A TRUE GENERATIVE CAPP SYSTEM FOR DFM APPLICATION TO MACHINED COMPONENTS

by

Charles Chi-Yu Chiang

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree
of
MASTER OF SCIENCE
in
Industrial and Systems Engineering

APPROVED:

Osama Eyada

Dr. O.K. Eyada, Chairman

Dr. C. F. Reinholtz

R. D. Dryden

Dr. R. D. Dryden

November 1994
Blacksburg, Virginia

A TRUE GENERATIVE CAPP SYSTEM FOR DFM APPLICATION TO MACHINED COMPONENTS

by

Charles Chi-Yu Chiang

Committee Chairman: Osama K. Eyada

Industrial and Systems Engineering

(ABSTRACT)

Today's highly competitive marketplaces require production systems that are flexible and responsive to changing demands. To remain competitive, companies need close coordination and exchange of computer interpretable information between product design and the manufacturing system. Computer-Aided Process Planning (CAPP) is an essential key for achieving closer links among design and manufacturing activities.

The purpose of process planning is to generate feasible sequences for producing a part in a given production facility. To generate process plans automatically (true generative CAPP), design information along with production facility information needs to be appropriately represented. Most CAPP systems assume feasible designs as input and lack the capability to evaluate designs for manufacturability with respect to the production facility. The objective of this research is to develop a true generative CAPP system that is an integral part of a design for manufacturability (DFM) application for machined components. It involves determining appropriate representation schemes of machined components and production facility resources.

The created CAPP Module, developed using C++, consists of five process dependent modules for automatic process plan generation and evaluation: (1) Process selection, (2) Machine/Tool Selection, (3) Setup/Fixture Planning, (4) Operation Sequence Planning, and (5) Process Plan Evaluation. Process plan generation is performed by the first four modules. Evaluation of process plans is performed by the Process Plan Evaluation Module. Criteria such as cost, resource utilization, and production requirement, are used to generate the most appropriate process plan and to select additional process plans as needed.

ACKNOWLEDGMENTS

I wish to thank Dr. Eyada for his guidance and support. He has provided continual patience and persistence that were integral to my accomplishments. I wish to also thank my committee members, Dr. Robert D. Dryden and Dr. Charles F. Reinholtz, for their interest and involvement in my research. Furthermore, I would like to express my gratitude to Krishna Krishnan for paving a solid foundation for my research. Many thanks goes out to my stepfather, Robert F. Britton for reviewing my final document. Also, I would like to thank Annie So for providing me valuable insights. Finally, I would like to thank my mother, my stepfather and my sister for their moral and financial support. I dedicate this thesis to my beloved late grandfather.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Today's highly competitive marketplaces require production systems that are flexible and responsive to changing demands. Consumers expect a greater variety of products to be costing less, having higher quality, and developed in less time. To remain competitive, companies need close coordination and exchange of computer interpretable information between product design and the manufacturing system. Computer-Aided Process Planning (CAPP) is an essential key for achieving closer links among design and manufacturing activities.

Process planning is defined by the Society of Manufacturing Engineers as the systematic determination of the methods by which a product is to be manufactured economically and competitively [DALL76]. The traditional practice of manual process planning requires experienced process planners. Experience requires time to accumulate and often does not apply to new processes. Also, it is not possible to rely on expert process planners alone to keep pace with ever-changing technology and remain competitive in the global marketplace. Because of this and the need for linking Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM), many research efforts have been directed towards automating the manual process planning activities.

To generate process plans automatically, design information from CAD systems, along with required manufacturing attributes needs to be appropriately represented. This research aims at determining some of the appropriate data representation of machined components for automatic process plan generation.

The tasks of process plan generation are described in the next section. A description of the components of a CAPP system and the various CAPP approaches are described in the following sections, followed by the proposed research work.

1.1 Process Plan Generation

Figure 1.1 illustrates the factors involved in process planning according to IDEF-0[1] representation [ELMA93], [COLQ91]. Input information includes both design and production requirements. The process planning activity is controlled by available resources and their processing capabilities. The actual planning is performed either by an expert planner and/or through a CAPP system. The output of the process planning activity is the process plan, detailing the information needed for downstream production activities. A typical process plan is composed of two parts: a route sheet and operation sheets; each part providing a different level of detailed information. A route sheet provides a sequence of operations to be performed on the part. Operation sheets provide specific information about the tool and machining parameters to be used at each operation.

Process planning is a design process in which iterative analyses are performed to produce a cost efficient process plan. The following non-sequential tasks are involved in process plan generation for machined components:

---

[1] IDEF stands for I-CAM DEFinition. It is a structured systems analysis methodology developed by the US Air Force. Refer to references for more information.

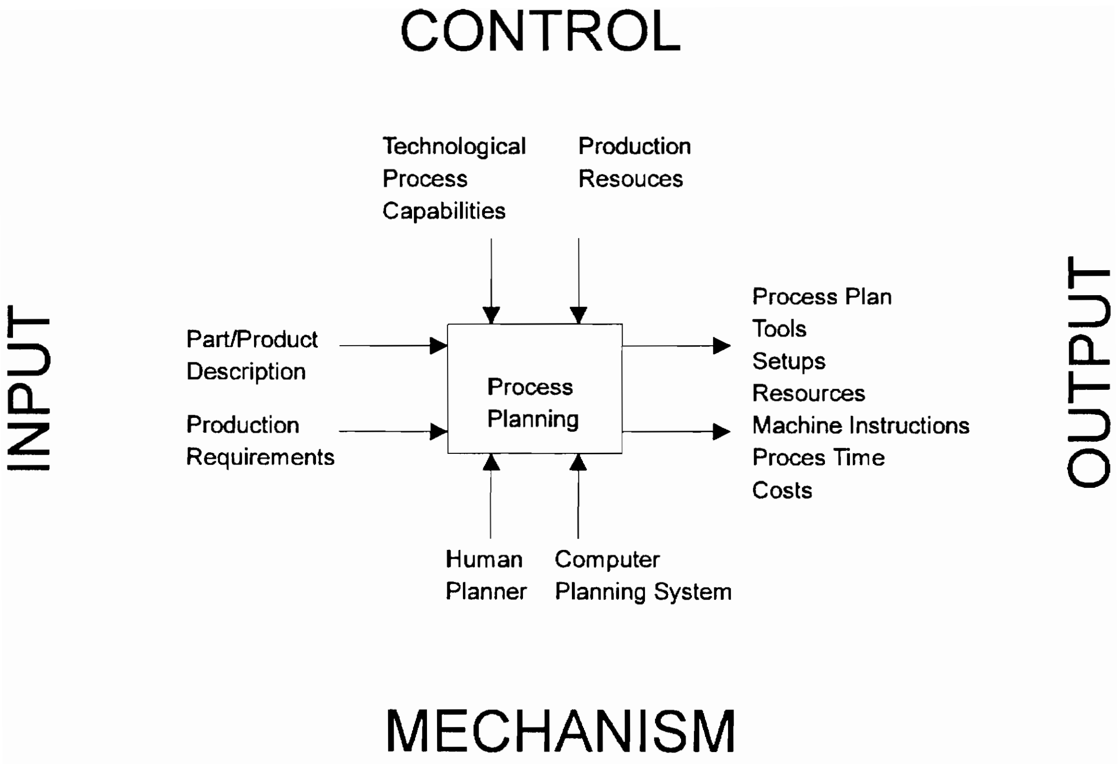


Figure 1.1. IDEF-0 of Process Planning Activity

- design interpretation and analysis
- selection of machining processes
- sequencing of machining surfaces
- determining operations and their setup requirements
- calculation/selection of machining parameters

Design Interpretation and Analysis

Design interpretation and analysis is the first task to be performed in process planning. The objective at this stage is to carefully consider both design and production requirements. This type of analysis is commonly known as Design For Manufacture and Assembly (DFMA). Design information such as material, dimensions, tolerances, and geometric configuration are interpreted and analyzed with respect to manufacture and assembly for their feasibility and economics. Many decisions to be made in later stages of process planning are highly dependent on the results of this stage. For example, if the tolerance of a dimension can be relaxed without affecting the part's functional requirements, fewer setups may be required, thus lowering manufacturing costs.

Selection Of Machining Processes

Machining processes are selected by matching the part manufacturing features, determined from the part design, with the processing capabilities. Most designs have multiple features and/or compound features. In many cases, there is more than one process that is capable of producing a certain surface or feature. Machining process selection also depends on production requirements (e.g., availability of equipment).

Sequencing Of Machining Surfaces

Machining surfaces are sequenced based on design requirements, including dimensions, tolerances, and material specification. Certain surfaces, for example, need to be generated before others to achieve the required tolerance. In some cases, there may be many different possible machining sequences for creating a surface or a feature. In cases where there is more than one possible sequence, each processing sequence represents an alternative process plan. Another method of creating alternative process plans is to use the same processing sequence, but change the operation and setup requirement.

Determining Operation And Setup Requirements

A machining operation may involve the generation (machining) of more than one surface on a workpiece using one or more cutting tools on a single setup. A change in part orientation on the same setup constitutes a new machining . A setup requirement specifies the method of part fastening, type of machine are types of tools to be used. The part is fastened to the equipment using standard (e.g., a vise) or custom designed (i.e., a fixture or jig) holding devices. The machine equipment and tooling for each operation are selected based on the surface(s) being created, material properties of the part, and the production volume. Equipment selection specifies the type of machine to be used to meet the desired production capacity. Tooling selection specifies tool type, tool size, and tool material.

Criteria used for determining the best operation and setup requirement are: (1) minimizing cost and (2) minimizing time. When trying to minimize cost, the machining sequence requiring the fewest setups is used. When trying to minimize time, mass production concepts (more setups) are used; machines, part holders, and tools with the most appropriate production rate are selected.

Calculation/Selection Of Machining Parameters

For each operation, machining parameters on the operation sheets specify the machine feed, speed, and the tool depth. Machining parameters are dependent on the type of cut being performed as well as tool and part material. Rough cuts can have higher feed and depth of cut than final cuts. Machining data can be determined through machine databases or calculated through mathematical models or empirical equations. Machine parameters from machining databases are compiled over many years by experienced machinists. One drawback to using compiled machining parameters is that it does not provide the best result. Another method of determining machining parameters is from mathematical models (optimizing least cost) or empirical models such as Taylor's tool-life equation.

Table 1.1 shows an example of a route sheet for the part shown in Figure 1.2a, where Figure 1.2b identifies the part surfaces. The operation sheets are shown on Tables 1.2, 1.3, and 1.4. The process plan is created based on the production requirement of one million units per year. For this particular example, all the part surfaces will be generated from raw stock. For other parts which are cast first, only certain surfaces need to be created.

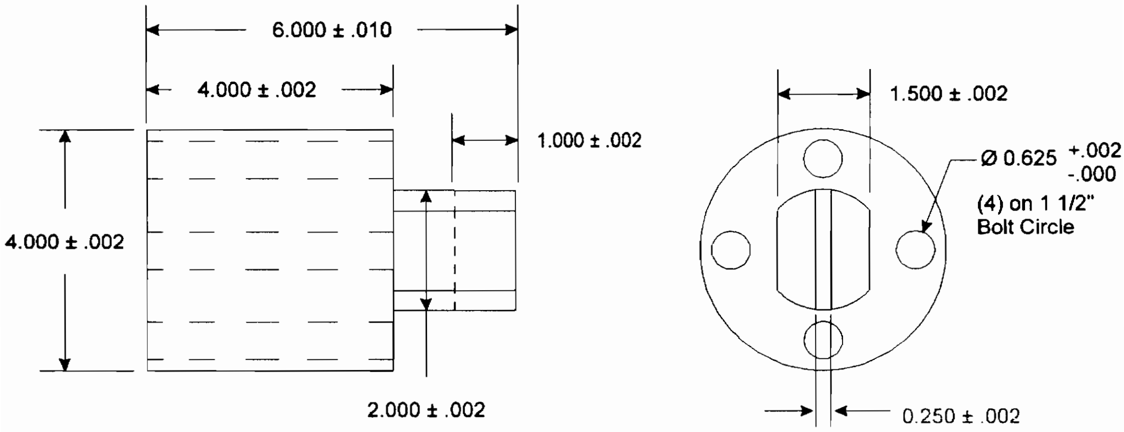


Figure 1.2a. Sample Part Drawing

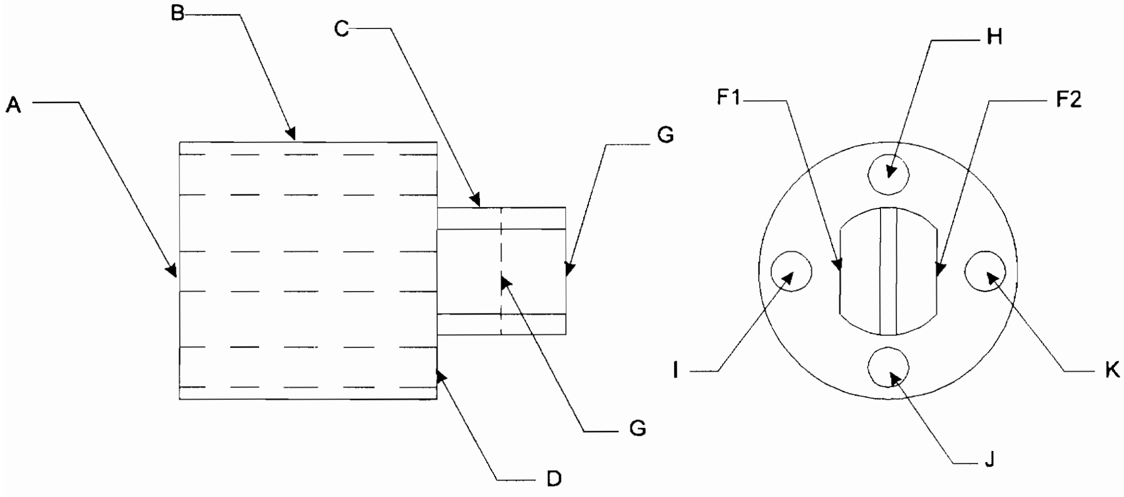


Figure 1.2b. Sample Part Surface Identification

Table 1.1. Route Sheet

Number	Description	Equipment	Jig/Fixture
10	Automatic Bar Stock Stop	Turret Lathe	Pneumatic 3-Jaw Chuck
20	Face mill F1, F2 While end milling G	Triplex Bed type milling machine	Circular milling fixture
30	Drill H, I, J, K (simultaneously)	Multi-Spindle Drilling Machine	Pneumatic 3-Jaw fixture

Table 1.2. Operation 10 Sheet

OPERATION 10					
SURFACE	TOOL TYPE	TOOL MATERIAL	v	f	d
A	Facing Tool	WC	350	0.02	0
B	Straight Turning Tool	WC	350	0.02	<.125
C	2 1/8 Flat forming tool	WC	200	0	1
D	Facing Tool	WC	350	0.02	0
E	1/8" Cut-Off Tool	WC	200	0	1

Table 1.3. Operation 20 Sheet

OPERATION 20					
SURFACE	TOOL TYPE	TOOL MATERIAL	v	f	d
F1, F2	Facing Milling Cutter D=2"	WC	175	0.01	0.25
G	End Milling Cutter D= 1/4"	WC	175	0.01	1

Table 1.4. Operation 30 Sheet

OPERATION 3O					
SURFACE	TOOL TYPE	TOOL MATERIAL	v	f	d
H,I,J,K	Drill D=.625	HSS	40	0.01	4

1.2 Computer Aided Process Planning

Originally, CAPP started as an application for assisting human process planners in retrieving existing process plans. Current developments are focusing on using computers to automate the entire process planning function. There are two basic approaches to computer-aided process planning: variant and generative.

1.2.1 Variant Process Planning

The variant process planning approach is also referred to as a data retrieval method [WANG91]. This approach is based on the concept of Group Technology (GT) to take advantage of similarities among parts by grouping them into families. In a variant system, a classification and coding system needs to be developed to represent certain characteristics or attributes of the part. A standard plan is assigned to each family. The computer serves as a file manager for identifying and retrieving existing process planning information. The retrieved standard plan will need to be edited for each different part. For example, when a new part needs a process plan, the part design attributes are translated into a GT code. The part code is entered into the variant CAPP system. The system will then search and retrieve the standard process plan of the common part family that closely relates to the part's attributes.

There are many problems associated with the variant approach. One major problem is the high front-end cost. When implementing variant CAPP systems, GT codes need to be developed to represent the part families. In addition, standard plans need to be determined for each part family. Another problem is that GT codes are insufficient in representing detailed design specifications; for example: exact size, tolerance and surface finish. Complete automation of the process planning function is not possible with the variant approach [CHAN90]. Because of this, variant CAPP systems also have very high operating costs. Standard process plans that are retrieved for a part still need expert process planners to perform certain tasks. Standard process plans often need to be modified and supported with detailed instructions.

Since the focus of this research is on generative process planning, variant approaches will not be discussed in later sections. More information on the various CAPP variant approaches can be found in the following references: [CHAN90] and [CHAN91].

1.2.2 Generative Process Planning

In contrast to variant CAPP systems, process plans are automatically generated by generative CAPP systems. Generative CAPP systems have the logic to perform high level design interpretation and decision making functions. Figure 1.3 illustrates the main elements that make up a typical generative CAPP system. These elements are: (1) Design Input, (2) Decision Logic, (3) Knowledge Base, and (4) Manufacturing Resource.

Design Input

Design input is the interface to a CAPP system. The main function of this module is to classify and identify the semantics of design information. Surfaces of a part need to be identified and classified according to the need for machining. Once this task has been completed, other components of the CAPP system can

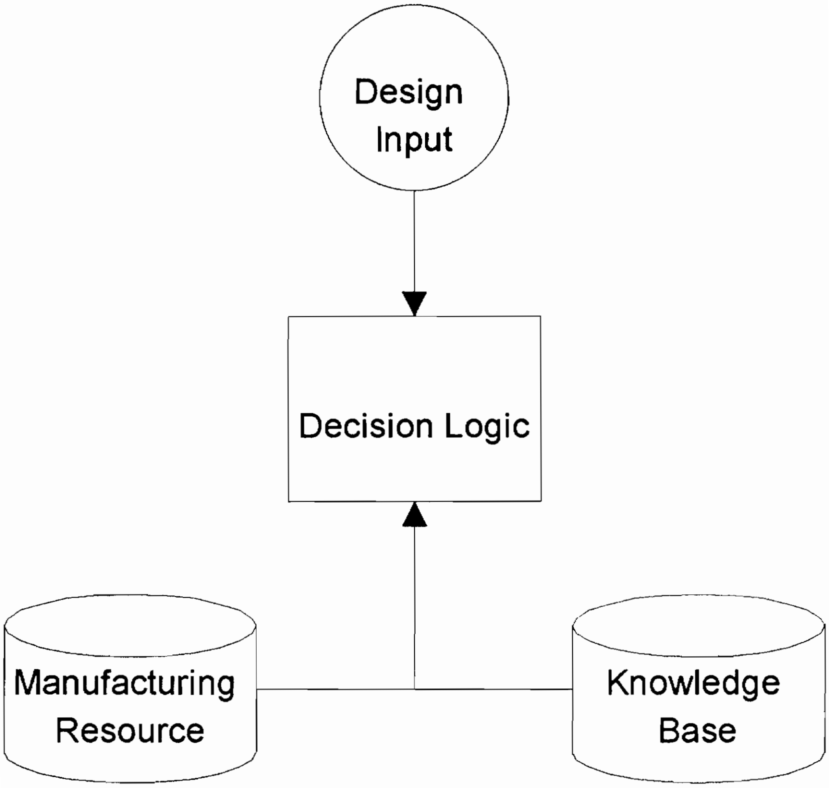


Figure 1.3. Components Of CAPP Systems.

collaborate to generate the necessary sequence of operations of a process plan. Design input can be represented in many formats. However, for an ideal true generative environment, the interface should come directly from a CAD database. Furthermore, the part design should be analyzed for DFMA. The majority of generative CAPP systems that act on CAD databases assume that the part is already designed for manufacture and assembly.

