planning. If all the details of a process plan are operated upon entirely, the planning process would take an unreasonably long time because of the vast search space involved in the plan. The hierarchical approach decomposes the problem into manageable sizes that reduce the search time and still allow for reaching the near-optimal results.

The implementation of rule based systems needs a close observation. Rule matching is used extensively in rule-based programs and the execution can be very slow when large number of rules and facts are involved. Some expert systems have been reported to spend more than nine tenths of their total run time performing rule matching [5].

To make the rule matching process many times more efficient, RTCAPP uses the rule decomposition scheme in conjunction with the hierarchical planning structure. The hierarchical structure separates the whole knowledge base into several independent rule bases each of which contains a family of rules such as Preconditional Relationship Rules, Process Selection Rules, and Tool and Resting Face Selection Rules. The classification of rules is feasible because output of propagated rule from a higher level rule base can only affect the engagement process of the lower level rules but not vice versa. This allows for search of alternatives at higher levels (such as feature sequence) without needing to evaluate all the details down to the level of machining parameter settings. The lower level issues are addressed only if the higher level decisions are already established. The impact of the hierarchical structure is, therefore, on the plan execution time.

## MANUFACTURING COST ESTIMATION

Generally, the term *manufacturing cost estimation* refers to any estimation of the costs involved in the manufacture of a component part, subassembly or final assembly. The manufacturing cost estimation includes material cost, labor cost, machine cost and tool cost. The estimation could be accomplished by various methods [Verno68]: conference, comparison, or detailed analysis. The most reliable method is the detailed analysis method since it includes a complete examination of all important factors in the manufacturing of a part. However, this method is more time consuming than the other methods, but with the power of current computers and computing methods the detailed analysis has become more popular.

The detailed analysis of the manufacturing cost is based on consideration of the time necessary to complete a given manufacturing operation. Manufacturing time consists of the following components: machining time, tool change time, part handling time, set-up time and transportation time. The manufacturing cost per unit workpiece may be expressed as:

$$C_p = \sum_{\text{all machines used}} \frac{C_b + C_{tr}}{N_b} + \sum_{\text{all features}} C_t\left[t_m + t_h + \frac{t_m}{T}\left(t_t + \frac{C_o}{C_t}\right)\right]$$

where

$C_p$ = manufacturing cost per part (\$/part)
$C_b$ = setup cost per batch (\$/batch)
$C_t$ = machine operation cost (\$/minute)
$C_o$ = tool cost (\$/tool) ( e.g., for HSS tool it is the regrinding cost, for carbide tool it is the cost of one insert cutting edge)
$C_{tr}$ = transportation cost between two subsequent machines
$N_b$ = batch size
$t_m$ = machining time for each feature
$t_h$ = part handling time for each feature operation
$T$ = average tool life
$t_t$ = tool change time for each feature

All of the above components have to be evaluated (computed or estimated) in order to estimate the manufacturing cost. The feature machining time can be very accurately calculated using machining parameters (such as cutting speed and feed) or material removal rates for desired machining processes. An example may be drilling time, which is computed using the following formula:

$$t_m = (h + 0.5\ d)/(nf)$$

where

$h$ = hole depth (in)
$d$ = hole diameter (in)
$n$ = v/d. = tool speed (revolution/minute)
$v$ = cutting speed (in/min)
$f$ = tool feed (in/revolution).

Tool change time depends on cutting speed and on the famous Taylor's tool-life / cutting-speed equation for a given tool and its material. The process planning system may influence this cost very significantly by selecting appropriate cutting speeds. Other time components, including part handling time, set-up time and transportation time, depend on selected processes and machines for two subsequent features in the sequence of feature processing.

RTCAPP performs the manufacturing cost estimation for each feature (and process/machine/tool/resting-face combination) using the method of detailed analysis and estimation based on lookup tables. These cost estimates serve as the basis for selection of the process plan with minimal cost using the optimization procedure described in the following section.

### DYNAMIC PROGRAMMING

Dynamic programming is a mathematical optimization technique often used for making a sequence of interrelated decisions. It provides a systematic procedure for determining the combination of decisions that maximizes an overall effectiveness measure [1]. RTCAPP employs the dynamic programming technique to accomplish the economic machine and tool selection task for a given part design. The problem can be divided into several stages each representing the feature to be machined. Each stage has a number of states each representing the candidate machines and tools. The effectiveness measure is the estimated manufacturing cost function. The cost function at each state of a given stage provides the cost of transition from the current state into a state associated with the next stage.

According to the principle of optimality in dynamic programming, given the current stage, an optimal machine and tool selection decision for the remaining stages is independent of the selection adopted in previous stages. A recursive relationship that identifies the optimal selection for each state at stage $N$, given the optimal policy for each state at stage $N$-1, is obtainable by a recursive relationship. Using the recursive relationship the solution procedure moves forward (or backward) stage by stage, each time finding the optimal selection for each state of that stage until it finds the overall optimal selection from the beginning to the end stage.