Decision Logic

Decision Logic, representing the decision making process of process planning, attempts to match certain attributes between the design input and the knowledge database (representing manufacturing information). Each time a match is made between an input attribute and a knowledge-base attribute, the decision logic will interact with manufacturing resources to assign specific machines, fixtures, tools, needed to create the feature.

Knowledge Base

A knowledge base represents facts, rules, and heuristics about the Design Input. It is represented by production rules such as if-then statements. The knowledge base is usually a reflection of the underlying Design Input.

Manufacturing Resources

In every factory, there is a collection of machines, tools, fixtures and materials. The Manufacturing Resource acts as a manager of the factory resources; operations are assigned to machines after verifying that capabilities of the machines meet the necessary design requirements.

1.3 Approaches To Generative Process Planning

Approaches to generative process planning can be classified into the following two categories: semi-generative and true generative. The major distinction between semi-generative and true generative CAPP systems is the design input. In a semi-generative CAPP system, design information needs to be first manually converted into the specific format of the CAPP system. However, the design input for a true generative CAPP system comes directly from a CAD database.

1.3.1 Semi Generative Approach

The types of input formats for a semi generative CAPP system are: (1) GT code, (2) descriptive language, and (3) custom graphics interface. GT codes for semi-generative CAPP systems have lengthy part codes for higher resolution (e.g., M-GEPPS [CHAN91]). Descriptive language describes part designs by key words and their attributes (e.g., XTURN [HERM93]). In a custom graphics interface, the part design is redrawn or reentered into the CAPP system format interactively along with the necessary technical information (e.g., TIPPS [CHAN82]).

Once a semi generative system receives the part design, a new process plan is automatically generated. In most cases, semi generative CAPP systems perform most of the decision making functions of a process planner. However, input files of semi generative systems lack complete geometric information. The output of these systems is less capable of providing information needed by downstream activities such as assembly planning and numerical control (NC) code generation.

1.3.2 True Generative Approach

A true generative CAPP system uses CAD data models as design input. It has the potential for eliminating the need for human involvement in process planning and can automate downstream activities such as cutter path generation and production scheduling. Figure 1.4 shows an ideal true generative CAPP system which should be able to analyze the design input with respect to the production facility, provide feedback to the designer(s), and perform the process planning activity automatically. Furthermore, the CAPP system should be an integral part of a computer integrated environment, i.e., computer-integrated manufacturing (CIM).

The two types of CAD data models used for input are: conventional CAD model and feature-based CAD model. Conventional CAD models, such as wireframe and solid models, represent a designed part in terms of geometric information, such as vertices and edges. However, the wireframe model contains only geometric information, whereas the solid model contains both geometric and topological information. A common wireframe standard is the Initial Graphics and Exchange Specification (IGES) or the data exchange file (DXF). Boundary representation (B-rep) and constructive solid geometry (CSG) are common standards of solid models.

Feature-based CAD models represent a part design by embedding higher levels of abstract attribute information within the geometric data. Parts are modeled using features that link non-geometric attributes to geometric features, such as creating a slot with dimension, tolerance, location and orientation. No standard format or definition currently exists for feature-based models. However, most feature-based CAD packages use solid model geometric representations

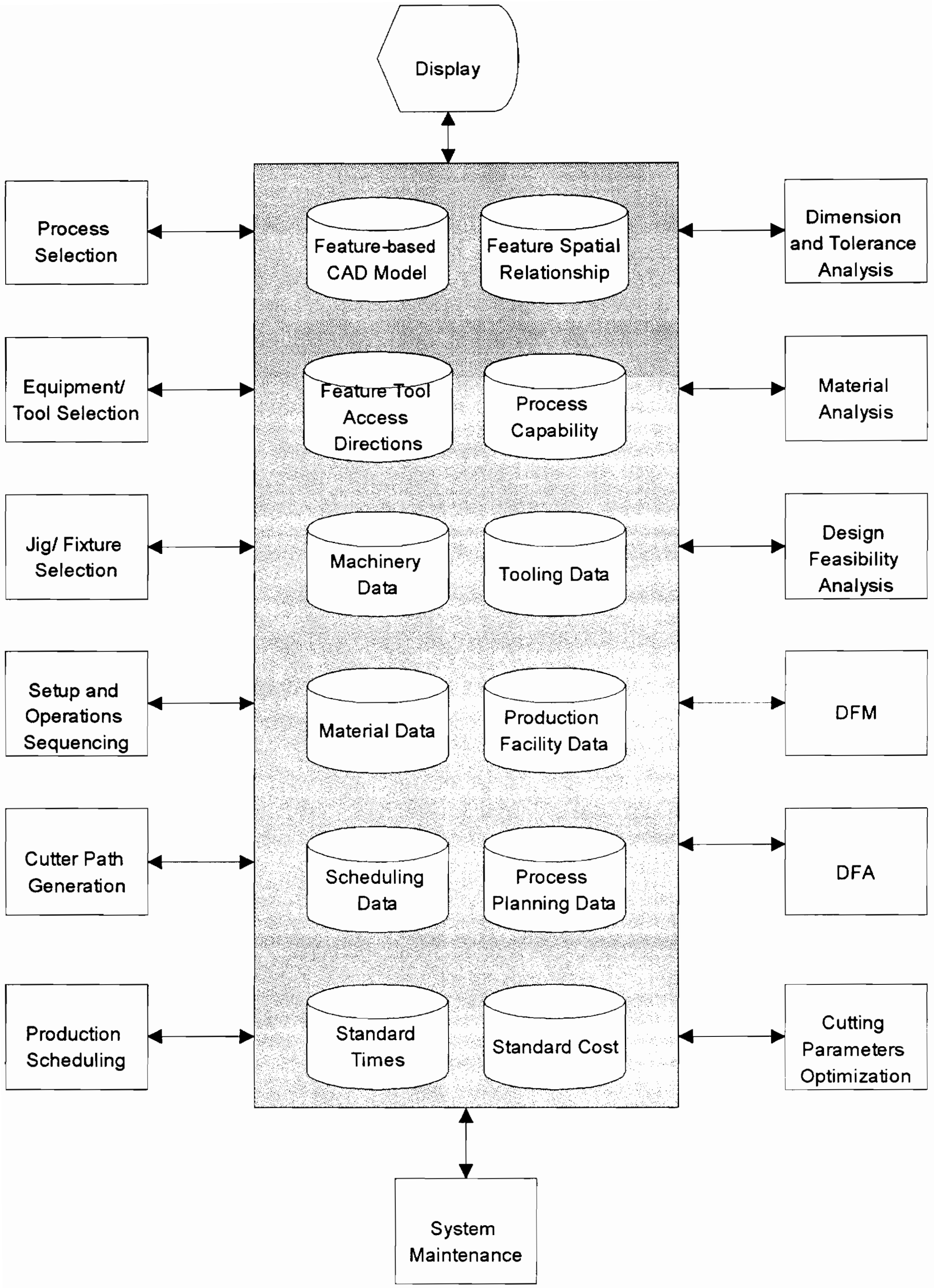


Figure 1.4. An Ideal CAPP System For Machined Components

based on a CSG model, a B-rep model, or a dual CSG/B-rep model. In addition to geometric information, feature-based data models can contain tolerance, dimensional, and functional information.

CAPP systems using conventional CAD data as input must rely on feature recognition techniques to obtain necessary process planning information. Feature-based CAD models, on the other hand, can explicitly incorporate process planning information. Feature recognition techniques are pattern recognition algorithms that search for pre-defined feature information, such as machining features. Information is extracted from the geometry and the topology of the CAD data model [WOO82]. However, feature recognition is dependent on the CAD data model and is unsuitable for recognizing complex features and feature interactions. In addition, most CAD systems represent tolerances as texts. In other cases, feature recognition will break down if there is no starting point for the search routine. Feature recognition programs become limited due to the possibility of incompleteness or excessive computational load [ISHI92].

The need for including higher level process planning information has helped motivate the recent development in feature-based CAD systems. A CAPP system can extract explicit manufacturing feature information and tolerancing information from the part description file without depending on complex feature recognition algorithms. Integration of different applications within a larger framework is possible with feature-based models. It is worth noting that the definition of features is application dependent. Without a standard feature-based model definition, there is a lack of generality among different feature-based CAD systems [QIAO93]. International efforts to define a standard for product model data known as Standard for Transfer and Exchange of Product data model (STEP) along with related application protocols will facilitate development of applications such as CAPP. The objective of STEP is to provide a neutral mechanism capable of describing product data throughout its life cycle. Continuing developments in STEP will ultimately provide more defined product and feature models.

In conclusion, feature-based CAD systems will invariably become the more prevalent approach for achieving true generative CAPP and other applications. An ideal CAPP system should have the following features:

- Be an integral part of a larger system that performs design evaluation.
- Adequately represent data necessary for process planning.
- Adequately represent manufacturing resources and their status.
- Generate alternative process plans when needed to meet production schedule.

1.4 Problem Statement

The purpose of process planning is to generate feasible sequences for producing a part in a given production facility. Up to this point, most developments in true generative CAPP systems have been focused on automating design interface. However, a true generative CAPP system should be part of a complete computer integrated environment [ALTI89]. At the conceptual stage of design, the manufacturing feasibility of the part should be performed. Later in the design stage, process planning sequence of the nearly finalized part should also be analyzed for its manufacturability.

A Literature survey of CAPP systems shows that feature-based design models are the most promising approach for achieving true generative process planning. However, some systems still lack adequate process planning information such as tolerance [CHEN94b], and feature interactions [LAKK90], [DELB93], [TONS94]. Furthermore, most feature-based CAPP systems assume feasible designs as input and lack the capability to evaluate the design for its manufacturability, [LAKK90], [CHAN90], [DELB93], [CHEN94b], [TONS94].

1.5 Research Objective

The primary objective of this research is to develop a true generative process planning system to be an integral part of a DFM framework. The scope of the research focuses on metal removal processes. The developed CAPP system generates alternative process plans while considering production facility information. A representation scheme for manufacturing resources for process plan generation has been developed. This also facilitates the evaluation of the process plan based on the selection of resources.

1.6 Document Outline

In chapter 2, a literature review of relevant material is provided. Chapter 3 details the development of a true generative CAPP system that will be an integral module of a DFM framework. It also describes the data representation structure for representing key manufacturing resources of the production facility. Chapter 4 demonstrates the proposed methodology with a case study and software implementation. Details of the software implementation are also described. Chapter 5 concludes with the contributions of this research and discusses future research.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews literature on generative process planning. The majority of the published work can be classified into two categories: (1) semi generative and (2) true generative. In a semi generative CAPP system, the part file or drawing is first translated into the application dependent format for process plan generation. A true generative CAPP system, on the other hand, automatically performs reasoning on part files in a CAD model for process plan generation. Section 2.1 discusses the various types of input format used by semi generative CAPP systems. Section 2.2 discusses the various types of CAD model input format used by true generative CAPP.

2.1 Semi Generative Process Planning

The majority of semi generative CAPP systems can be classified based on the input file format: GT code, descriptive language, and custom graphics language.

2.1.1 GT Code

GT is based on the concept of taking advantage of similarities. GT code consists of a sequence of symbols that describe a part's design characteristics and/or manufacturing attributes. Codes used in semi generative process planning tend to have a higher degree of resolution than those used in variant process planning. The degree of resolution depends on the number and type of digits used.

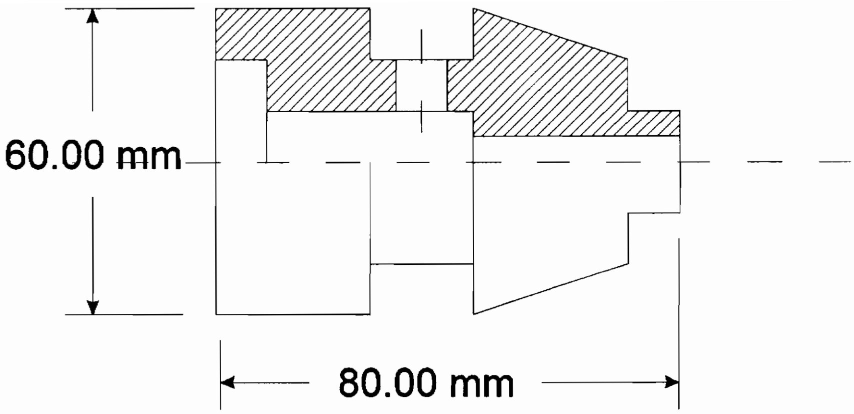
Several commercial coding schemes have been developed; such as OPITZ, DCLASS, KK3 [CHAN91]. Most semi generative CAPP systems use these commercial coding schemes. Examples of semi generative CAPP systems are: APPAS, GENPLAN, COBAPP, M-GEPPS, and CORE-CAPP [CHAN90].

Using a semi generative CAPP system requires manual entry of the part design. For example, a sample part being planned using M-GEPPS along with its part code is shown in Figure 2.1. M-GEPPS is a semi-generative CAPP system used for turned part process planning [CHAN91]. It uses the 21 digit KK-3[2] coding system as the input. The part code entered into the CAPP system for this example would be: 097222300000200030004. The system would then prompt the user for header data: part name, part number, work material, drawing number, cost, process planner, etc.. Additional attributes such as length, diameter and tolerance of the raw part as well as the finished part are also prompted by the system.

In the above example, the part coding was a manual task. Manual coding requires thorough understanding of the coding system and is a tedious and time consuming process. To overcome this drawback, automated coding systems have been developed. Such systems utilize feature recognition techniques to classify and organize part features from B-Rep, CSG, or feature-based CAD data structure [NADI93], [LIAO94], [NAGA94]. These automated coding systems have yet to be successfully applied to commercial CAPP systems.

---

[2] See Chang et al. [CHAN91] for complete KK-3 coding structure, definition and vocabulary



Code Digit	Item	Component Condition	Code
1	} Name	Control Valve	0
2			9
3	} Material	Copper Bar	7
4			2
5	Dimensional length	Greater than 50 mm less than 100 mm (80 mm)	2
6	Dimensional Diameter	Greater than 50mm less than 100 mm (60 mm)	2
7	Primary Shape and ratio of chief Dimension	L/D 1.3	2
8	External surface	With functional tapered surface	3
9	Concentric Screw	None	0
10	Functional cutoff	None	0
11	Extraordinary shaped	None	0
12	Forming	None	0
13	Cylindrical Surface	None	0
14	Internal Primary	Piercing hole with dia. variation, No cutoff	2
15	Internal Curved Surface	None	0
16	Internal flat surface	None	0
17	End surface	Flat	0
18	Regularly located hole	Holes located on circumferential line	3
19	Special hole	None	0
20	Noncutting process	None	0
21	Accuracy	Grinding process on external surface	4

Figure 2.1. Coding A Component Using KK-3 Coding System

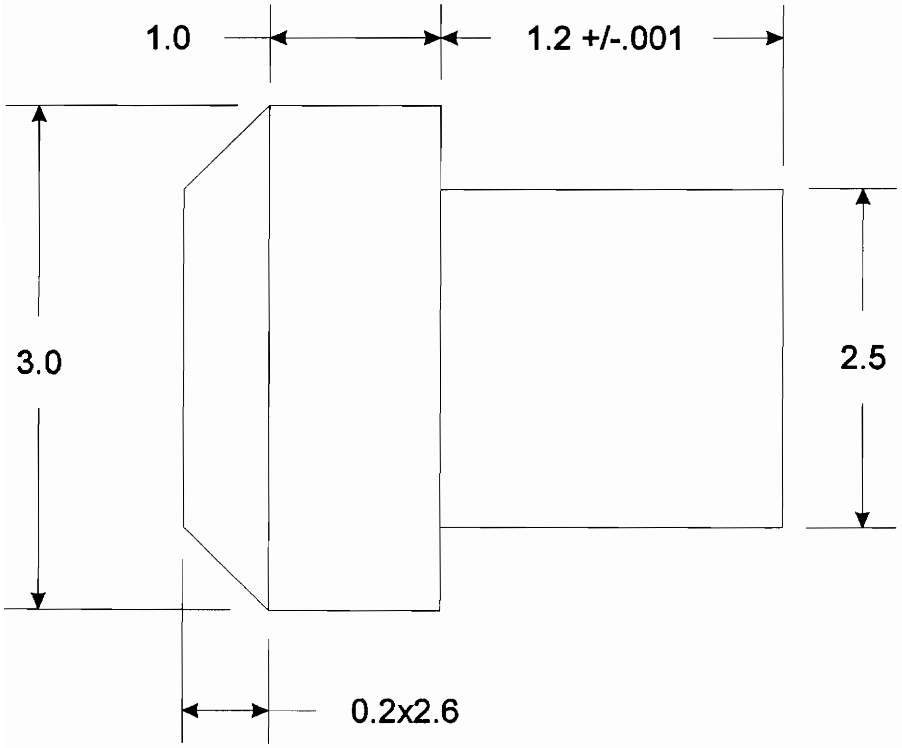
Using GT codes as input to CAPP systems is unsuitable for true generative process planning. Codes are insufficient in representing detailed design specifications such as exact size, tolerance, and surface finish. For this reason, the above example of M-GEPPS requires user input for the 80 mm diameter and the 60 mm length.

2.1.2 Descriptive Languages

Another method of describing part design is by descriptive language. Descriptive languages can be specially designed to provide all the necessary detail needed for process planning. A process planner models a component by entering the description in the application specific format. Each description is composed of a key word and its attribute. Examples of semi generative CAPP systems using the descriptive language format as inputs are: AUTAP, CIMS/PRO, GARI, AUTOTECH [CHAN91], and XTURN [HERM93].