The characteristic of recurrence and independence in dynamic programming provide for the incorporation of cost optimization in the incremental process planning approach. After each addition of the design, the design change may violate the current process plan and require a backtracking and a subsequent resequencing of operations. This addition, however, could rearrange only a small portion of the current process plan and thereby leave the cost computations for the unaffected stages unaltered and useful for the optimization of the modified plan. Other optimization techniques would generally require the solving of the entire problem, regardless of how small the change is in the design. More details on this issue are provided in the Plan Maintenance Module subsection presented later in this paper.

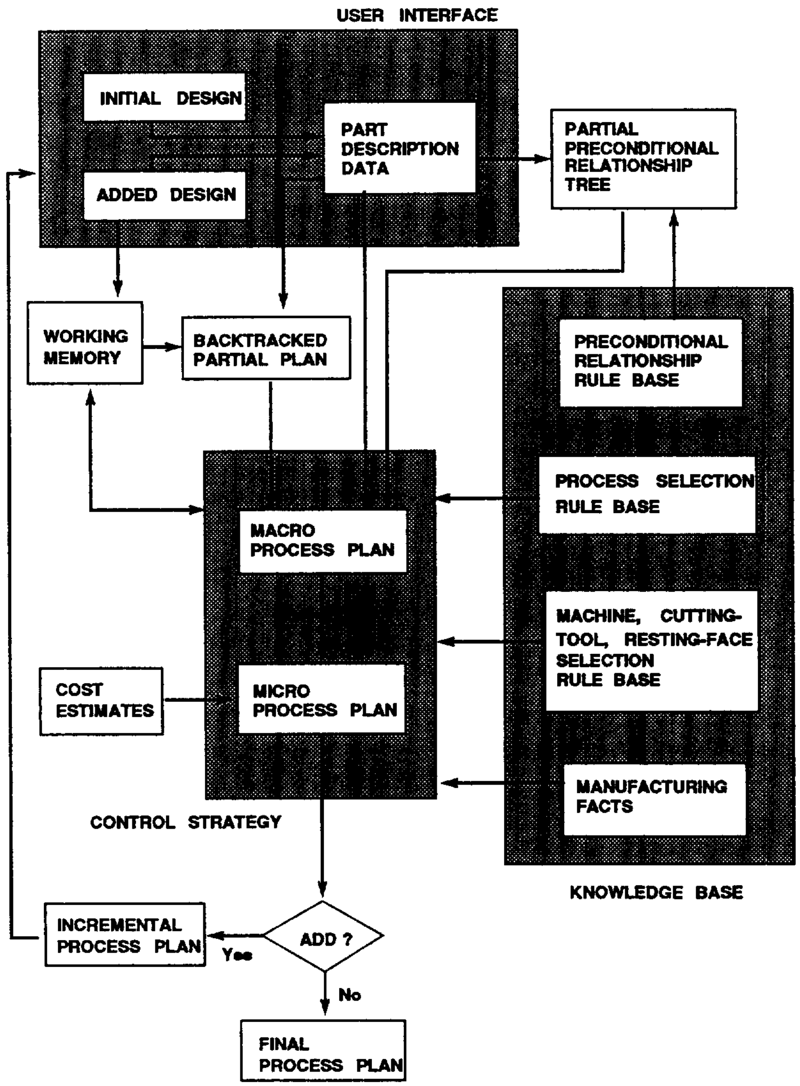
## SYSTEM ARCHITECTURE

The modular structure of RTCAPP is illustrated in Figure 2. The RTCAPP system considers the idea of incremental planning and a combination of symbolic and numeric procedures in its planning process. The following subsections describe the major components of the system structure.

### USER INTERFACE MODULE

The activities performed at the User Interface Module represent the communication between the user (e.g. designer) and the system, and the transformation

FIGURE 2. RTCAPP system structure.



of the information on the design into a formalized description for understanding the designed part. An interactive feature identification method is used for the activities, where a set of features of a given part is identified and the properties of each feature are described.

Since humans have a well developed pattern recognition ability, the interactive feature identification method using the designer's input is still the most reliable means of feature identification. A solid modeler and an automatic feature extraction mechanism are being used in our current prototype systems.

After the information on each feature is acquired, the intermediate part description and the corresponding process plan are generated by RTCAPP. Since the designer may not be particularly interested in the details of the process plan but rather in the total cost figure associated with the plan, a logical addition is a process plan evaluator that computes the cost of implementing each of the generated plans. The user may choose to change or update the current design and input an incremental part information repeatedly and compare the cost values until satisfactory results are obtained.

The representation of a given part includes: a) part properties including name, material, hardness, batch size, and the feature list; b) feature properties including feature type, location, dimension, tolerance, surface quality, and normal vector direction; and c) interrelationships between features including parallelity, perpendicularity, intersection between two features, and relative location of two features with respect to one another.

Each input information is coded in a special description language. The description language is based on the work of Jared's [7] and has been also used by GARI and HI-MAPP. In this language, each feature of the part is interpreted as a working element in a machining process. This representation not only specifies the feature, but it also identifies the feature relationship with some of the other existing features in the overall part design.

### PARTIAL PRECONDITIONAL RELATIONSHIP TREE MODULE

Process planning is concerned with identification and arrangement of the required machining processes in a proper (if possible, optimal) order for a given set of prescribed features of a part. Because of the large number of possible arrangements of orders, the planning process would be more efficient if a partial preconditional relationship of the features is defined that eliminates a large number of unattractive or infeasible alternatives. Knowledge of economic, geometric and technological properties in feature manufacturing is instrumental in constructing such relationships.

A preliminary preconditional relationship can be built on the basis of the design evolutionary steps which are intuitively identified by the designer through the user interface module. For example, in the 4th step of Figure 3.b it is reasonable to establish the preliminary preconditional relationship as the following:

FZ1 > H2, FZ2 > H2, FX2 > H1, N1 > H1, FX1 > N1.

where, FX1, FX2, FZ1, and FZ2 are outside faces, H1 and H2 are holes, and N1 is a notch (slot). The notation Fi > Fj implies that operation of feature Fi precedes that of feature Fj.

Certain technological constraints imposed by a newly added feature may violate an already feasible preconditional relationship. The designer, however, may not always be aware of these constraints. These types of technological constraints, therefore, must be compiled in a rule base for automatic reference. RTCAPP stores these rules in its Preconditional Relationship Rule Base.

A sample rule related to the technological constraints is that if a notch has a certain depth and one of its sides is thinner than a certain limit, then attempting to drive a drill bit into this side may deflect the part and result in a poor quality hole (out of geometric tolerance). Assuming that this rule applies to the example part in Figure 3, then after referring to this rule, making of hole H2 must precede the making of notch N1 and the above partial preconditional relationship should be modified as (see the 4th step in Figure 3.b):

H2 > N1, FZ1 > H2, FZ2 > H2, FX2 > H1, N1 > H1, FX1 > N1.

This updated partial preconditional relationship information is to be fed into the Macro Process Planning Module.