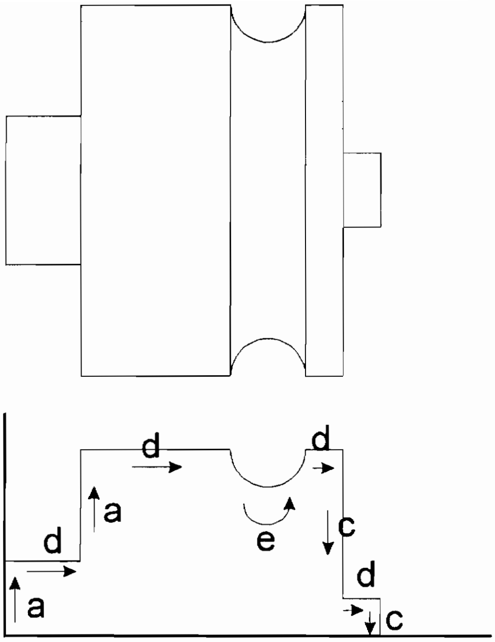
A descriptive language consists of geometric and technological information. Figure 2.2 illustrates a sample part modeled by part description of the AUTAP system [CHAN90].. Geometric elements, such as cylinders and blocks, are represented by key words and attributes. Detailed part features are represented by subordinate elements such as chamfer and holes. The AUTAP system performs process planning through translating component description commands to corresponding manufacturing actions.

Another example of a part description using descriptive language is shown on Figure 2.3. The part is described by the concatenation of feature primitives, represented by a string. Each feature primitive is a contour description, denoted by a character; "a", "b", "c", "d", or "e". The representation scheme for this example is ideal for rotational parts. The part description in this



```
10  CYLINDER/3,1/
11  DFIT/K,5/
12  CHAMFER /.2,2.6/
20  CYLINDER /2.5, 1.2/
21  LTOL/+0.001,-0.001/
```

Figure 2.2. AUTAP Part Description



Part Description: adadedcdc

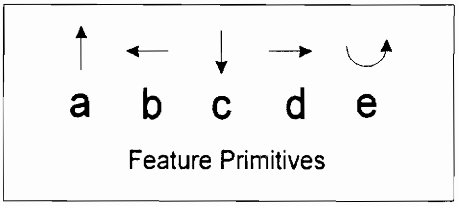


Figure 2.3. Part Description Using Descriptive Language

example is: adadedcdc. Additional information such as length/radius and tolerance also need to be entered for each feature primitive.

Although descriptive languages can be designed to provide complete information needed for process planning, part description must be entered manually. Modeling complex components may be difficult. This process is irreversible. Once the part file is translated, it cannot be regenerated from description. This is usually required to develop process pictures for schematic illustrations of the part transitional processing stages, to be used on operation sheets.

2.1.3 Custom Graphics Interface

Custom graphics interfaces, like GT codes and descriptive languages, require manual entry of the part description information. Entry of part description, unlike GT codes and descriptive languages, is entered interactively on the CAPP system through a custom graphics interface. Users use visual tools on the CAPP interface to redraw the part and enter technical information.

TIPPS [CHAN82] is probably the first semi generative CAPP system developed using custom graphics interface. Input to the system is a CAD boundary representation data. Users identify surfaces to be machined through a graphics terminal. Using a custom graphics interface saves the user time and effort in converting part information into GT codes or descriptive languages. However, it still requires the user to manually redraw the part or drawing. Manual entry of part information is an inefficient process and increases the chances of human errors. This process can be automated by using CAD models as input to the CAPP system for true generative process planning.

2.2 True Generative Process Planning

A true generative CAPP system reasons directly from CAD models. This type of CAPP system removes the need for any human involvement in the process planning loop. Generative CAPP systems can be classified based on the type of CAD model used to extract manufacturing features: (1) Conventional CAD models and (2) feature-based CAD models. Section 2.2.1 describes conventional CAD model such as wireframe and solid model. Section 2.2.2 describes feature-based CAD models.

2.2.1 Conventional CAD Models

Conventional CAD models contain low level geometric information. CAPP systems, along with other applications such as DFM analysis, need higher level feature information. Because of this, feature recognition has been the approach for extracting feature information from conventional CAD models. True generative CAPP systems must rely on user feature recognition or automatic feature recognition to infer feature information.

User feature recognition can be performed by the user interactively. The part design is displayed on the screen. Certain entities, such as faces, that represent a manufacturing feature are selected and defined. The manual feature recognition approach is used in ICAPP, a true generative CAPP system that uses a wireframe model as input [SSEM93].

Park and Khoshnevis [PARK93] created RTCAPP, which also relies on interactive feature identification to provide data input. FBAPP [DONG94] is a true generative CAPP system created by Dong and Parsaei, that graphically compares the part, designed in solid model, with a blank to obtain an overall removal volume. The overall removal volume is then converted to manufacturing specific information needed for process planning.

Although the interactive feature recognition process is feasible for simple features, multiple compound features represent a more difficult task. User feature recognition is tedious and time consuming. This process has to be performed for each design and is prone to human errors. User feature recognition is often the approach used to define unrecognized features after performing automatic feature recognition.

Automatic feature recognition is performed by feature recognition algorithms. Input to a feature recognition algorithm is a geometric model of the part. In true generative CAPP systems, the geometric model is the CAD file. The function of the feature recognition algorithm is to search the geometric model for generic features based on common patterns. Once a feature has been found, its parameters, i.e., location, diameter, and depth, are computed from the geometric information. Feature information of the part is arranged in a feature graph for further processing. CAPP systems can use the feature graph as input to perform process planning functions.

The following section describes various approaches to feature recognition based on the type of CAD models. Conventional CAD model can be classified into: (1) wireframe models and (2) solid models.

2.2.1.1 Wireframe Models

Two commonly used standard wireframe formats are: initial graphics exchange specification (IGES) and Autocad data exchange file (DXF). IGES is a wireframe model format that is independent of the CAD modeler. DXF is a popular file format of the Autocad modeler.

Wireframe models are the most popular and simplistic CAD representation schemes. Wireframe models contain only edge, vertex, and curve information. Topology information, such as surfaces and their adjacency relationship, is not represented. Because of this, wireframe models suffer from the problem of creating ambiguous objects. Creating complex wireframe objects may be confusing to the eye and difficult to draw. Topology information is needed to resolve problems of ambiguity.

Li and Bedworth [LI88] used syntactic pattern matching to detect rotational features from IGES file format. Syntactic pattern matching uses a sequence of geometric elements to describe 2-D patterns. A sequence is described by a string of codes; each code represents a particular geometric element. The string is then searched for patterns that may represent particular features. A similar approach has also been taken by Swift et al. [SWIF94] to extract profile features of turned part in the AutoCAD DXF format. However, syntactic pattern matching algorithm perform only in 2-D because of the sequencing problems of geometric elements in 3-D. This is a major drawback due to information loss and/or limited application to simple 2-D parts.

2.2.1.2 Solid Models

Solid models define the complete geometry and topology of a part. It is possible to compute the volumetric properties of an object. Because solid models allow for complete and unambiguous representation of 3-D objects, they can be used directly by many engineering applications. The two most commonly used solid model representation schemes are: boundary representation (B-rep) and constructive solid geometry (CSG).

Boundary Representation

B-rep models, contain face, edge, and vertices; topology information is part of the tree data structure (Figure 2.4). A B-rep object is represented by its bounding faces. Each face points to a vector (containing surface information) pointing to the inside of the solid. Geometric information is explicit in the B-rep model. Feature information, however, needs to be extracted through evaluation of the data structure.

Feature recognition techniques for boundary representation can be categorized into: (1) pattern matching and (2) volume decomposition. Pattern matching techniques vary in the method of pattern description [BRON93]. These methods can be classified as: syntactic, rule-based, and graph-based.

The method of syntactic pattern matching was used by Choi et al. [CHOI84] to recognize machining features in B-rep solid models. Sequences of geometric elements are used to describe geometric patterns. Each element is represented by a code. A feature is recognized when a sequence of code from the model matches a geometric pattern.

Rule-based pattern matching uses production rules. Henderson and Anderson [HEND84] converts B-rep design models into facts and compares them to rules defining a particular type of feature. This system is capable of recognizing simple 3-D features. Their feature recognition algorithm was written in Prolog.

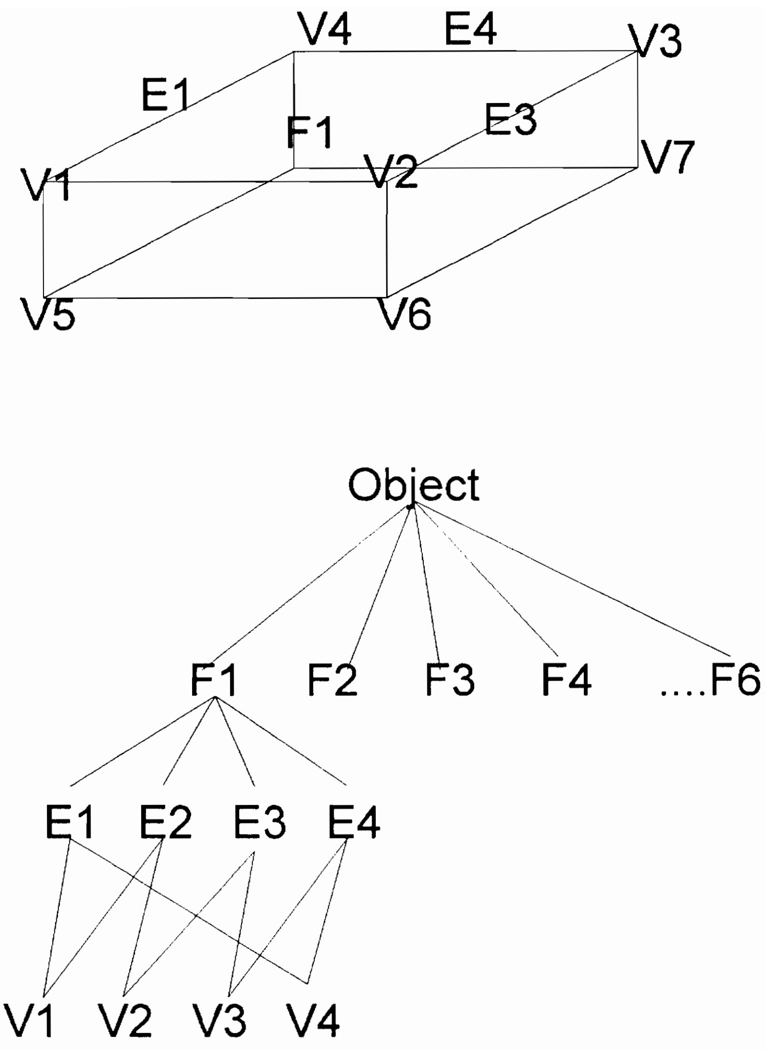


Figure 2.4. B-Rep Solid Model Data Structure

Graph-based pattern matching utilizes B-rep's explicit geometric data structure for inferring feature information. Graphs representing topological relationships among geometric entities are analyzed for feature recognition. The nodes of a graph structure represent the boundary elements. And the links represent adjacency relationships. Topological models can be based on edge-vertex graphs or edge-face graphs. Other graph structures are possible.

Joshi and Chang [JOSH88] uses a face-edge graph with attributes for feature recognition. Faces of the object are represented by nodes. Edges of the object are represented by links. Each link connects two adjacent faces. Attributes, indicating a convex or concave relationship, are attached to the links. This is determined by the angle formed by the two adjacent edges. Features of interest are defined by describing their face-edge relationship in the graph form. The part model must also be described in the graph form. Sub-graphs of the model are then searched for matching features. Sakurai and Gossard [SAKU88] added more attributes to nodes and links to include more features. Chuang and Henderson [CHUA90] uses a vertex-edge graph structure for feature recognition.

The Volume decomposition approach to feature recognition attempts to derive material removal volume and decompose it into sub volumes. These sub volumes correspond to machining operations. T. C. Woo [WOO82] introduced the decreasing convex hull algorithm to decompose machining volumes from the part. The difference between the object and its convex hull is computed recursively, until the object equal its convex hull. The object could then be represented as a sequence of convex volumes associated with difference operators, with each corresponding to a machining operation. In some cases, the removal volume may not represent odd shapes. Problems can result if the algorithm does not converge for certain types of part geometry [BRON93].

Constructive Solid Geometry

CSG models are constructed by combining different solid primitives, such as cuboids, cylinders, spheres, and tori. Primitives are combined by the application of boolean operators, such as union, difference, and intersection. The user creates a CSG object by entering the size, location, and orientation of the primitives or boolean operations.

The CSG model is represented by a binary tree structure (Figure 2.5). Each terminal node represents a primitive or a compound primitive. Each non-terminal node represents a boolean operator. The data structure of CSG models provides explicit feature information. The binary tree structure captures both design histories and operating constraints. Geometric information, however, is implied. A major drawback of the CSG model is its non-uniqueness, i.e., one object can be represented by several CSG models [BRON93]. Figure 2.5 also shows an alternative binary tree representation of the same part.

Lee and Fu [LEE87] suggested a method for rearranging the CSG model for resolving the non-uniqueness. The proposed method consists of a three phase procedure: (1) extraction, (2) rearrangement and (3) unification. In the extraction phase, the CSG tree is traversed. The orientation, size, and principle axes of each feature are extracted. In the rearrangement phase, primitives are clustered based on spatial relationships. Each cluster, representing a feature, is moved into the same sub-tree. In the unification phase, feature subtrees are replaced by others that are known to be equivalent but contain more material

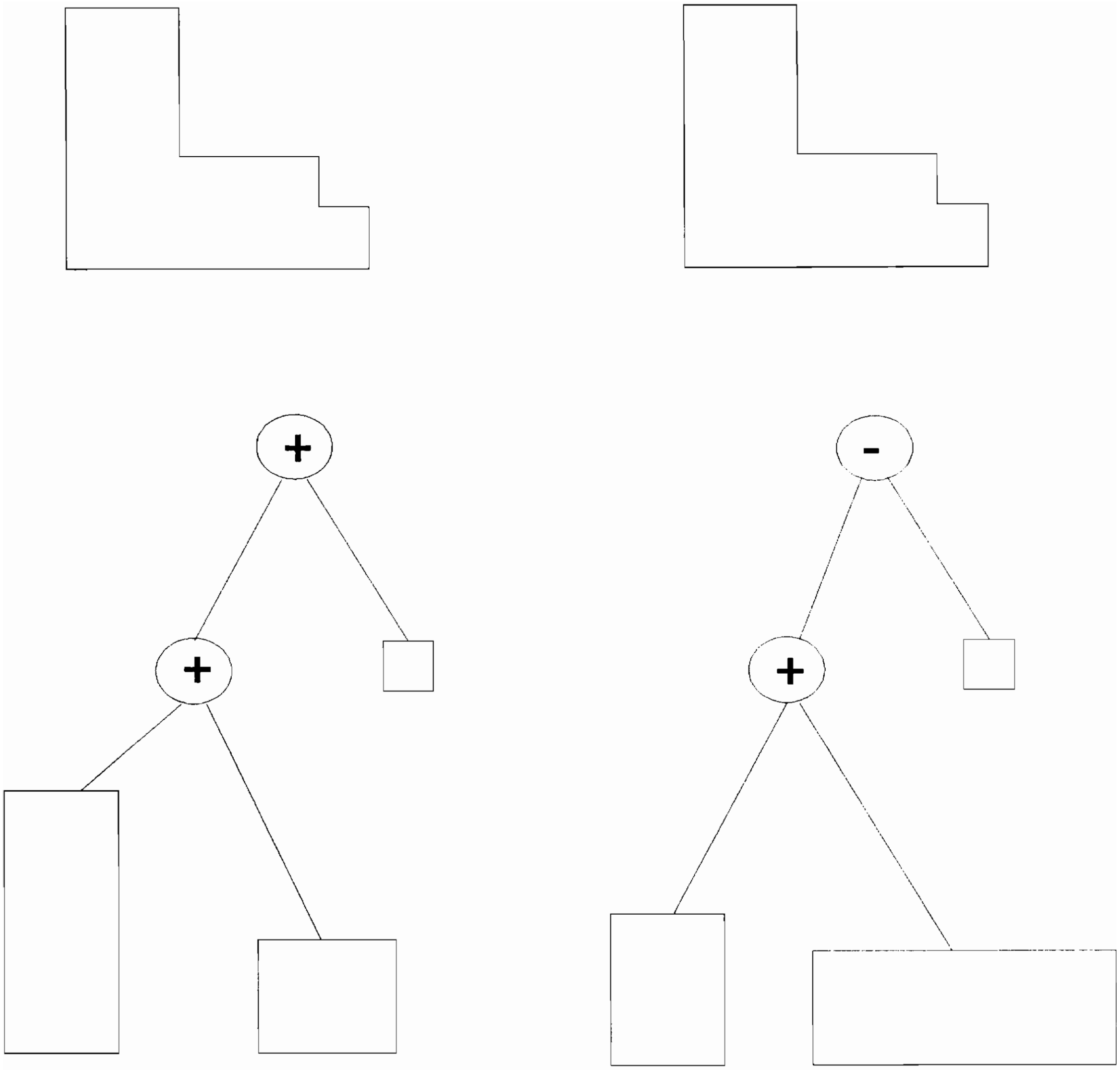


Figure 2.5. Two CSG Representations Of An Identical Part

and difference operations. Using only the difference operation may be desirable because it corresponds to manufacturing machining processes.

Perng et al. [PERN89] developed a procedure for transforming the CSG tree to a destructive solid geometry tree. The resulting tree resembles a machining sequence starting from a raw stock. Features are successively subtracted at each node (using the difference operator).

2.2.1.3 Neutral File Format

Not all CAD file formats contain the same geometric information. Because of this, approaches to feature recognition are different. R. V. Auradkar [AURA92], suggested converting different CAD models into a single neutral file format. The neutral file would provide a consistent format for feature recognition algorithms and be independent of CAD models. This neutral file concept was demonstrated by converting a DXF wireframe model to a feature-based neutral file for feature extraction.

2.2.2 Feature-Based CAD Models

From the process planning point of view, manufacturing features are the key to process plan generation. Because conventional CAD models contain only low level geometric information, feature recognition techniques are needed to extract feature information. One major drawback to feature recognition is the complexity and limitations of the process. Features that can be extracted are limited to a small number of form features and the definition of features is not precise. In addition, feature recognition is performed at a high cost since this information has already been generated during the design process. This information is lost when the result of the design process is stored into pure geometric form [SALO93]. Furthermore, most conventional CAD models do not appropriately represent tolerance and material information.

To overcome the deficiencies of feature recognition in conventional CAD models, the feature-based design approach captures both design and manufacturing intent by incorporating this information into a feature-based CAD model. Parts are modeled using a library of features; high level feature information is explicit and captured right from the start. Each feature in the library represents a parameterized shape associated with a set of geometrical and functional attributes. The Designer can add, create, and modify the design library which can be used with a set of operators which can add, delete, and modify a feature representation [CHAN92].

Feature-based modeling builds upon a number of advanced solid modeling techniques, such as parametric modeling and constraint-based modeling [BRON93]. In most feature-based CAD modelers, geometric representations of the part are based on solid models such as CSG, B-rep, or dual CSG and B-rep. The dual representation schemes have been suggested to be the most ideal geometric representation for feature based CAD models for the following two reasons [GARD93]: (1) CSG is non-unique and (2) B-rep does not contain generic information (history and constraints). The Quick Turnaround Cell (QTC) system [CHAN90] uses the dual scheme approach. Features are represented using the CSG model. The part design is represented by B-rep. The B-rep data structure has been modified by an extra field representing geometric entities. This extra field represents a tag that is used to mark each face of the B-rep with the design feature type used to create it; it also marks the orientation of the feature. Tolerance information is also included in the product model.

Although there are no agreed upon definitions of features[3], it can be viewed as a generic shape with some engineering meaning [SALO93]. Because the term "features" has different meanings depending on the application domain, many feature-based CAD systems use feature conversion techniques to map design features to application specific features [GARD93], [SALO93], [BRON93]. Bronsvoort and Jansen [BRON93] suggested the need for feature conversion among arbitrary application domains. This can be accomplished only with product model standardization and product data exchange. Lack of product model standardization in feature-based design has resulted in inconsistent feature library and conversion techniques.

A design in a feature-based CAD model is often referred to as a product model. A product model is composed of multiple feature models that are relevant to the feature-based design system. For design systems that incorporate process planning functions, the product model consists of form features, material features, and precision features. Form features refer to certain topological/geometric configurations on the surface of the part; such as holes, slots, pockets, and ribs. Material features refer to a part's material properties. Precision features refer to a part's variational geometry (dimension and tolerance).

Feature-based CAD modeling has a number of advantages over conventional CAD modeling methods. Product details needed by downstream functions are captured during the design process. This approach also simplifies the design process because features are more meaningful design elements than

---

[3] Various feature standardization has been initiated by CAM-I, PDES/STEP, and ESPRIT [BRON93].

geometric entities. Designs may cost less to produce through standardization of features used during the design process.

From the above, it can be concluded that the feature-based CAD model is the ideal form of input for achieving true generative CAPP. It provides designers with a rich vocabulary of features and the resulting part definition can be used by process planning functions for the automatic generation of necessary manufacturing details. In addition, a product model can be used by many functions to refine a design to ensure its feasibility and manufacturability before reaching the production stage. Various methods of process plan generation have been proposed within the feature-based environment and are reviewed below.

Lakko et al. [LAKK90] incorporated design evaluation into their feature-based CAD system consisting of parametric design, modeling of mechanical assembly, and process planning of prismatic objects for a 3-axis milling machine. The system uses LISP frames and object-oriented programming (OOP) concepts for representing the feature models; and manufacturing analysis is performed in the process planning module. It consists of: (i) General Manufacturability Analysis, (ii) Tooling Analysis and (iii) Generation of Change Proposals. General Manufacturability Analysis performs preliminary evaluation on producibility of each feature with respect to the size, and orientation. Tooling Analysis generates feasible cutting plans for the part based on the tool access direction, dimensional information and tolerance information. Although the system is able to provide manufacturability suggestions at a general level, it is not able to provide specific recommendations for individual features. In addition, the process planning module provides only a feasible machining sequence not including setups. The system also lacks clear representation of manufacturing resources such as tools and machinery.

Tonshoff et al. [TONS94] introduced a feature-based design environment called SESAME for design, process planning, and NC-programming. The product model consists of feature parameters (surface finish, tolerances, and dimensions) and form features represented in a CSG like model. Geometric tolerances are defined as explicit feature interactions. Designs are created using a set of features defined in the feature library. The output of the modeling system is a feature based workpiece description in a component description language (CODL). The CODL file constitutes the input for process planning. Design features are mapped to manufacturing features for process selection. The current process planning system covers a single machine with milling and drilling capabilities. Incomplete process plans can be completed interactively by the user. One major drawback to the system is the assumption of feasible designs as input to process planning. The system also lacks clear representation of manufacturing resources such as tools and machinery.

The QTC [CHAN90] is a feature-based design and process planning system that converts a design directly into final NC code for a single machining center. The architecture of the QTC is shown in Figure 2.6. Design features are converted to manufacturing features through feature mapping. Sequencing of features was determined at the global level and at the setup level. The main criterion for feature sequencing is to minimize the number of setups and tool changes. Feature precedence and tool approach directions are used to cluster features into setup groups and determine the order of operations within each setup. Precedence constraints are determined by the following information: (1) process constraint, (2) good manufacturing practice constraint, (3) tolerance constraint, and (4) datum constraint. Tool approach direction is used to group features into setup groups. Although a feature may have many tool approach directions, each feature may belong to only one of the final clusters. Features within each setup group must also satisfy the precedence constraint. The order of operations within each setup is determined by the commonality of tools and precedence. The main drawback to the system is the assumption of valid and feasible designs as input to process planning.

Chen and LeClair [CHEN94b] introduced a method for setup generation and a method for feature sequencing of machined parts. Working in a feature-based design environment, the following process plan related specifications are created: (i) Cutting-tool specification, (ii) Depth of cuts, (iii) Feed rates, (iv) Number of paths, and (v) Total machining time. Setup sequencing is created using a neural net type algorithm called "general unsupervised learning" using the following information: feature orientation, tool approach direction, and tool type. This method creates setup clusters based on the similarity of tool approach direction and tool type commonality. Similarly, feature sequencing is determined by using "episodal associative memory" to resolve geometric, i.e. feature interaction, tooling commonality. Feature interactions are addressed by defining types of feature intersections and using tool access directions to resolve the machining sequence. The main drawback to the approach is the lack of tolerance information. Without tolerance information, features are grouped in setups and sequenced only on the basis of tool approach and tool type. Thus the design will not be able to be manufactured as planned. In addition, the authors assume a feasible design as input which could result in generating infeasible setup and processing sequences.

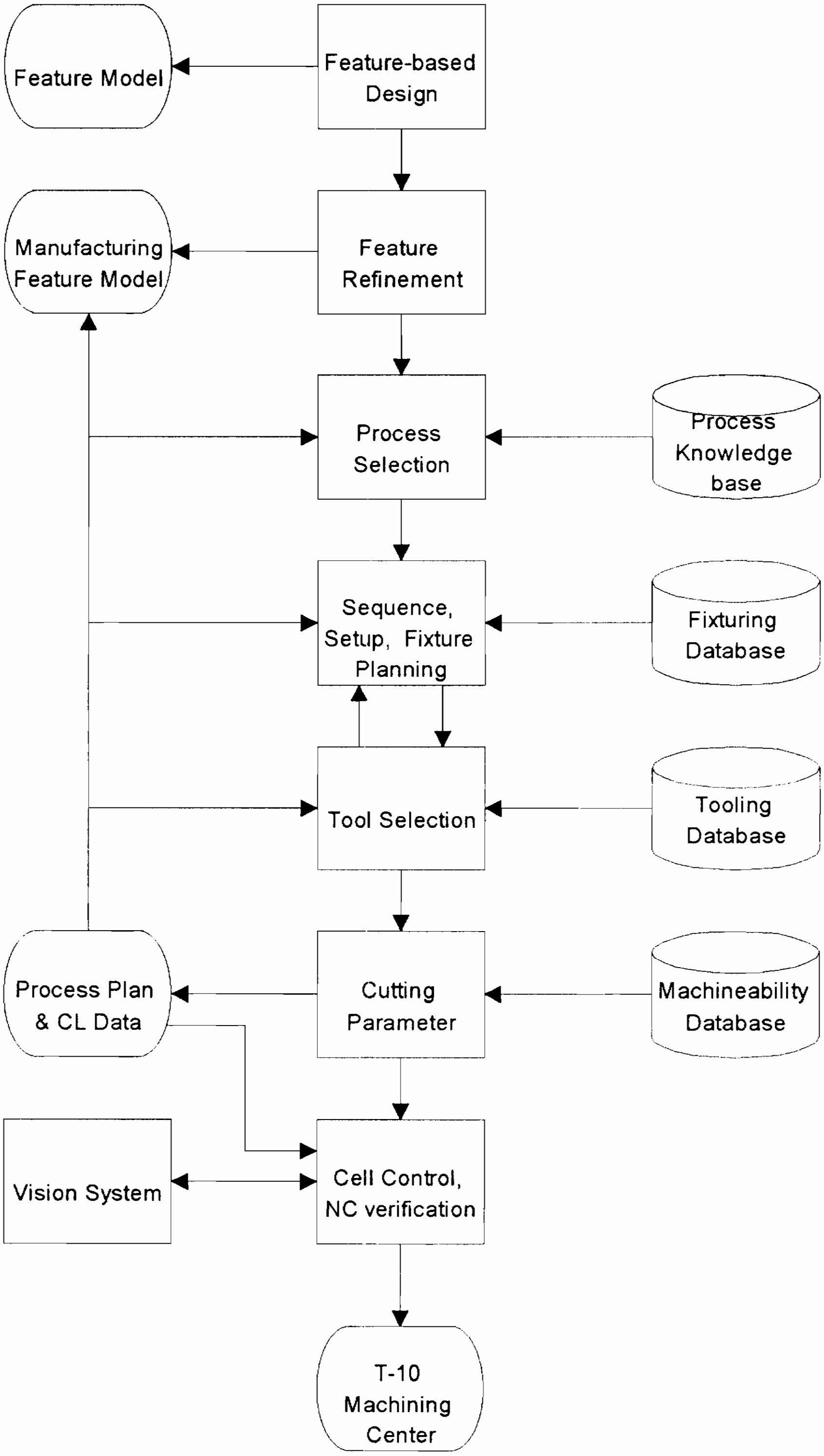


Figure 2.6. QTC System Architecture

Delbressine et al. [DELB94] proposed a method for automatic generation of setups for machined parts, given a feature-based design representation based on their design methodology of taking manufacturing restrictions into account. Manufacturing restrictions are enforced through the use of design features, called applied manufacturing object (ApplyMO), that are mappable to manufacturing operations. Feature-based form feature data structure is based on dual CSG and B-rep representation. Design transformations are first recorded and represented as a CSG tree; referred to as the design tree. The design tree records the initial state and all design operations successively applied to the design. Types of information used in setup generation include: internal and external (orientation, position relative to reference) tolerances, surface roughness, nominal position, and tool access direction. Setups are generated through the use of precedence and tolerance chart. Features meeting the precedence chart with higher tolerances are first selected. The feature is grouped in the current setup if its approach direction is feasible with respect to the machinery model; otherwise a new setup is created. The current design system is very limited, allowing only pockets to be created; through-holes or slots require two design operations. Only part of the total process planning is presented in the paper; all other aspects of process planning such as fixture planning, manufacturing operations sequencing and final generation are not explained. In addition, designs are not evaluated prior to process planning.

C. S. Chen [CHEN94a] introduced a feature-based product and process design framework based on the volume composition concept, similar to the method proposed by Woo [WOO82], to extract form feature information. This approach requires form features and their spatial relationships to be specified in the product model. Features need to be defined by type, volume, surface, location, orientation, and face constraints. Tolerance data is also included in the product model. A producibility evaluator plays an important function by providing feedback information on the feasibility, technical producibility, and costs and schedule compatibility of a designed part. The proposed process planning procedure (Figure 2.7) converts each form feature into a material removal volume (MRV) and assigns a machine/tool instance. The initial work material and raw stock size is determined by traversing the product model from base node to initiate a bounding box and then appending the remaining features one at a time to the bounding box. Each feature (MRV) is grouped according to its orientation and approach directions, and sequenced based on its affiliation with datum references. Feasible manufacturing resources are assigned and then selected based on evaluation of the process plan. Information missing from the framework is the procedure for manufacturing resource assignment. A Machinery model is needed to determine the clustering of setups. No details were provided on the methods of evaluating the producibility of a particular design.

K. K. Krishnan [KRIS94] proposed a DFM framework in which designs are analyzed for manufacturability before and after the process planning stage (Figure 2.8). DFM Feasibility Module is the first stage that checks the feasibility of designs, such as dimension, tolerance and material compatibility. Process Planning Module is the second stage that generates a feasible plan for the design. DFM Analysis Module is the third stage that analyzes the process plan for any possible improvements. The DFM framework assumes a feature-based CAD file based on B-rep. Types of Information that are included in the data structure and are critical in both DFM analysis and process planning are feature orientation, feature spatial relationships, tool accessibility, material, and tolerance information (Figure 2.9).

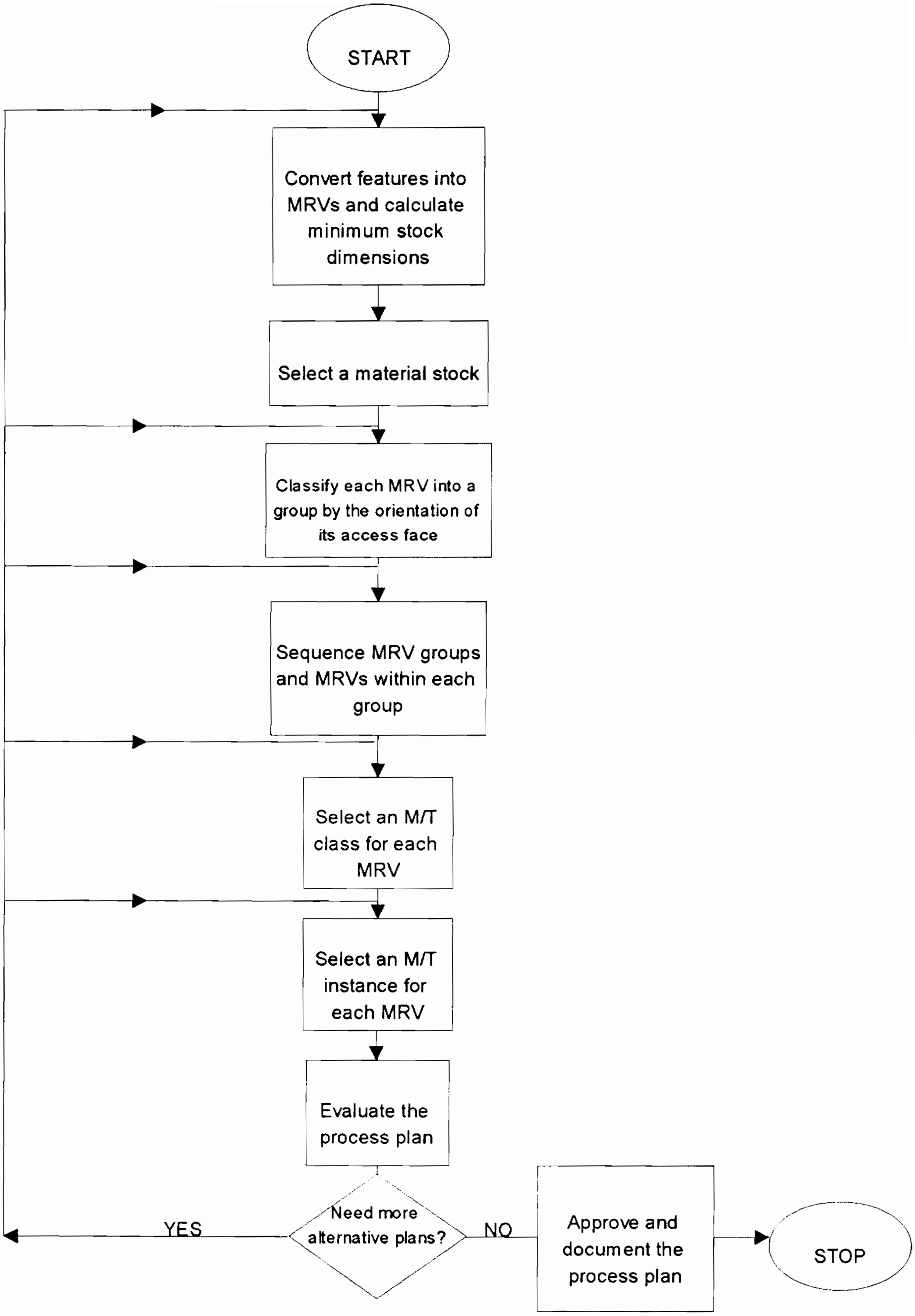


Figure 2.7. Process Plan Generation Procedure

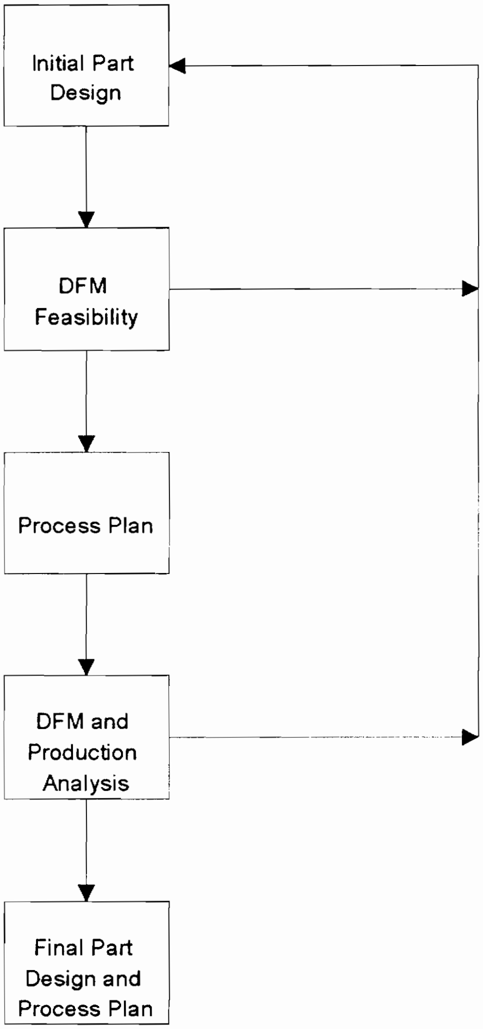


Figure 2.8. DFM Framework Description

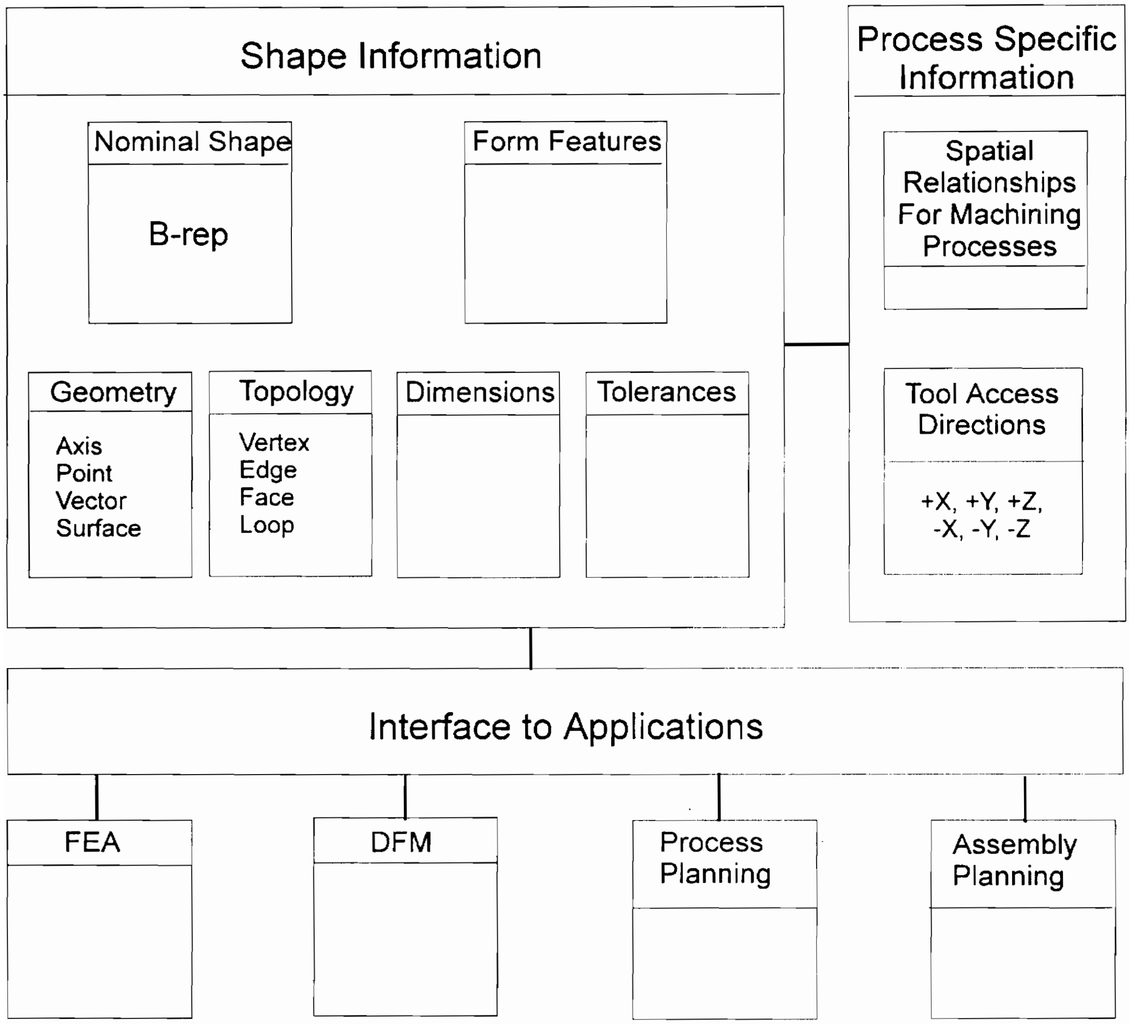


Figure 2.9. DFM Data Representation Framework

2.2.3 Process Plan Representation

Two methods of representing process plans, using object oriented representation and directed graph notation, are discussed below.