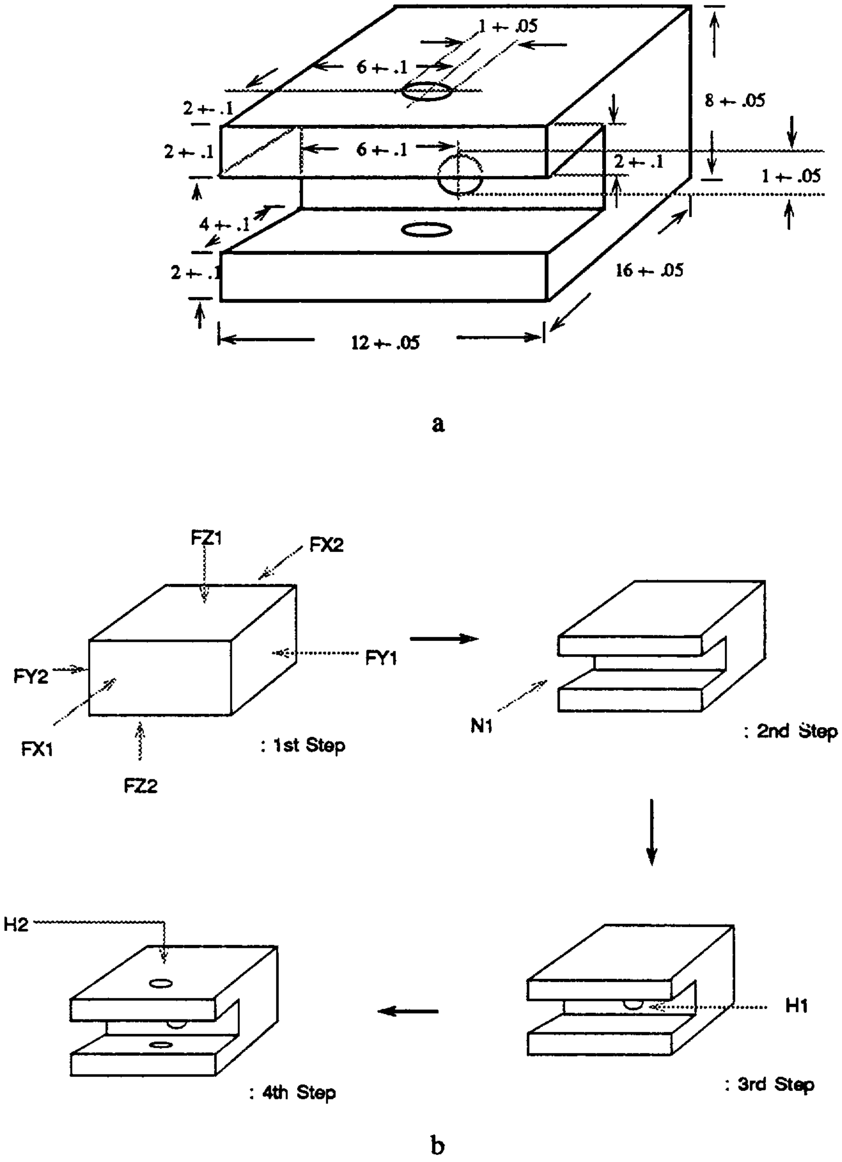
### MACRO PROCESS PLANNING MODULE

The function of the Macro Process Planning module is to establish a sequence for processing the features in such a way that the least amount of change of state with regards to machine, tool, and resting face may be attainable. The output of this module is a feature sequence only. The specific machines, tools and resting faces to be used are identified at the Micro Process Planning Stage. In essence the macro planning level prepares a good partial solution that can be further enriched by the micro planning level.

At this module for the newly created feature a feasible set of processes is selected by referring to the Process Selection Rule Base. Each process in the selected set may be performed by various machines, and each machine may use various cutting tools to perform a given process. Moreover, the part may have alternative resting faces on each machine.

A set of various feasible manufacturing parameter combinations (i.e. the set of all possible machines, cutting tools, and resting faces) can be generated for the alternative processes retrieved from the Process Selection Rule Base using the

FIGURE 3. Design update steps for a part.



information taken from the Machine and Tool Data Bases. For a given feature the retrieved set has the following format:

*(feature.name feature.type*
*(machine.i cutting_tool.j resting_face.k process.l)*
............
*(machine.m cutting_tool.n resting_face.o process.p))*

A heuristic search procedure based on 'generation and best selection' approach uses the above information along with the information on the Partial Preconditional Relationship constraints to find the best route from the initial design state (raw material) to the current state of design (the goal state). Followings are the steps of this search procedure:

Step 1. Generate all possible nodes (features) for open successors observing the partial preconditional relationship constraints among features.

Step 2. Evaluate the utility of each open successor node by applying the Conditions Overlap Evaluation Function (described later) and select the node which has the maximum value of this function.

Step 3. If the current state of the best node is equivalent to the goal state (finished part) then quit, otherwise go to step 1.

In step 1 above, the list of candidate open successors for a given node includes the unselected nodes and the newly added nodes for which the precondition is satisfied. If the current state is that of the raw material then the open successors are those nodes (features) which have no preconditional requirements (see Figure 4).

The Conditions Overlap Evaluation Function used in step 2 provides a measure of the extent to which an open successor node shares the conditions (machine, tool, and resting face) of the predecessor node. For each of machine, tool, and resting face condition overlap a weight factor (constant) is assumed. If a successor node shares any of the predecessor node's conditions, the corresponding weight factor is added to the evaluation function. The weight factors are given different values such that the highest priority is given to machine, then to tool and then to resting face. A higher value of the evaluation function, therefore, corresponds to lower cost sequences that incur as little changes in processing conditions as possible.

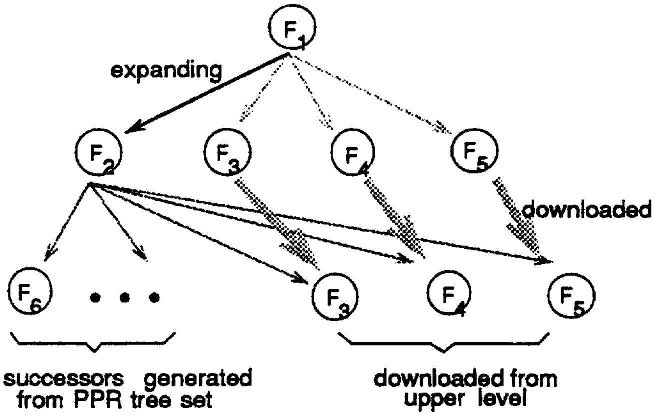


FIGURE 4. Generation of features in the search space.

MICRO PROCESS PLANNING MODULE

Given the feature processing sequence, which is established at the Macro Process Planning stage, the Micro Process Planning module uses dynamic programming (DP) to determine the optimal allocation of machines, tools and resting faces applicable to the given feature processing sequence. The steps involved at this module are as follows (see Figure 5):

Step 1. Identify the current feature as the current DP stage (starting from the first feature in sequence), and retrieve the set of parameter combinations for the feature from the Macro Process Plan. Identify each machine in each combination as a DP state.

Step 2. For all feasible parameter combinations, estimate setup cost, tooling cost, part handling cost, and machining cost. Associate the corresponding costs of possible machine, tool and resting face changes with the related state transition for each state.

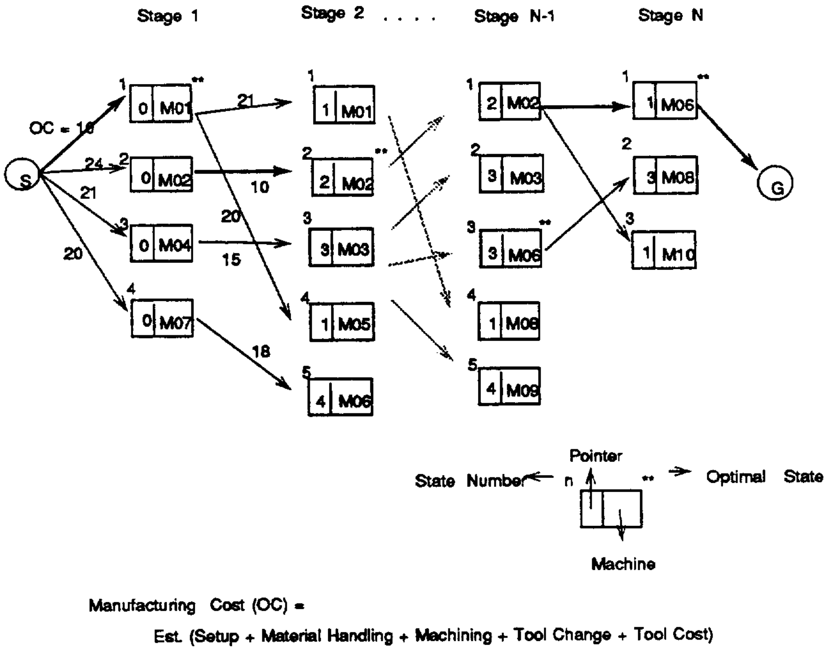


FIGURE 5. Dynamic programming network for micro process planning.

Step 3. Apply the dynamic programming technique recursively to find the optimal machine sequence (with the least cumulative manufacturing cost) from the start stage to the current stage. If arrived at the last feature stage, then stop, otherwise, go to step 1 for implementing the next stage.

In Step 2, the machining cost is calculated from the optimal cutting conditions data (depth of cut, speed, feed rate) which are stored in the Manufacturability Data Base. Setup and tool change costs are compiled in the Machine and Tool Data Base.

### PLAN MAINTENANCE MODULE

Upon updating the design information through the incremental feature description, the system creates a Partial Preconditional Relationship Tree for the newly added feature and traces back through the path which represents the macro feature sequence to date. The system then tries to find a node (Restart Node) in the macro feature sequence which satisfies the precondition necessary for processing the new feature. The macro planning process, which is basically concerned with process selection and feature sequencing, restarts from this node onward.

Upon introduction of a new feature one of the following conditions may occur:

Condition I. The subplan of the new feature can be simply attached to the end of current process plan, that is one of the last nodes in the PPR Tree meets the precondition for processing the newly added feature.

Condition II. The subplan of the new feature violates the current process plan by imposing a need to rearrange and modify a section of the current process plan.

The method used to support the Macro Process Planning function under condition II above uses the following steps:

Step 1. Define a boundary zone on the current process plan within which all features can be candidates for serving as a Restart Node.

Step 2. Select the most promising Restart Node among all candidates by evaluating the utility of each candidate in the boundary zone using the Conditions Overlap Evaluation Function (explained below).

To clarify the above procedure consider the 4th step in Figure 3.b, where the newly created incremental preconditional relationship upon the addition of feature H2 can be described as:

$$H2 > N1, \quad FZ1 > H2, \quad FZ2 > H2,$$

which is taken from the designer input and the Preconditional Relationship Rule Base. As shown in Figure 6, the system backtracks until it finds the Restart Node, FZ1, which satisfies its precondition and restarts the heuristic search until it finds the newly updated goal state.

The change in the macro plan should be reflected in the micro plan as well. The Plan Maintenance Module traces back through the dynamic programming network until it meets the stage whose process is identical to the Restart Node found by the above procedure, and makes the recursive decisions onwards. The DP computations prior to the Restart Node stage which are kept in the memory

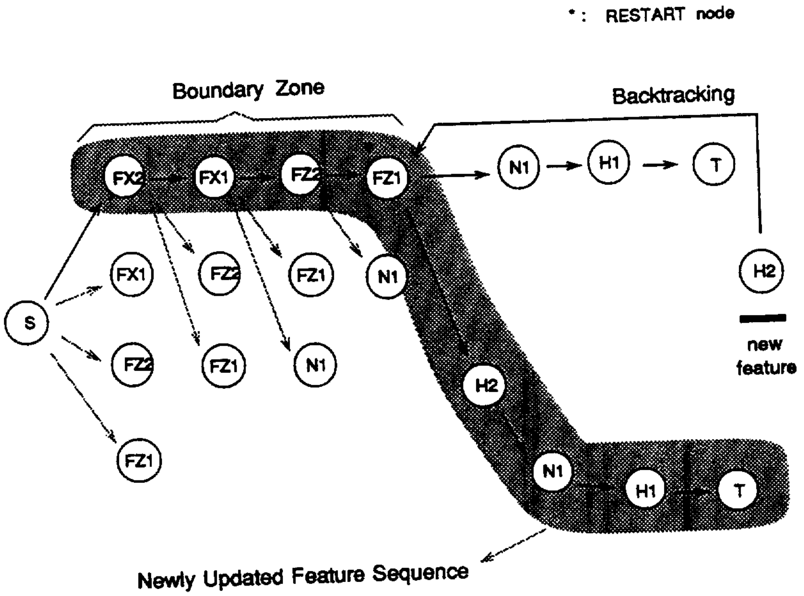


FIGURE 6. Backtracking resulting from design update.

are unaltered and valid for use. The process plan generated by this approach does not violate the required manufacturing conditions, and includes a reasonable and efficient process sequence which is found in a relatively fast manner.