Object Oriented Model

A proposed object oriented representation of process plans by C. S. Chen is shown on Figure 2.10 [CHEN94a]. The process plan in this representation scheme is composed of many sub-processes; each sub-process consists of one setup specification and one or more operation clusters. Each operation cluster contains specifications for producing a form feature.

Directed Graph Notation

Manufacturing functions downstream from process planning need information such as alternative sequencing approaches, are resource requirements. An intelligent scheduler, does not need to understand the logic behind a machine selection. It only needs to know the processing time and resource required. Tasks or sequences generated through plan formulation need to be expressed in a manner that facilitates downstream systems. The following section will discuss the application of direct graph notation for representing process plans with alternative paths.

Directed graph notation represents temporal relationships between process tasks (Figure 2.11). Many efforts in control systems use directed graph as their basis because of the following attributes [CATR91]: simplicity, clarity, precedence, alternative sequence, parallelism, and abstraction.

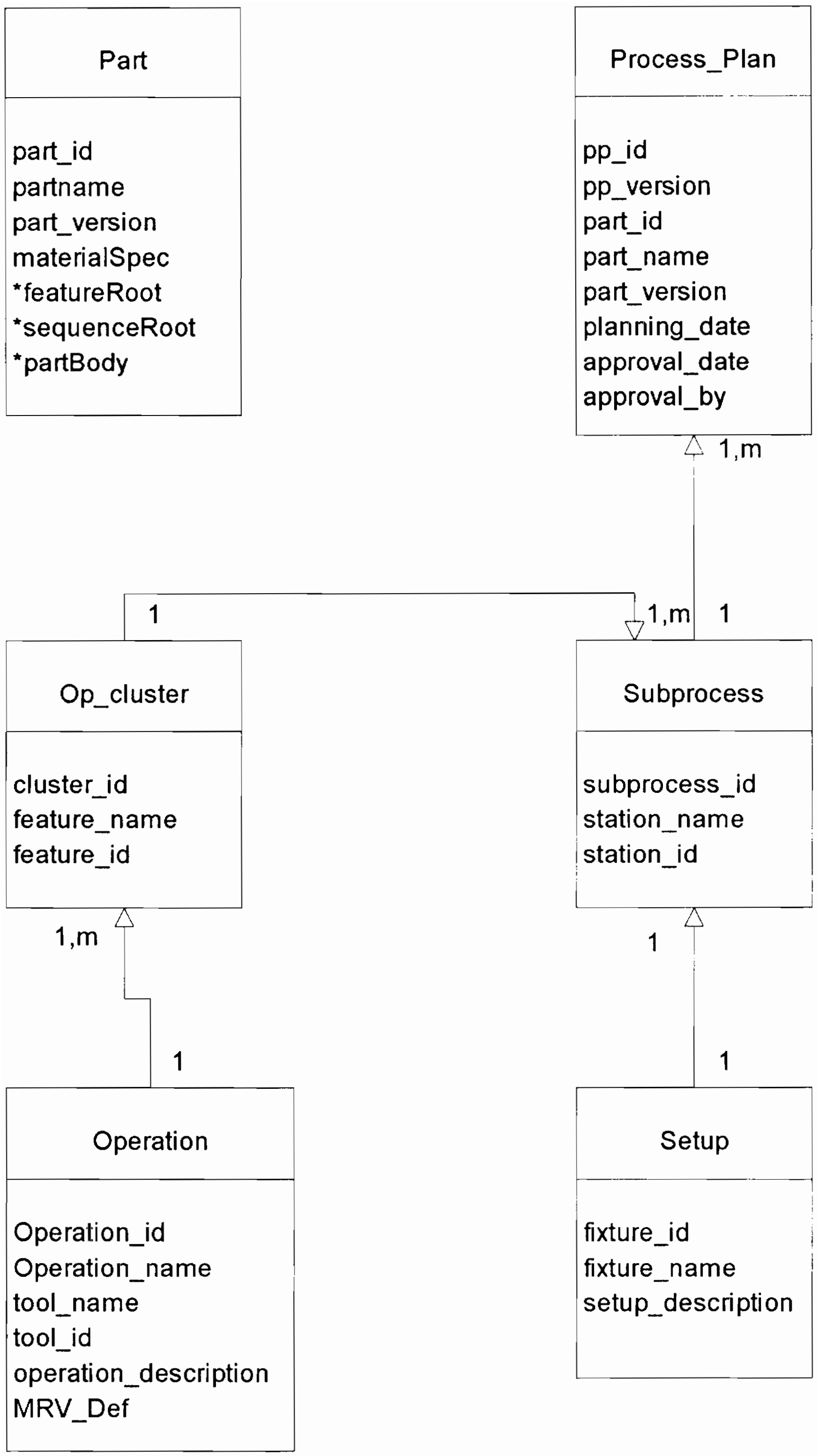


Figure 2.10. Process Plan Representation Structure

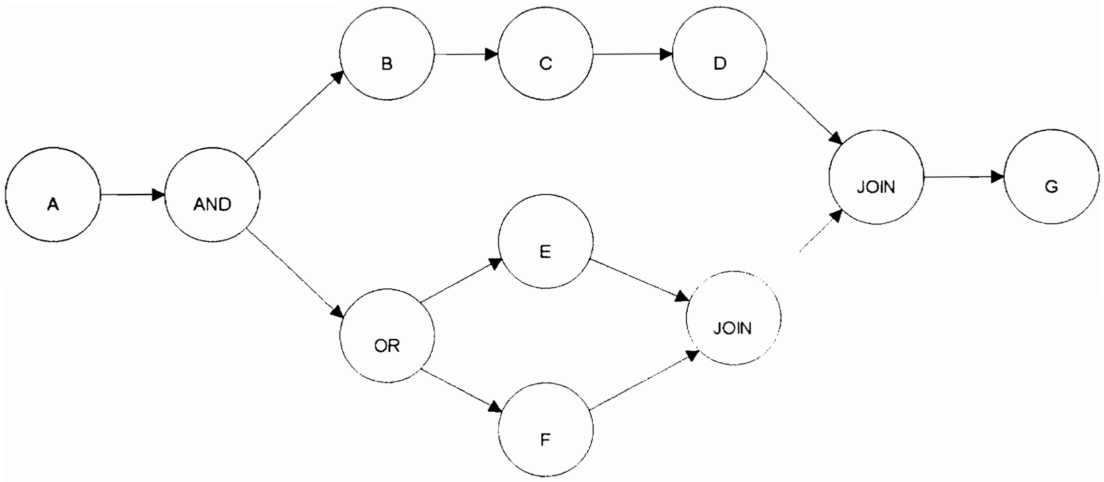


Figure 2.11. Direct Graph Notation Example

Petri Net (PN) is a form of directed graph. This framework was used by [SRIH90] and [KRUT92] as a way of representing real-time status information for process planning. The arc links represent time intervals between connecting nodes and nodes represent a set of places and transitions.

A Language For Process Specification (ALPS) was introduced by Catron and Ray [CATR91]. ALPS is built around a directed graph structure. There are seven classes of graph nodes used for describing process specification and sequence. Each class will be discussed in detail.

- **Termination nodes** are denoted by *START* and *END* to delimit the beginning and termination of a graph. These nodes contain no processing information.
- **Task Nodes** are denoted as *PRIMITIVE* and decomposable tasks such as *COMPLEX* or *MACRO*. *PRIMITIVE* task nodes specify activities to be executed directly by the target controller; attributes for these nodes are user-defined for specific tasks. *COMPLEX* nodes allow for hierarchical abstraction to be executed by another system. *MACRO* task nodes specify another decomposable process plan; equivalent to a subroutine.
- **Split Nodes** are used for representing (alternative, concurrent, iterative) branching paths. Two subclasses are denoted by *PREDICATED* and *PARAMETERIZED*. The split subclasses are a super-set of *AND, OR, ELSE* nodes of generalized and/or graphs. *PREDICATED* nodes allow for the evaluation of boolean functions associated with each outgoing path. Paths that are evaluated to be true may have their path followed. If no paths are eligible, *ELSE* node path can be taken. *PARAMETERIZED* nodes allow path(s) to be chosen based on parameters associated with each outgoing path.

- **Join Nodes** are denoted by *MULTIPLE* and *SINGLE* nodes. These nodes join executed path(s) and can be used also as a loop control.
- **Resource Nodes** denoted by *ALLOCATE* and *DEALLOCATE* provide for explicit specification of resource management.
- **Synchronization Nodes** are used for multi-tasking functions. This class is divided into two subclasses: semaphore and clock. Semaphore nodes denoted by *RENDEZVOUS, SIGNAL, AWAIT, LOCK*, and *UNLOCK*. Clock nodes are denoted by *WAIT* and *DELAY*.
- **Information Nodes** provide for general purpose, user definable operations.

While directed graph notation is ideal for representing the temporal relationship among sequential tasks, it is more difficult to represent the details of those tasks in a simple manner. Directed graph does not have a rich set of tools for representing multi-level information. The object oriented model, on the other hand, is capable of representing directed graph notation. In addition, representing multi-level information, as shown by C. S. Chen [CHEN94a] is possible using the object oriented model.

2.3 Summary

A major obstacle towards achieving true generative CAPP is the transfer of sufficient and explicit feature information from design. Semi-generative CAPP systems vary in the input format of the required feature information. These systems operate independent of a geometric model and require direct user input of the feature information in a customized syntax or in an interactive manner and thus do not meet the needs of true generative process planning.

True generative CAPP systems reason directly from CAD models. Features are extracted from a conventional wireframe or solid model with automatic or manual feature recognition techniques. The feature recognition process, however, is complex and often works for a limited set of geometry. In addition, conventional CAD models do not contain tolerance and material information needed for process planning. Although there is no standardized feature-based CAD model, it represents the best approach for including process planning information. Feature based CAD model have most potential for realizing the goal of true generative process planning.

Feature-based CAD models are built on top of solid modeling techniques. Most geometric information in feature-based CAD models uses either CSG, B-rep or a dual representation scheme. Dimensional, tolerance and material information needed for process planning can be maintained in the model. There is a need to define a set of standard library of features that can be converted/mapped among different applications. CAPP should be performed after a part has been analyzed for its feasibility and manufacturability. Most research in true generative process planning assumes a valid design as input and lacks a complete methodology for analyzing the manufacturability of a design. Additionally, process plans need to be evaluated relative to the status of the production system. Some machines can be over utilized, which could lead to production delays.

CHAPTER 3

METHODOLOGY

This chapter presents the research methodology for developing a true generative CAPP module that will be an integral part of a DFM framework for machined components. The proposed research is divided into the following phases:

1. Construct A Framework For True Generative CAPP
2. Identify Data Requirements For Process Plan Generation
3. Construct Machine Data And Tool Data Representation Scheme
4. Construct Process Plan Representation Scheme
5. Develop Algorithms For Process Plan Generation

Section 3.1 provides a general framework for true generative process planning followed by the details of the proposed procedure for process plan generation. Data requirements and representation structure for machine data, tool data, and the process plan will be discussed in section 3.2. Section 3.3 summarizes the expected contributions of this research.

3.1 CAPP Module Description

The approach for true generative CAPP requires feature-based designs for automating the generation and analysis of process plans. The proposed

CAPP Module will be an integral part of a DFM methodology[4] (Figure 3.1) which consists of three major components: (1) Preliminary DFM Feasibility, (2) Process Plan Generation and (3) DFM Value Analysis. Designs are first created in a feature-based design system. After initial designs are completed, preliminary DFM Feasibility analysis is performed to check for any initial design inconsistencies in the design, i.e., material, dimensions, tolerances, and part configuration. Any inconsistencies, such as locating a surface by more than one dimension and assigning parallelism tolerance on two perpendicular surfaces, are feedback to the designer for modifications. The next phase following preliminary DFM Feasibility analysis is developing feasible process plan(s) for DFM Analysis by clustering feasible features sequences into groups of setups and then allocating operation sequences within each setup. The DFM Value Analysis Module (which covers tolerance control, tolerance analysis, and configuration analysis) then investigates the process plan for the possibility of improving the design or selecting the best process for production. Any suggestions are then feedback to the designer for modifications.

The function of the CAPP Module is to generate feasible process plan(s) for value analysis. The work flow for process plan generation is illustrated in Figure 3.2 and is performed by five process dependent modules: (1) Process Selection, (2) Machine Tool Selection, (3) Setup/fixture Planning, (4) Operation Sequence Planning, and (5) Process Plan Evaluation. The details of each module and its function are described below.

---

[4] This DFM design environment is based on K. K. Krishna's dissertation work on DFM methodology and data representation framework for machined components [KRIS94].