### DOMAIN SPECIFIC KNOWLEDGE BASE

The knowledge base used by RTCAPP contains two different classes of information. These are:

1) Manufacturing facts: Information about available machines, cutting tools and part descriptions. It encompasses the Machine and Tool Data Base, Part Description Data Base, and Manufacturability Data Base. The manufacturing facts are represented as frames. In each frame individual instances share the common knowledge through inheritance, or define their own properties.

2) Manufacturing rules: Include the Preconditional Relationship Rule Base, Process Selection and Sequencing Rule Base, Machine, Tool and Resting-face Selection Rule Base.

A Preconditional Relationship Rule defines the partial relationship among features in certain condition. For the example which was explained earlier (see the subsection of Partial Preconditional Relationship Tree Module), the representation has the following form:

```
((if (type ?x straight-hole) (type ?y slot)
 (normal-vector ?x ?d1) (normal-vector ?y ?d2)
 (not = ?d1 ?d2) (not opposite ?d1 ?d2)
 (intersect ?x ?y) (depth ?x ?d)
 (width ?y ?w) (> ?w (* 0.6 ?d)))
(then (do-after ?y ?x)))
```

A Process Selection Rule sets the conditions for selecting candidate processes. For example, if the feature is a face whose required quality, *q1*, is higher than the quality of the raw material, *q2*, then a set of feasible recommendations is:

```
((if (type ?x face)
 (required-quality ?x ?q1)
 (current-quality ?x ?q2)
 (< ?q1 ?q2))
(then (recommended-process ?x
        (face-milling slab-milling)))
```

Tool and Resting Face Selection Rules provide, for each process, the recommendations on the tool and the face of the part to rest on. For instance, if the recommended process is face milling, and the recommended machine type is milling machine, and there exists a top face whose normal direction is opposite to the normal direction of the feature, and a special tool which is for face milling is available, then these rules indicate the available tool and the top face to rest on:

```
((if (recommend-process ?x face-milling)
 (recommend-machinetype ?x milling-mc)
 (machine-type ?m mill-mc)
  ; match Manufacturing Facts
 (tool-for ?t face-milling)
 (tool-material ?x face-milling ?tm)
 (tool-diameter ?t ?d)
 (feature-width ?x ?w)
 (call.check.diameter ?d ?w))
(then (recommend-machine ?x ?m)
 (recommend-tool ?x ?t)))
((if (process ?x ?p)
```

```
(machine ?x ?m)
(tool-head ?m vertical)
(normal-vector ?x ?nv1)
(normal-vector ?y ?nv2)
(?y topface)
(opposite-direction ?nv1 ?nv2))
(then (recommand-restingface ?x ?y ?m)))
```

## AN EXAMPLE

The part shown in Figure 3 is selected in this example. It is assumed that the designer calls for the process planning feedback after the inclusion of hole H1 and notch N1 (i.e the 3rd step of part design in Figure 3). The associated part representation and process plan are shown in Figures 7 and 8, respectively.

For the 4th step in the part design process the intermediate part representation, and the updated process plan are shown in Figure 9. Notice in these figures that after the addition of the new feature, H2, only a portion of the process plan has been affected, that is, although the addition requires some backtracking and change of some of the previous sequences, the first four (out of seven) steps of the process plan remain unaltered.

## CONCLUSION

A major emphasis in concurrent engineering is on the simultaneous creation of product design and its manufacturing plan. This concurrence which results in the increased awareness of designers of the manufacturing cost consequences of their designs is shown by empirical studies to have significant potential impact on the overall product cost. This paper has presented a new approach to process planning, which allows for creation of real time feedback of manufacturing cost consequences to the engineering designer.

The system presented in this paper is at the prototype stage. Further developments are underway to connect RTCAPP to a solid modeler for automatic representation of design features. The cost analysis approach used in RTCAPP may be enhanced dramatically by including more elaborate value analysis and cost estimating techniques. The design of a flexible cost evaluation module for RTCAPP that allows for user inclusion of various cost evaluation methodologies applicable to various manufacturing domains is underway.