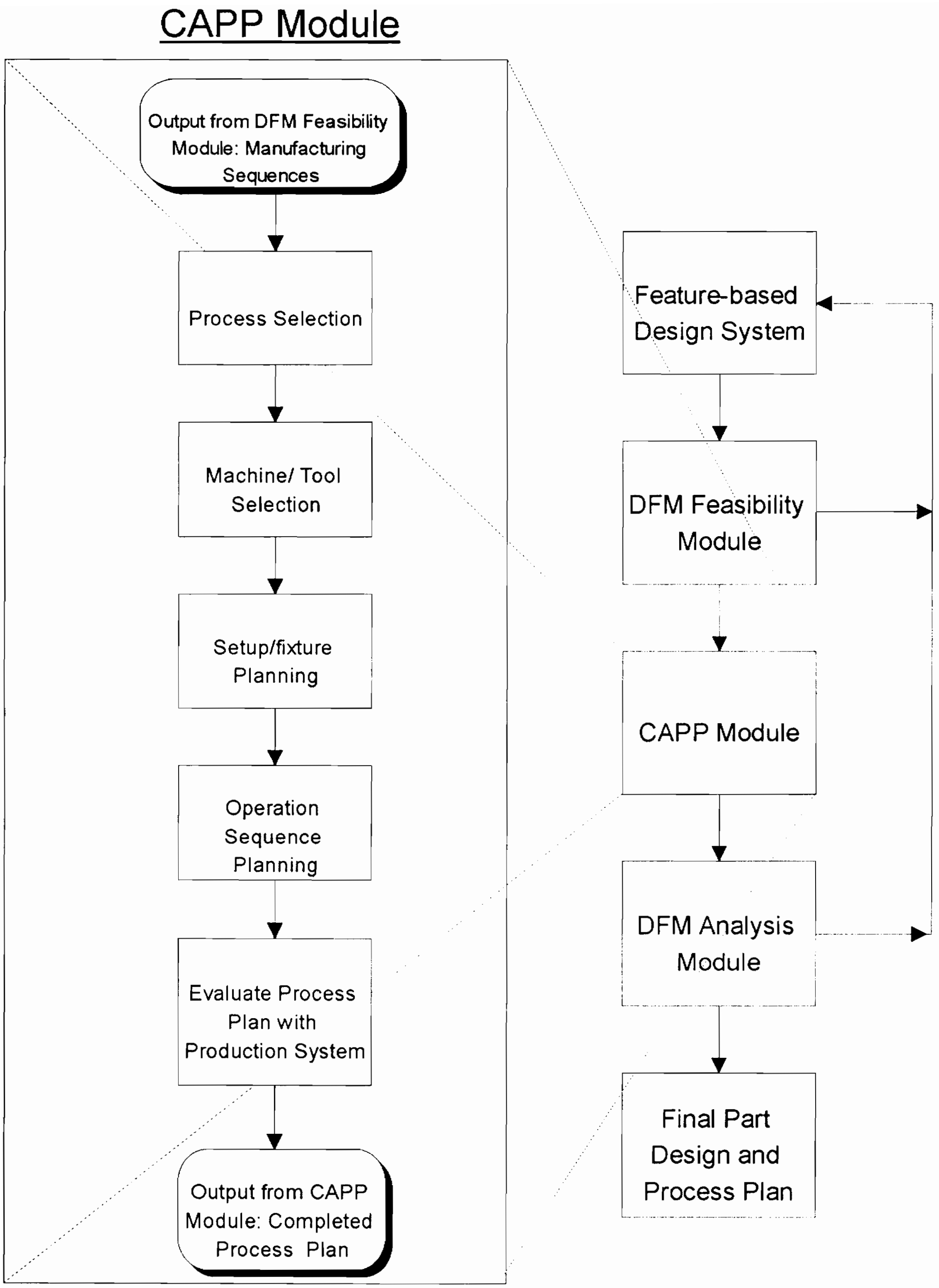


Figure 3.1. DFM Framework With CAPP Module Close-Up

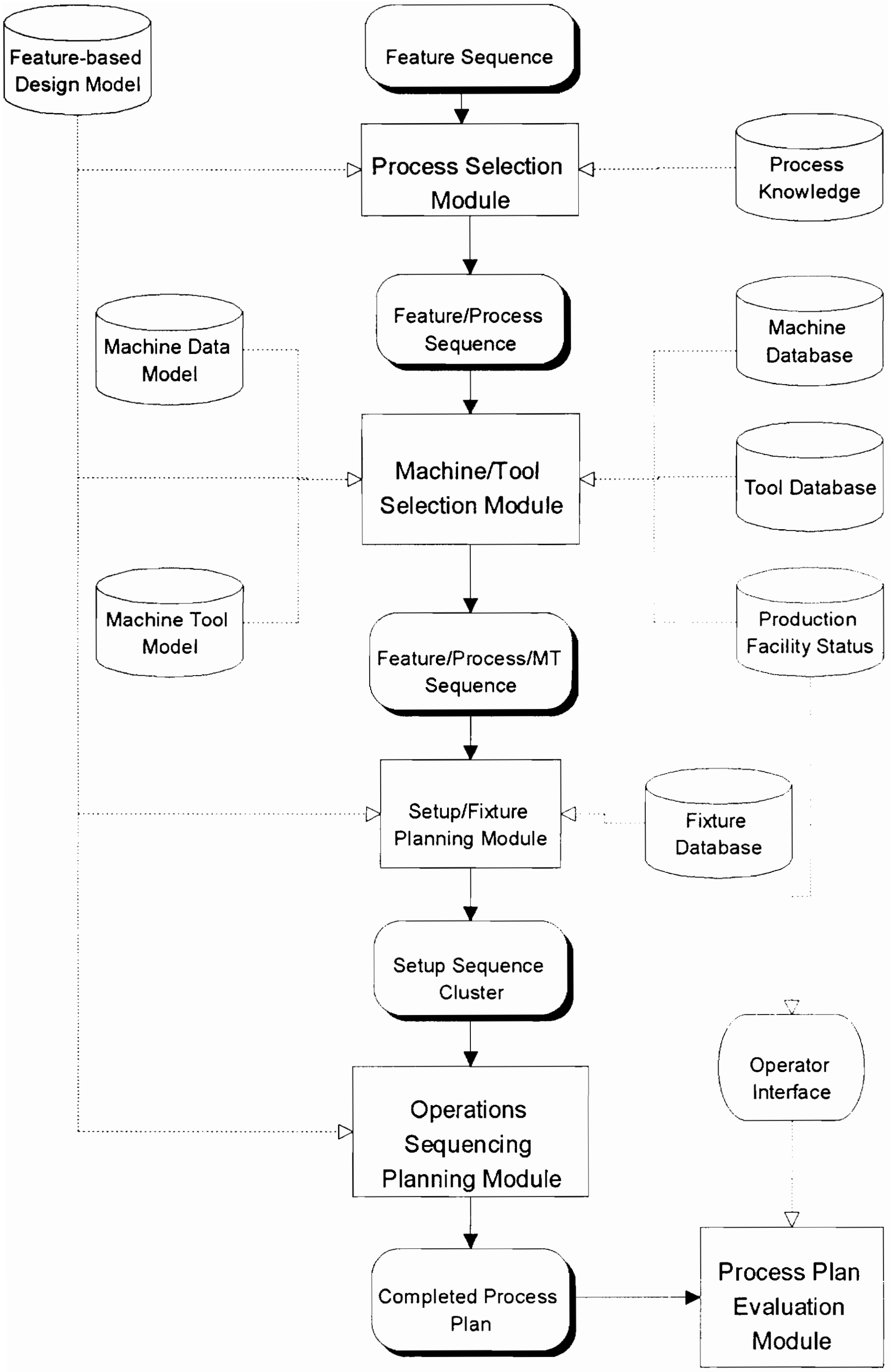


Figure 3.2. CAPP Module Work Flow

3.1.1 Process Selection Module

Input to the Process Selection Module is a feasible manufacturing feature sequence. The main function of the module is to assign feasible manufacturing processes to each feature. Machining processes are selected by matching the manufacturing features and their tolerance specifications with the tolerance capabilities of the processes. Features used in the DFM Design Phase correspond directly to manufacturing processes; feature recognition algorithms are not needed.

Mapping of design features into manufacturing processes is shown in Figure 3.3. Each feature may be assigned more than one feasible process; the actual selection of the ideal machining process will be considered in the Setup/Fixture Planning Module. Mapping information is kept in the Process Knowledge Database which consists of a collection of machining process capabilities relating to each design feature. Figure 3.4 illustrates the hierarchic linking of feasible manufacturing processes for each design feature. The Process Selection Module is called a feature/process sequence (Figure 3.4).

3.1.2 Machine/Tool Selection Module

The Machine/Tool Selection Module selects feasible machine/tool pairs for manufacturing processes. Feasible machine/tool pairs denote a general machine and tool type. The structure of machine and tool representation is constructed from the Machine Data Model and Machine Tool Model (Figure 3.2). Feature information such as dimension and tolerance, along with process information are used to select feasible machine and tool types.

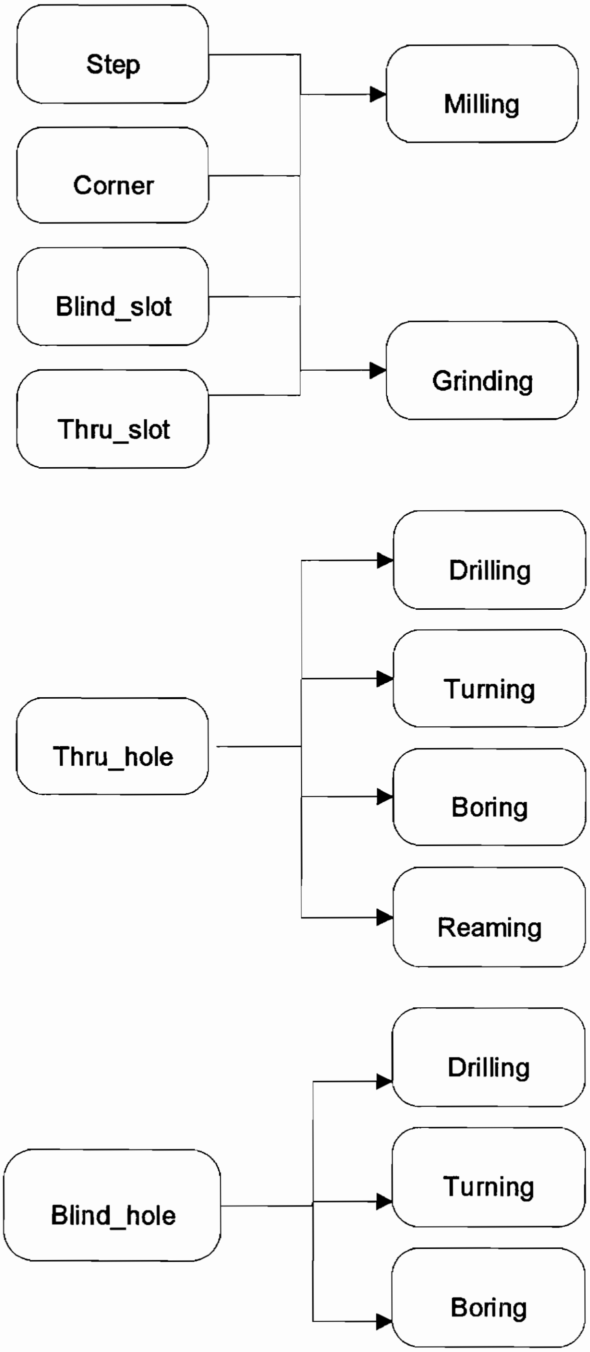


Figure 3.3. Mapping Of Design Features To Machining Processes

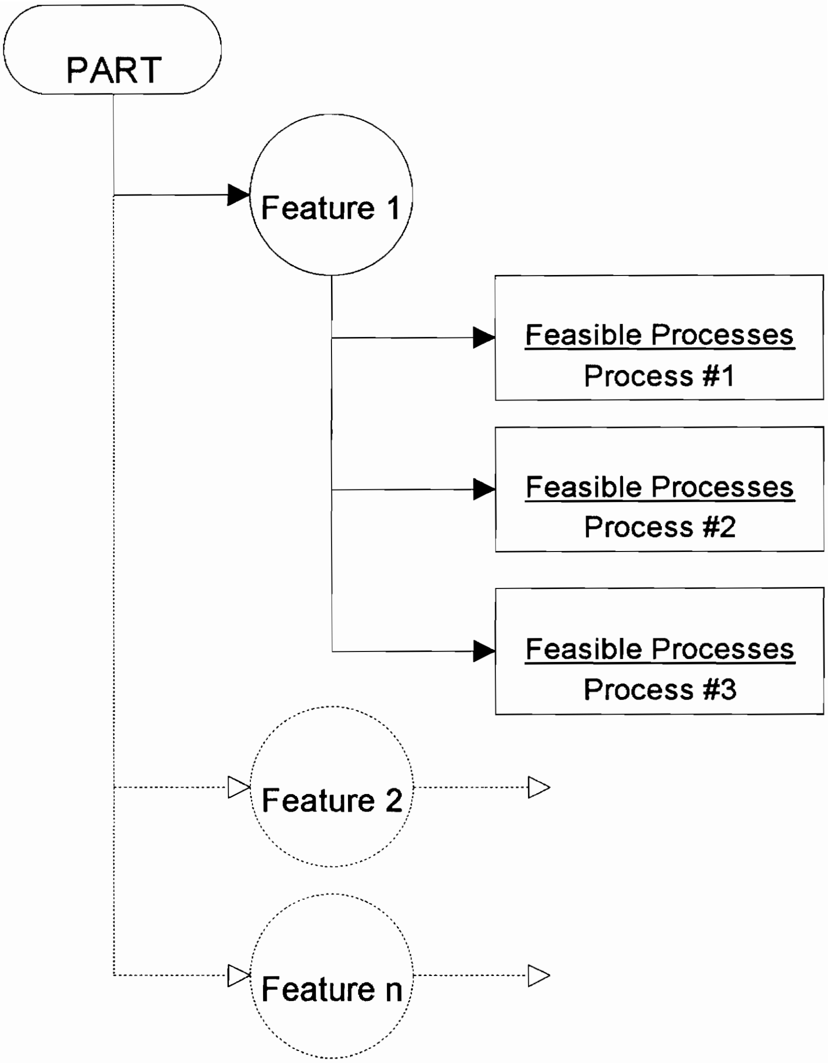


Figure 3.4. Feature/Process Sequence List

Various classes of machinery and tools are defined in the machine database and in the tool database. Specific instances of machines and tools as well as their status in the production facility are kept in the Production Facility Database (Figure 3.2).

Figure 3.5 illustrates the linking of feasible machine/tool to existing feature/process sequence; the output is the Feature/Process/MT Sequence. The final selection of a specific machine/tool pair is made in the following Setup/Fixture Planning Module.

3.1.3 Setup/Fixture Planning Module

The previous modules identified a number of feasible manufacturing resources: processes and machine/tools for each feature. The function of the Setup/Fixture Planning Module is to select the most appropriate manufacturing resource (depending on criteria) and divide features into subsets of setup clusters. A setup specifies the work-material position and orientation on the selected manufacturing machine plus the specified fixture for part fastening.

The default criterion is to minimize cost by minimizing the number of setups. To minimize the number of setups, the number of features to be generated on a single machine in a single setup is maximized. The following procedure is the proposed algorithm for the automatic generation of setups for minimizing costs:

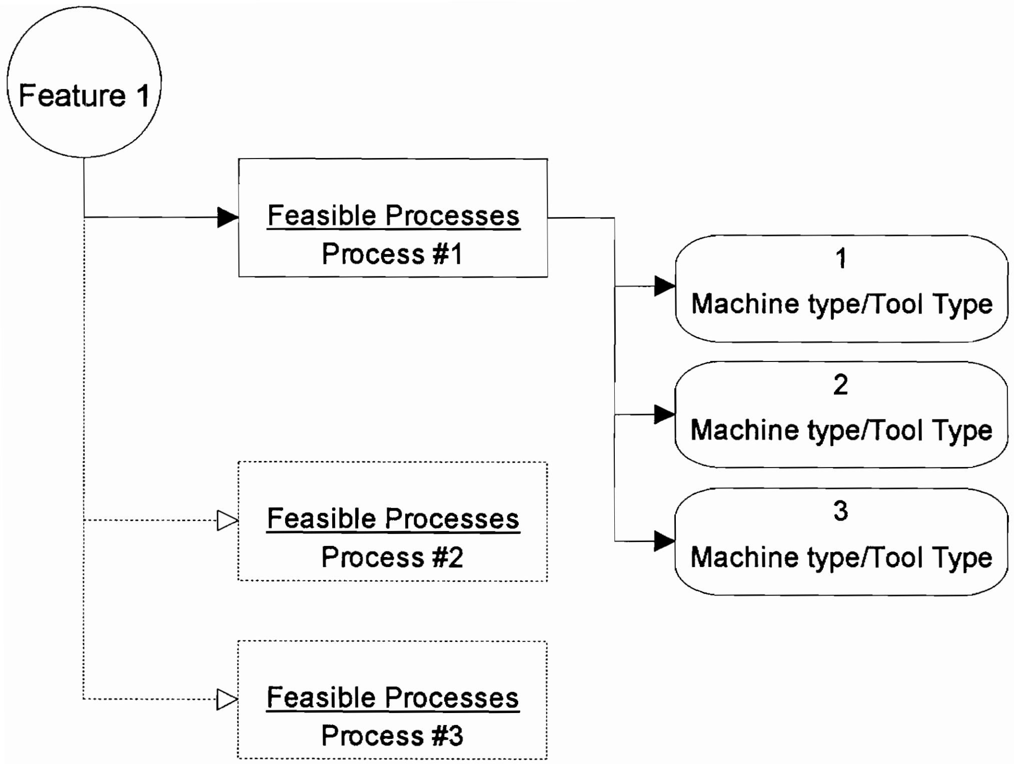


Figure 3.5. Feature/Process/MT Sequence List

1. Surface/features are organized into clusters of six tool approach directions: X1, Y1, Z1, X0, Y0, Z0 (where 1 denotes the positive direction and 0 denotes the negative direction).
2. Remove any cluster violating the precedence constraint in the feature sequence list.
3. Search for the most common machine type within each remaining cluster, weight each cluster by the number of machine commonality.
4. Features within the cluster with the highest machine commonality are grouped as a setup group. Resolve ties with the cost index.
5. Remove any features appearing in the setup group from the remaining clusters.
6. Repeat steps 4 and 5 until all features belong to a setup group.

To generate an alternative process plan, feasible machines and tools assigned to each surface/feature that were not selected on the first setup are again considered and the setup generation procedure is repeated. The generation of alternative process plans can continue until all feasible machines or tools of a surface/feature have been selected. In addition, alternative process plans should also be generated by choosing a different criteria or by user specification.

Minimizing time is another criteria that can be used to generate process plans. The objective of the setup planning algorithm attempts to create as many setups as possible to minimize the processing time at each machine. The following procedure is the proposed algorithm for the automatic generation of setups for minimizing time:

1. Determine the number of distinct machines assigned to the set of surface/features.
2. Determine the average surface/feature per machine, total number of surface/features / number of machines (Refer to this value as AVG).
3. Select the machine with the fewest number of feasible surface/features assigned.
4. Determine the largest number of feasible surface/features assigned from a single tool access direction that does not violate the precedence constraint.
5. If the number determined in step 4 is less or equal to the AVG, select these feasible surface/features to a setup. Else, determine the largest number of tool commonality within feasible surface/features identified in step 4. If this value is less or equal to AVG, select these feasible surface/features to a setup. Else, arbitrarily select AVG feasible surface/features, meeting the precedence constraint, to a setup.
6. Repeat steps 1 through 5 until all feasible surface/features are selected to a setup.

The output of this module is the setup cluster that includes the number of setups and its features. The sequence of the setups as well as the operation sequence is determined in the next module.

3.1.4 Operation Sequence Planning Module

The function of the Operation Sequence Planning Module is to organize setup sequence among setups and feature sequence within each setup. Setup sequence is determined by the precedence relationship and machine type commonality. Features that are grouped in a setup must satisfy its precedence relationship with features in other setups. An operation sequence consists of one or more material removal processes using one or more cutting tools on a workpiece in a single setup on a piece of equipment. The operation sequence within each setup is determined by precedence and by tool commonality. The output of this module is a complete process plan, with possible machining parameters and NC code generation in future implementation. However, the merits of a process plan should be based on its impact on the overall production system. Within the DFM Framework, the completed process plan can be further analyzed with respect to its cost, required production rate and the available manufacturing resources. The next module addresses these areas.

3.1.5 Process Plan Evaluation Module

The function of the Process Plan Evaluation Module is to allow users to evaluate the process plan. This module should be performed after the completed process plan has been evaluated by the DFM Analysis Module. Figure 3.6 shows the details of this module which is composed of three sub-modules: Cost Determination, Resource Utilization, and Production Requirement.

The function of the Cost Determination sub-module is to estimate the production cost of the given process plan. The estimated production cost covers the raw material, machinery cost, and tooling cost. The raw material cost is determined by the type of material and the amount of raw material used. The machinery cost is determined by adding up the machining time multiplied by the machine's cost; each type of machine is assigned a machining cost ($/time) in the Machine Database. The tooling cost is also determined in the same manner as the machinery cost. Each tool, depending on the type of tool and the material has an assigned cost in the Tool Database.

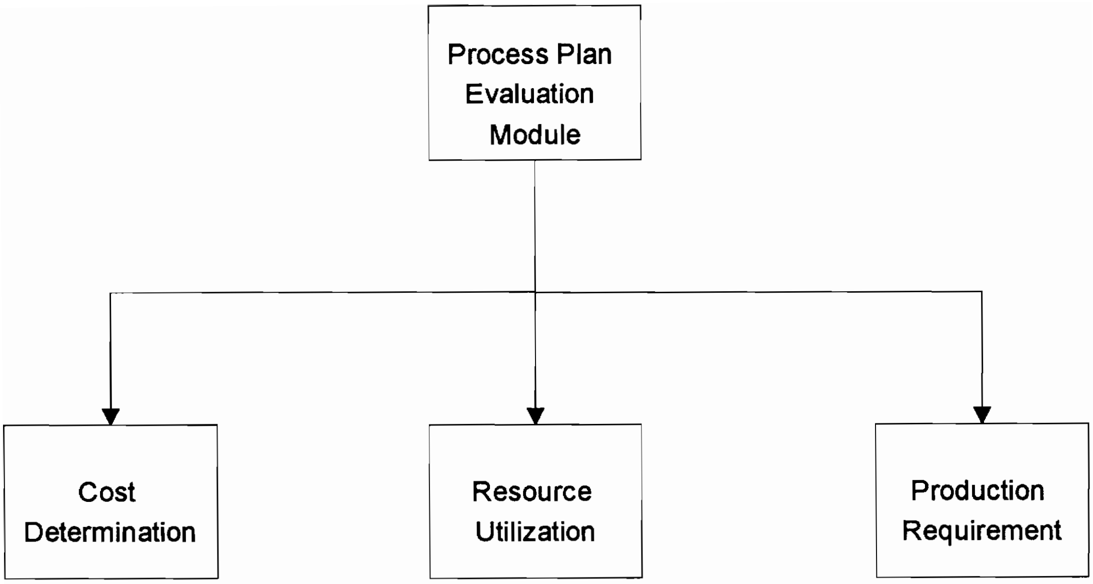


Figure 3.6. Process Plan Evaluation Module

The function of the Resource Utilization sub-module is to perform capacity utilization of machines and tools selected in the process plan. Utilization of the production resource should be balanced. Uneven utilization of equipment can create an unpredictable maintenance schedule in which some machinery and tooling may have earlier breakdowns. Machinery and tooling utilization can be based on time of usage divided by the time since it was first introduced (or reintroduced) into the system. This information can be kept in the Machine Database and Tool Database.