**FIGURE 7.** Part representation for the sample part.

```
>> The property list of FX1 FX2 FZ1 FZ2 ::

      (type FX1 topface) (quality FX1 0.0002)
      (normal_vector FX1 x_pos) (dimension FX1 (8.0 12.0))
      (open_from FX1 nothing)

      (type FX2 topface) (quality FX2 0.0002)
      (normal_vector FX2 x_neg) (dimension FX2 (8.0 12.0))
      (open_from FX2 nothing)

      (type FZ1 topface) (quality FZ1 0.0002)
      (normal_vector FZ1 z_pos) (dimension FZ1 (12.0 16.0))
      (open_from FZ1 nothing)

      (type FZ2 topface) (quality FZ2 0.0002)
      (normal_vector FZ2 z_neg) (dimension FZ2 (12.0 16.0))
      (open_from FZ2 nothing)

      (distance FX1 FX2 16.0) (tolerance FX1 FX2 0.05)
      (distance FZ1 FZ2 8.0) (tolerance FZ1 FZ2 0.05)

>> The property list of N1 ::

      (type N1 notch) (quality N1 0.00025)
      (normal_vector N1 x_pos) (width N1 4.0)
      (length N1 12.0) (depth N1 4.0)
      (open_from N1 FX1)
      (children N1 P1) (direction P1 x_neg)
      (children N1 P2) (direction P1 x_pos)
      (children N1 P3) (direction P1 z_pos)

>> The property list of H1 ::

      (type H1 straight_hole) (quality H1 0.00020)
      (ptolerance H1 0.05)
      (normal_vector H1 x_pos) (normal_vector H1 x_neg)
      (diameter H1 1.0) (depth H1 8.0)
      (open_from H1 N1) (end_into H1 FX2)
```

**FIGURE 8.** Process plan output after the 3rd feature addition.

```
                  ********************
                   PROCESS PLAN OUTPUT
                  ********************

------------------------------------------------------------------
 No. Feature  Process    Machine   Tool  Resting_Face   Cumul_Cost
     SetUp  Handling  Machining   Toolchange  Tool      Total_Cost
------------------------------------------------------------------

 1   FZ2    FACE.MILLING    MC100  T130     FZ1           5.605
     0.72     0.6       3.83       0.18       0.275       5.605

 2   FZ1    FACE.MILLING    MC100  T130     FZ2           10.61
     0.72     0.0       3.83       0.18       0.275       5.005

 3   FX2    FACE.MILLING    MC100  T130     FX1           14.604
     0.72     0.0       2.887      0.18       0.207       3.994

 4   FX1    FACE.MILLING    MC100  T130     FX2           18.598
     0.72     0.0       2.887      0.18       0.207       3.994

 5   N1     SIDE.MILLING    MC100  T260     FX2           33.218
     0.0      0.0       13.398     0.18       1.041       14.62

 6   H1     DRILLING        DR200  T520     FX2           36.976
     0.12     0.6       2.578      0.03       0.43        3.758

 7   H1     REAMING         DR200  T560     FX2           38.61
     0.0      0.0       1.5        0.03       0.104       1.634
```

**FIGURE 9.** Feature representation and process plan output after the 4th feature addition.

```
>> The property list of H2 ::

      (type H2 straight_hole) (quality H2 0.0002)
      (ptolerance H2 0.05)
      (normal_vector H2 z_pos) (normal_vector H2 z_neg)
      (diameter H2 1.0) (depth H2 8.0)
      (intersect H2 N1)
      (open_from H2 N1) (end_into H2 FX2)

                    *********************
                     PROCESS PLAN OUTPUT
                    *********************

   ----------------------------------------------------------------
    No. Feature  Process   Machine  Tool  Resting_Face  Cumul_Cost
        SetUp  Handling  Machining  Toolchange  Tool    Total_Cost
   ----------------------------------------------------------------

    1   FZ2   FACE.MILLING   ML200  T130    FZ1         6.13
        0.384   1.44      3.83      0.201     0.275     6.13

    2   FZ1   FACE.MILLING   ML200  T130    FZ2         10.783
        0.384   0.0       3.83      0.165     0.275     4.654

    3   FX2   FACE.MILLING   ML200  T130    FX1         14.386
        0.384   0.0       2.887     0.124     0.207     3.603

    4   FX1   FACE.MILLING   ML200  T130    FX2         17.989
        0.384   0.0       2.887     0.124     0.207     3.603

    5   H2    DRILLING       DR200  T520    FZ2         21.746
        0.12    0.6       2.578     0.03      0.43      3.758

    6   H2    REAMING        DR200  T560    FZ2         23.381
        0.0     0.0       1.5       0.03      0.104     1.634

    7   N1    SIDE.MILLING   MC100  T260    FX2         39.32
        0.72    0.6       13.398    0.18      1.041     15.94

    8   H1    DRILLING       DR200  T520    FX2         43.078
        0.12    0.6       2.578     0.03      0.43      3.758

    9   H1    REAMING        DR200  T560    FX2         44.712
        0.0     0.0       1.5       0.03      0.104     1.634
```

## ACKNOWLEDGEMENT

The research reported in this paper has been supported by a NSF Industry-University Cooperative Research Center administered by the Institute for Manufacturing and Automation Research (IMAR).