The function of the Production Requirement sub-module is to ensure that the required production rate of the part can be achieved with the equipment selected. This module would need additional information about the production facility such as material handling information, fixturing information, along with the machinery and tooling information. In addition, this sub-module also needs production scheduling information so that due dates can be met.

3.2 Data Representation Scheme

Feature-based data representation structure developed by K. K. Krishnan [KRIS94] will be used to provide design part information for true generative process planning. Design part information includes nominal shape, form feature, tolerance, and spatial relationship information. Besides design part representation, process planning also needs data representation of the following manufacturing resources: (1) Machine Data and (2) Tool Data. The data structure of machine data and tool data is covered in sections 3.2.1 and 3.2.2 respectively. Representation of process plans is critical to facilitating downstream operations, i.e. production scheduling. The data structure of process plans is described in section 3.2.3.

3.2.1 Machine Data Representation Structure

A model for representing machine data is shown in Figure 3.7. The machine data model is composed of three classes: (1) Machine Information, (2) Capability Information, and (3) Cost Information. Machine Information Class provides a general identification of the machine instances. Machine Capability Class provides a physical and processing capability of the machine instances. Cost Information Class provides information that is useful in the evaluation of the process plan, such as an operating cost index. Each class is described below.

Machine Information Class

Machine Information Class includes Machine_identifcation and Availabiltiy_information sub-classes (Figure 3.8). Machine_idenfication sub-class includes description data identifying the machine instance by name (e.g. milling machine), type (e.g. 3-axis CNC), and machine number. Availibility_information sub-class provides the status of the machine instance (e.g. in_use) and the time when the status is estimated to change (e.g. 12:00 a.m. 9/8/94).

Capability Information Class

Machine Capability Class includes Process_information, Physical_dimension, and Capability_information sub-classes (Figure 3.9). Data in the Process_information sub-class describes the types of process the machine instance is capable of performing (e.g. drilling); as well as the types of spindle it uses. Data in the Physical_dimension sub-class describes the physical size of the worktable and the working envelope of each machine.

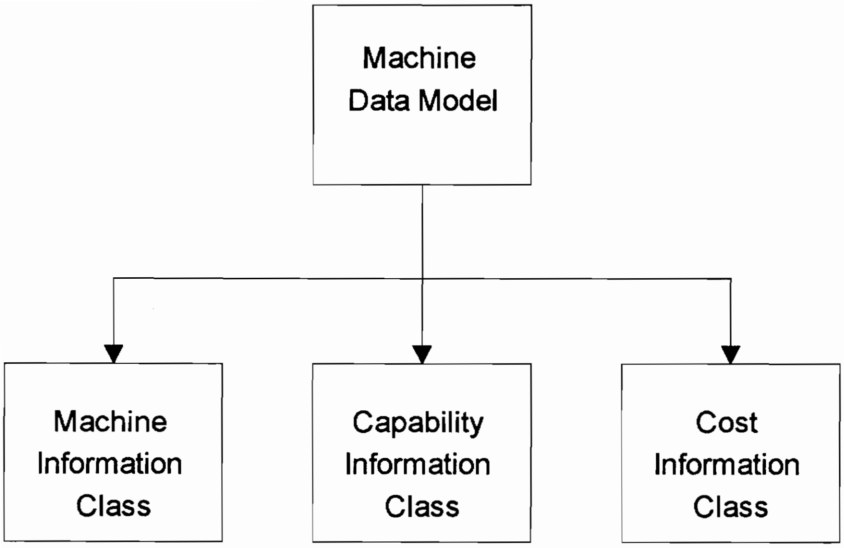


Figure 3.7. Machine Data Model

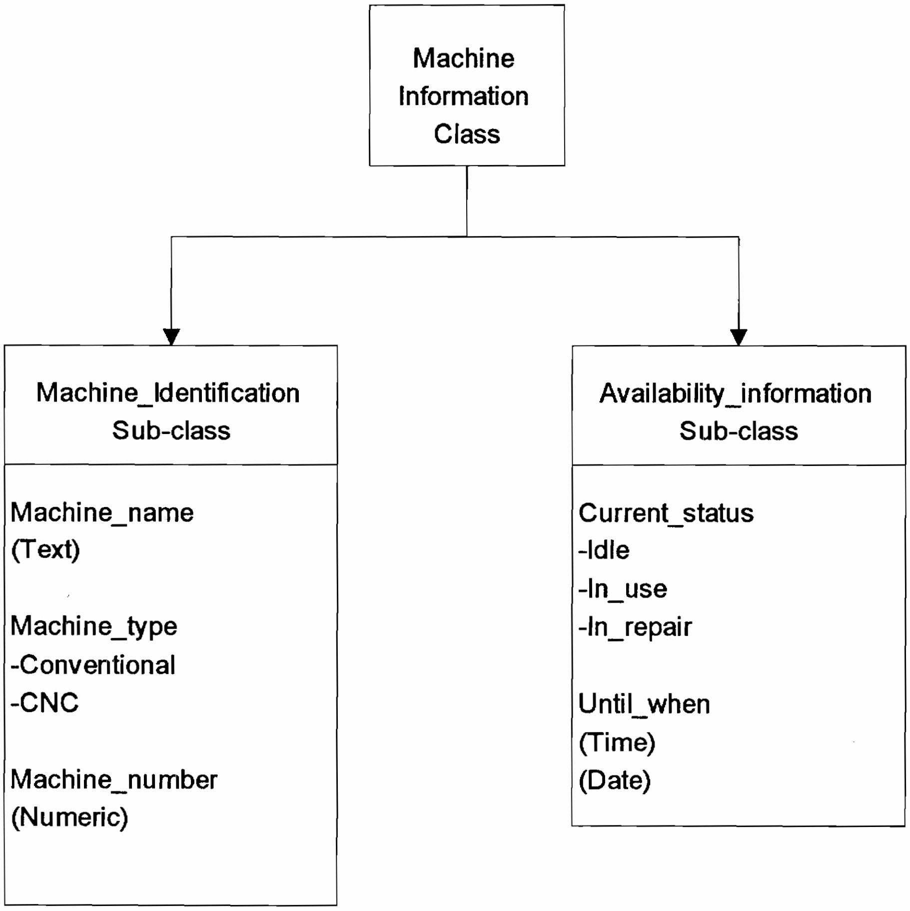


Figure 3.8. Machine Information Class

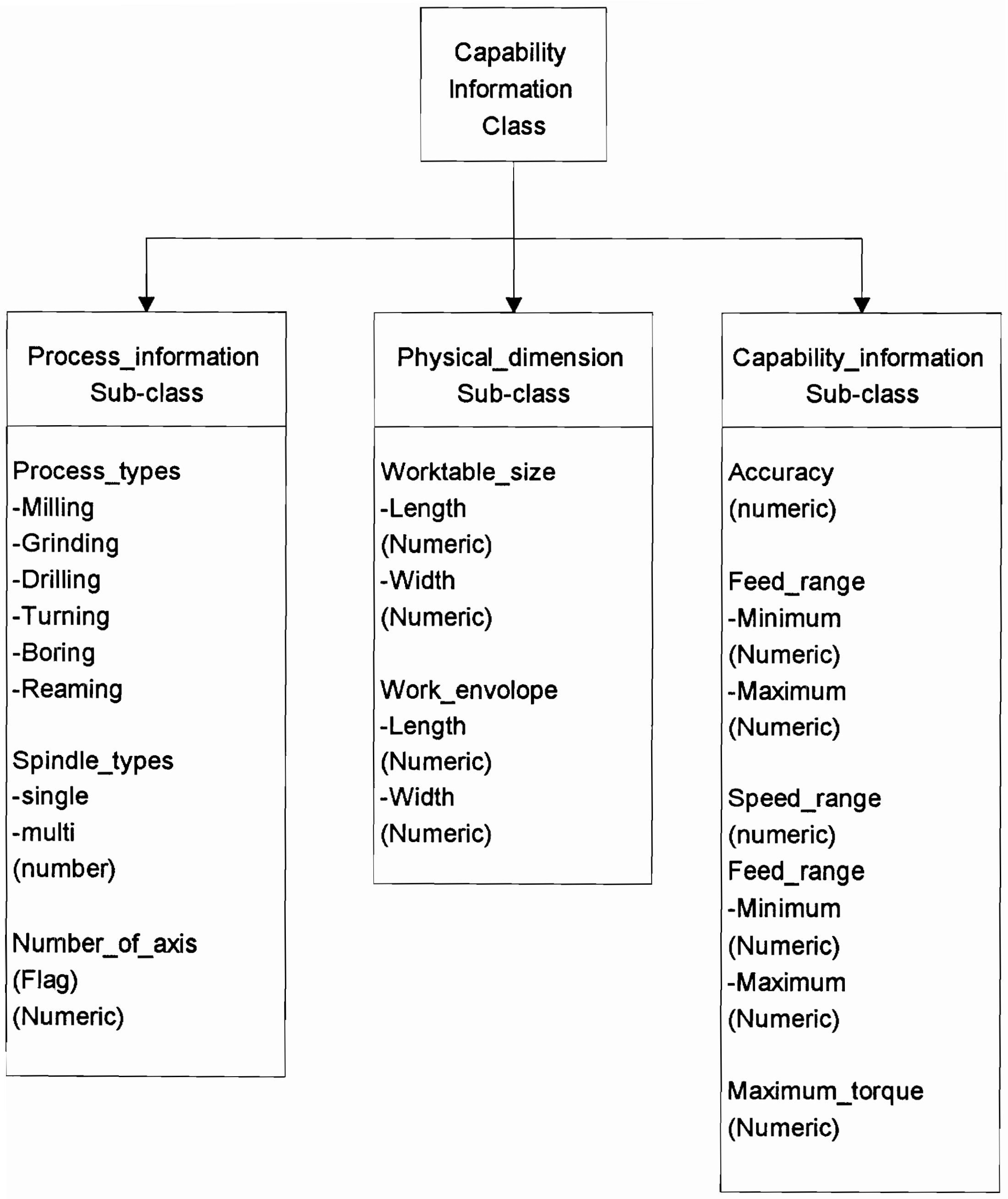


Figure 3.9. Capability Information Class

Data in the Capability_information sub-class provides information on a machine's accuracy, feed range, speed range, maximum torque, and number of axes.

Cost Information Class

The Cost Information Class (Figure 3.10) is composed of three sub-classes of data that is used by the Process Plan Evaluation Module: Cost_information, Resource_utilization, and Production_requirement. The Cost_information sub-class provides data on the operating cost of the machine instance as well as its maintenance cost and schedule. The Resource_utilization sub-class provides a utilization rate of the machine instance. The Production_requirement sub-class provides an estimated production rate of the machine instance.

3.2.2 Tool Data Representation Structure

A model for representing tool data is shown in Figure 3.11. The tool data model is composed of three classes of data similar to the Machine Data Model: (1) Tool Information, (2) Capability Information, and (3) Cost Information. Tool Information Class provides a general description of the tool instances. Tool Capability Class provides detailed physical and processing ability of the tool instances. Cost Information Class provides information that is useful in the evaluation of the process plan. Each class is described below:

Tool Information Class

Tool Information Class includes Tool_identifcation and Availability_information sub-class (Figure 3.12). Tool_idenfication sub-class

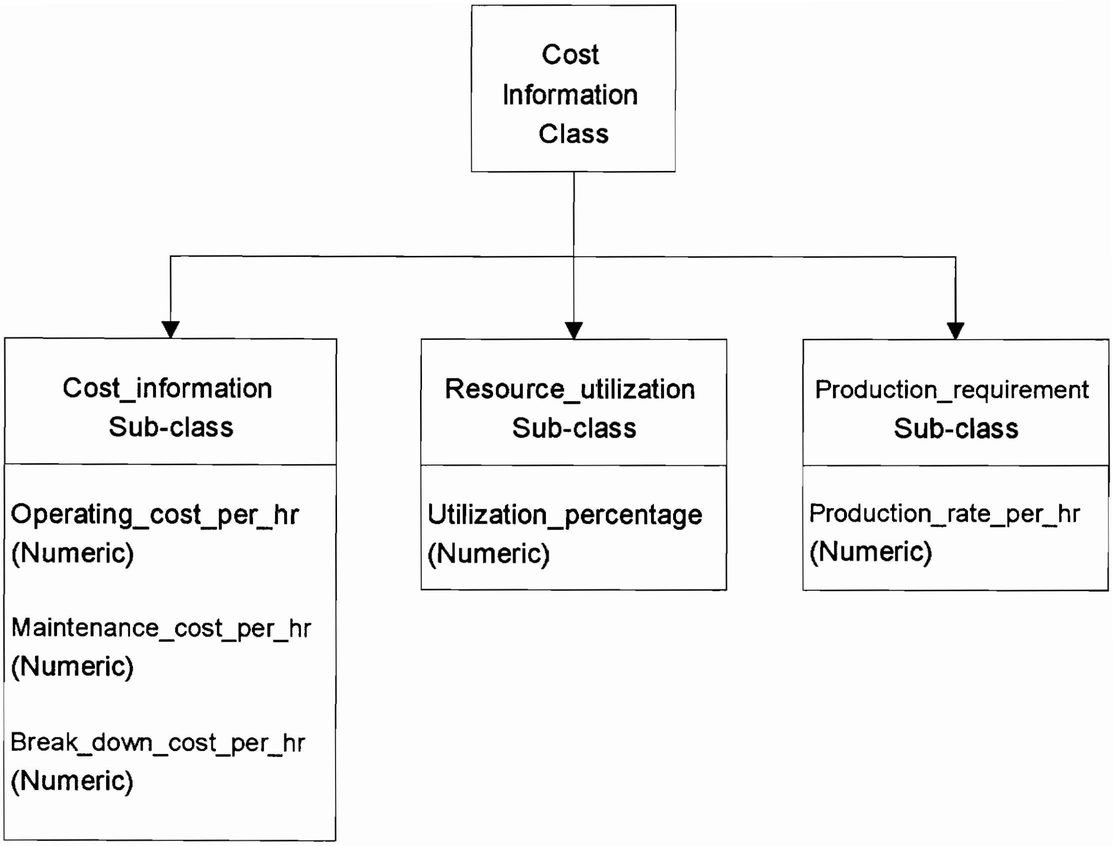


Figure 3.10. Cost Information Class

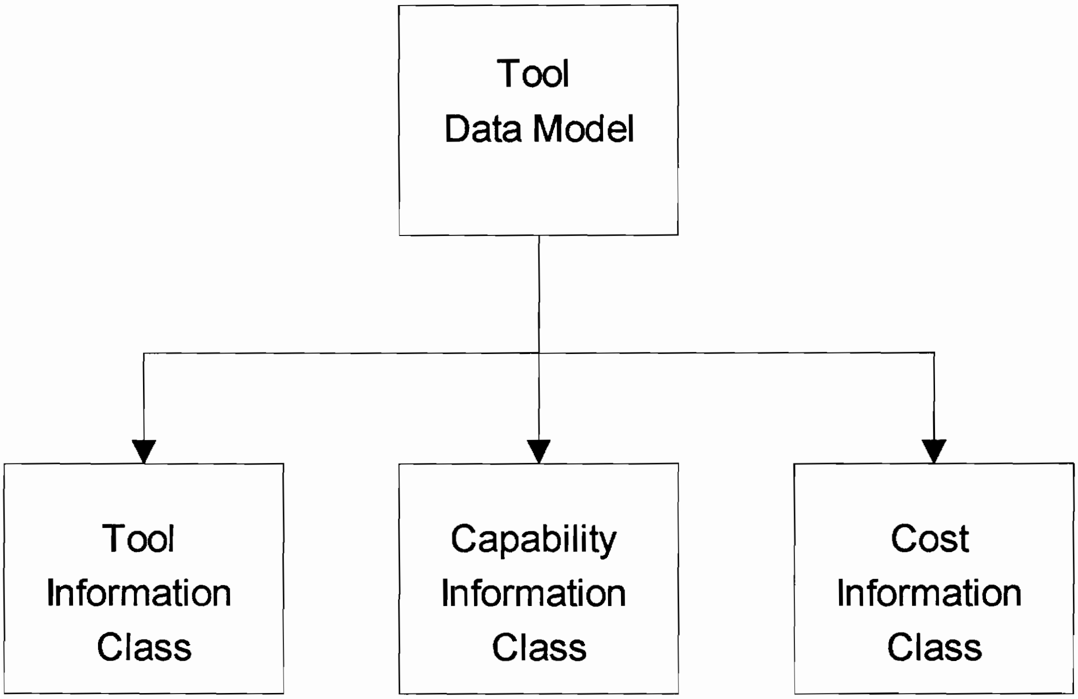


Figure 3.11. Tool Data Model

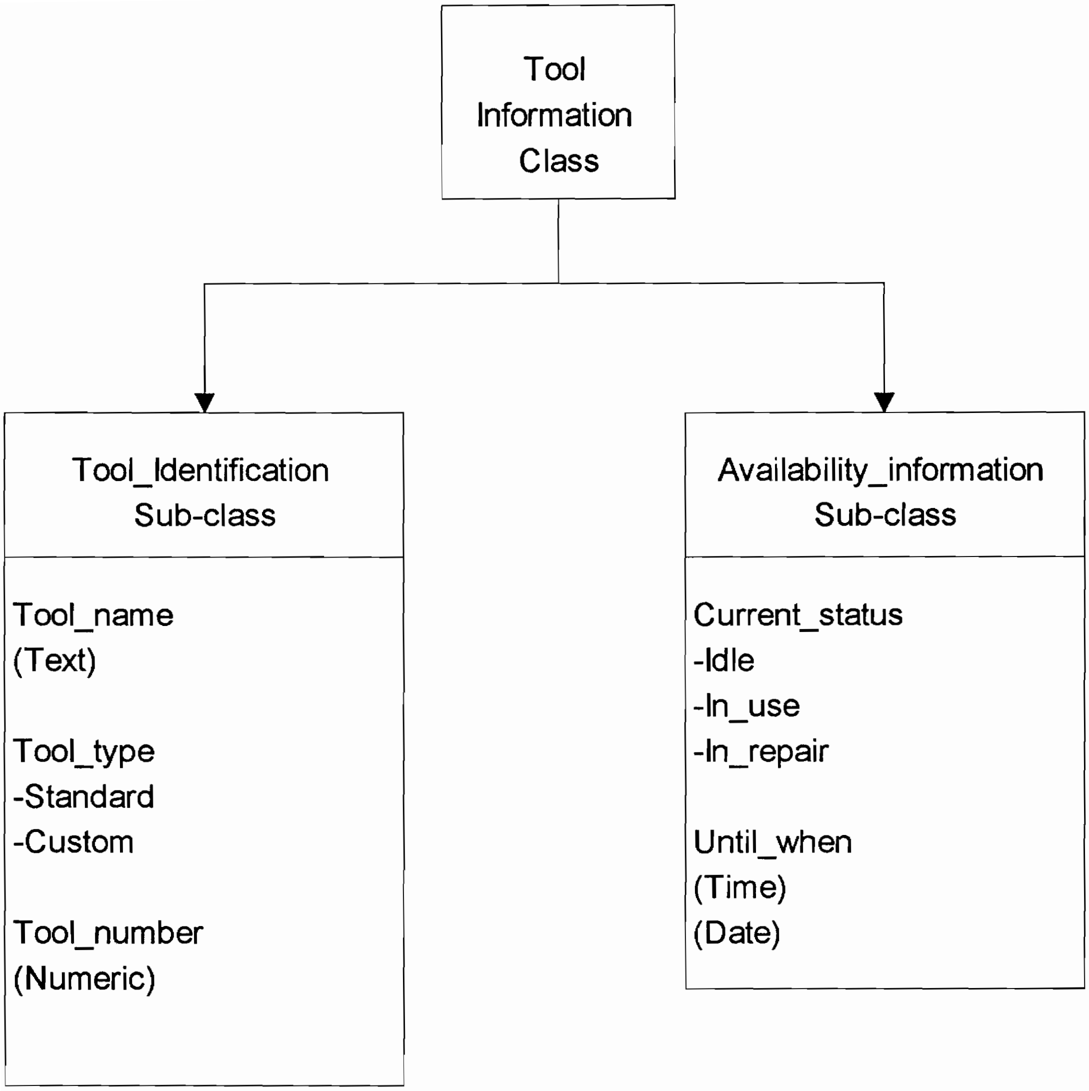


Figure 3.12. Tool Information Class

includes description data identifying the tool instance by name (e.g. Twist Drill), type (e.g. Standard), and drill number. Availibility_information sub-class provides the status of the tool instance (e.g. in_use) and the time when the status is estimated to change (e.g. 12:00 a.m. 9/8/94).