## REFERENCES

[1] Bellman, R., Dreyfus, S., *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, N.J., 1962.

[2] Berenji, H., Khoshnevis, B., "Manufacturing Generative Process Planning with Relational Production System," *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, Nov., 1985.

[3] Brown, D. R., M.R. Cutkosky and J.M. Tenenbaum, "Next-Cut: A Computational Framework for Concurrent Engineering," 2nd International Symposium on Concurrent Engineering, February 1990, West Virginia.

[4] Descotte, Y., Latombe, J., "GARI: A Problem Solver that Plans How to Machine Mechanical Parts," *Proceedings of 7th International Joint Conference on Artificial Intelligence*, Aug., 1981.

[5] Fleischer, G., Khoshnevis, B., "Incorporating Economic Impact Assessment into Computer-Aided Design," *Proceedings of the Institute of Industrial Engineering Conference*, May, 1986.

[6] Forgy, C., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence*, Vol. 19, 1982.

[7] Gregory, S., "Design for Cost Manufacture in West Germany", *Design Studies*, Vol 4. No. 4, Oct. 1983, pp. 245-257.

[8] Jared, G., "Shape Features in Geometric Modeling," *Solid Modeling in Computer*, edited by J. Boyse and M. Pickette, Plenum Press, New York, 1983.

[9] Kutay, A. and S. Finger, "Integration of Economic Information with Design and Manufacturing Systems," Technical Report No. CMU-RI-TR-91-02, Carnegie Mellon University, 1991.

[10] Khoshnevis, B. and Q. Chen, "Integration of Process Planning and Scheduling Functions," *Journal of Intelligent Manufacturing,* Vol 1., 1990, pp. 165-176.

[11] Matsushima, K., Okada, N., Sata, T., "The Integration of CAD and CAM by Application of Artificial Intelligence Techniques," *Annals of the CIRP*, Vol. 31, No. 1, 1982.

[12] Park, J. Y., "Incremental Process Planning as a Support Tool for Economic Product Design," Unpublished Ph.D. Dissertation, University of Southern California, May 1990.

[13] Park, J. Y., and B. Khoshnevis , "A Real-Time Computer-Aided Process Planning System as a Support Tool for Economic Product Design," *Journal of Manufacturing Systems*, Vol. 12, No. 2., 1993, pp 181-193.

[14] Phillips, R., Zhou, X., Mouleeswaran, C., "An Artificial Intelligence Approach to Integrating CAD and CAM through Generative Process

Planning," *Proceedings of ASME International Computers in Engineering Conference*, Las Vegas, Nevada, 1984.

[15] Rich, E., *Artificial Intelligence*, McGraw Hill, 1983.

[16] Sacerdoti, E. D., "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence*, Vol. 5, 1974.

[17] Sacerdoti, E.D., "The Nonlinear Nature of Plans," *IJCAI 4*, 1975.

[18] Subramanyam, S., Lu R., "A Characterization of The Process Planning Task From An Artificial Intelligence Perspective," *19th CIRP International Seminar*, Jan., 1987.

[19] Vernon, I., (Editor), "Realistic Cost Estimating for Manufacturing", SME Series Title: *Manufacturing Data Series, Dearborn, MI*, 1968

## BIOGRAPHICAL SKETCHES

**BEHROKH KHOSHNEVIS** is an Associate Professor of Industrial and Systems Engineering and is the Director of Manufacturing Engineering Program as USC. His primary research interests are in computer aided manufacturing and in computer simulation. His primary research interests are in computer aided manufacturing and in computer simulation. His current research activities are in automated process planning, concurrent engineering, measurement planning, and intelligent simulation environments. Dr. Khoshnevis is a senior member of the Society of Manufacturing Engineers and the Institute of Industrial Engineers, and is a member of the Board of Directors of the Society for Computer Simulation. Dr. Khoshnevis has designed several large-scale software systems that have been used by industries such as airframe and electronics manufacturing. Dr. Khoshnevis has recently designed and developed the EZSIM simulation software. His book on the subject of computer simulation is recently published by McGraw-Hill, Inc.

**JOO PARK** is a Research Engineer at Southwest Research Institute. Prior to joining SRI he worked at the Science Division of the Research Institute of Industrial Science and Technology in Pohang, Korea. He received his BS degree in industrial engineering from Seoul National University, Korea. Dr. Park's MS and PhD degrees in industrial and systems engineering are from the University of Southern California. As a graduate student at USC he was instrumental in design and maintenance of a computer aided manufacturing laboratory. His major research interest are in CAD/CAM related areas.

**DUSAN SORMAZ** is a Research Associate in manufacturing engineering at University of Southern California. He received his BS and MSc degrees in industrial engineering from University of Novi Sad, Yugoslavia and his PhD from USC. His research interests are in CAPP, CAD/CAM integration, and production planning. He is a member of SME and IIE.