Capability Information Class

Tool Capability Class includes Physical_dimension, and Capability_information sub-classes (Figure 3.13). Data in the physical_dimension sub-class describes type of process each tool instance is capable of performing (e.g. drilling) as well as the tool geometry and tool material. In addition, it describes the diameter and length of the tool instance as well as any other variable that distinguishes a certain family of tools from another. The type of spindle required is also included in this class. Data in the Capability_information sub-class provides information on a tool's accuracy.

Cost Information Class

The Cost Information Class (Figure 3.14) is composed of two sub-classes of data that is used by the Process Plan Evaluation Module: Cost_information and Resource_utilization. The Cost_information sub-class provides data on the operating cost and the maintenance cost of the tool instance. In addition, the number of operation hours before maintenance is also included in this class. The Resource_utilization sub-class provides a utilization rate of the tool instance.

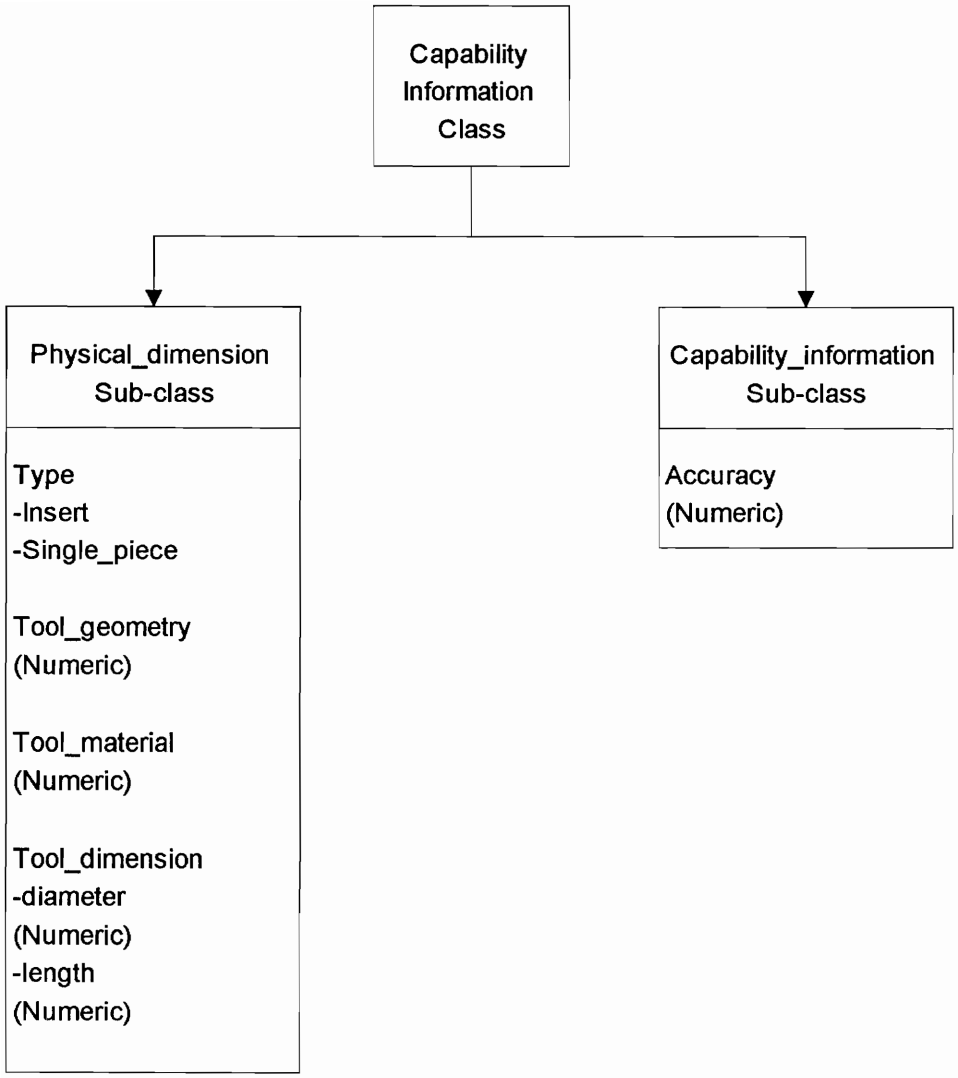


Figure 3.13. Capability Information Class

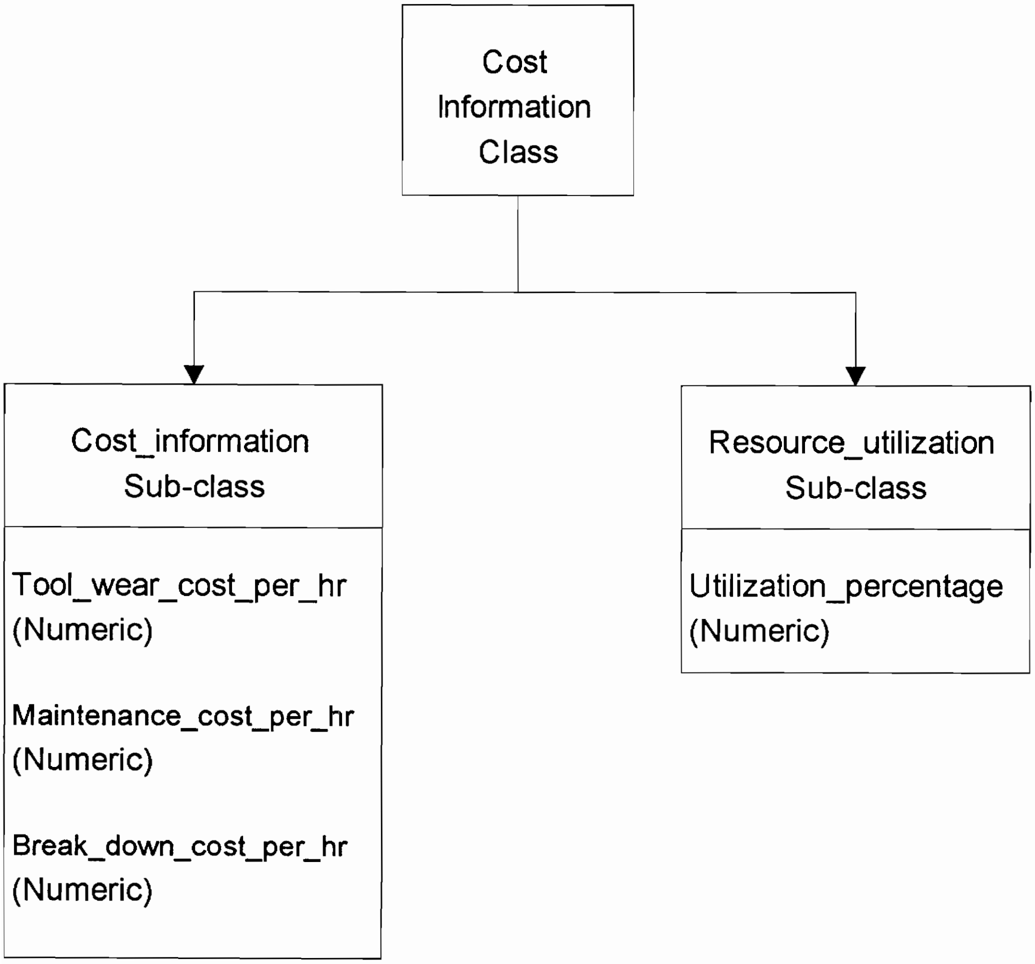


Figure 3.14. Cost Information Class

3.2.3 Process Plan Representation Structure

In developing the appropriate representation structure of process plans, the most important data requirements have been identified as: (1) Precedence constraint relationship among tasks and (2) Hierarchical relationship among process planning activities. The process plan representation structure[5] is shown in Figure 3.15. The process plan is the main structure linked to one or more setup cluster(s). Each setup cluster is composed of a setup with a fixture on a machine along with one or more operation clusters. An operation cluster is one or more operation(s), with each operation defined as a machining operation with the use of a tool.

Chapter 4 demonstrates the automation of this methodology with a case study and discusses the implementation details of the developed software.

---

[5] Based on C. S. Chen's process plan representation structure presented at Fourth International FAIM Conference [CHEN94a].

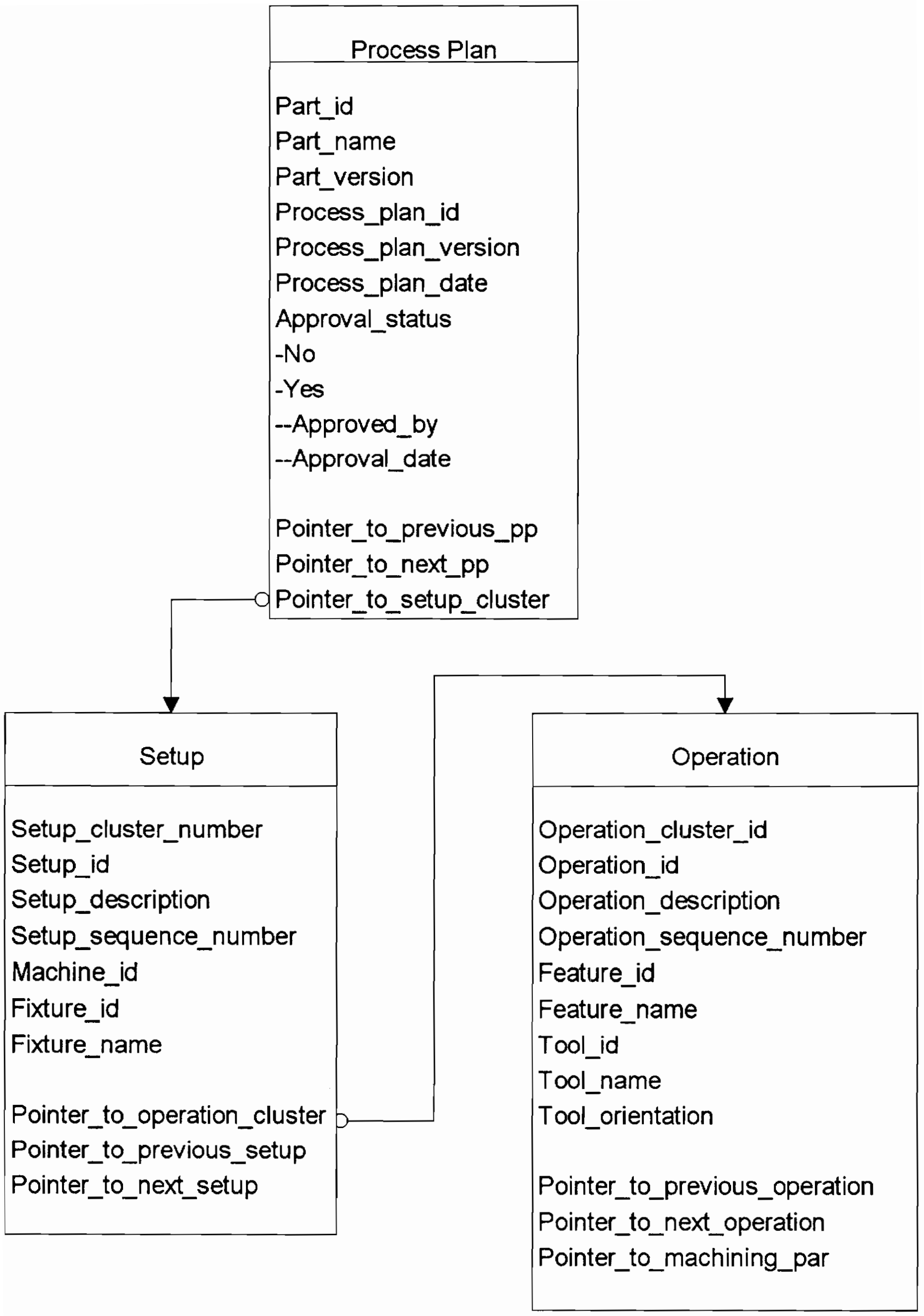


Figure 3.15. Process Plan Representation Structure

CHAPTER 4

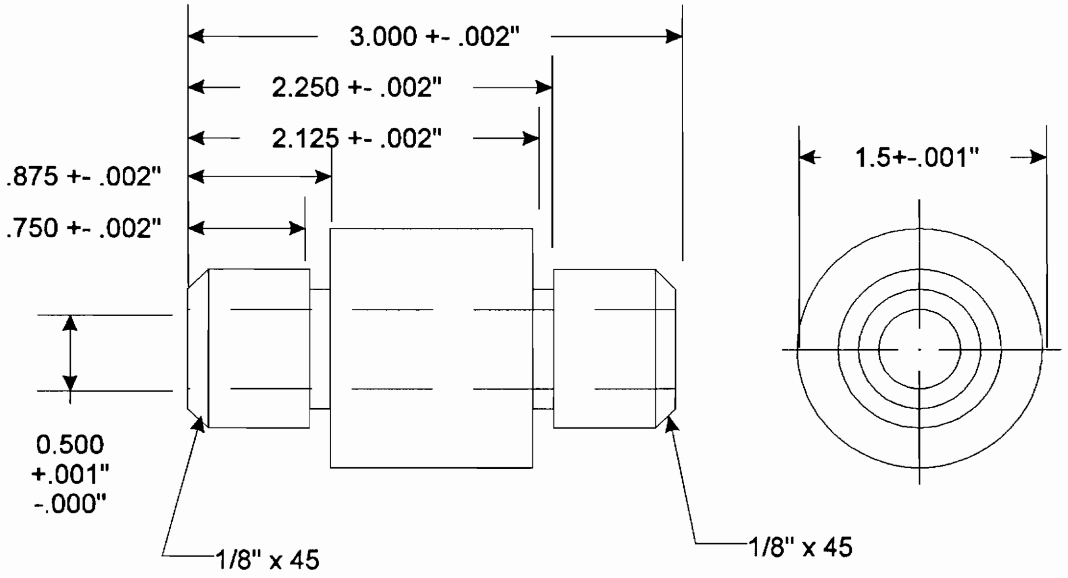
CASE STUDY AND DEMONSTRATION

This chapter demonstrates the proposed methodology through the use of a case study and software implementation. Section 4.1 demonstrates the procedure for process plan generation through the use of a case study on a sample part and a sample manufacturing facility. The software implementation of the procedure for process plan generation is described in section 4.2.

4.1 Case Study

The sample part to be used for the case study is shown in Figure 4.1. It is a rotational part with a material specification of medium carbon steel. The surfaces that need to be generated are identified in Figure 4.2. Based on the part dimensioning, the precedence of the surfaces is illustrated in Figure 4.3. Surface/feature "J", "E", "H", and "C" have no constraints and can be the first surfaces to be generated. Surface/feature "A", "F", "G", and "D" can be generated after "J". Surface/feature "I" can be generated after "H", while surface/feature "B" can be generated after "C".

A simplified manufacturing facility being used for this case study to demonstrate the generation of alternative process plans is shown in Figure 4.4. The facility contains four machines of various types: (1) Turret Lathe, (2) Engine Lathe, (3) Vertical Milling Machine, and (4) Drilling machine. Each machine has a different production capability, utilization rate, and cost information. The turret lathe is a high production, high cost, frequently utilized piece of equipment.



Material: Medium Carbon Steel 1035

Figure 4.1. Sample Part Drawing

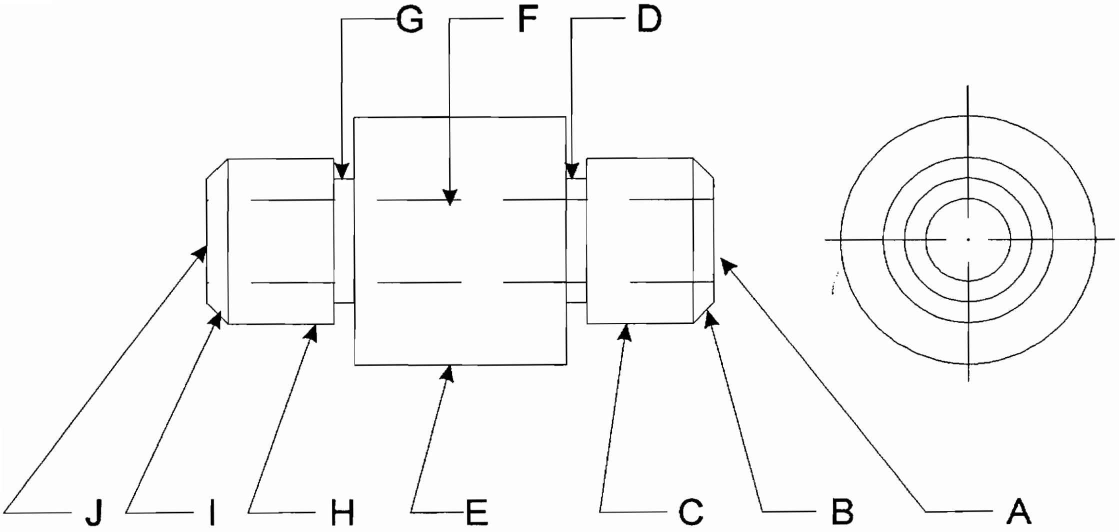


Figure 4.2. Surface Identification Of Sample Part

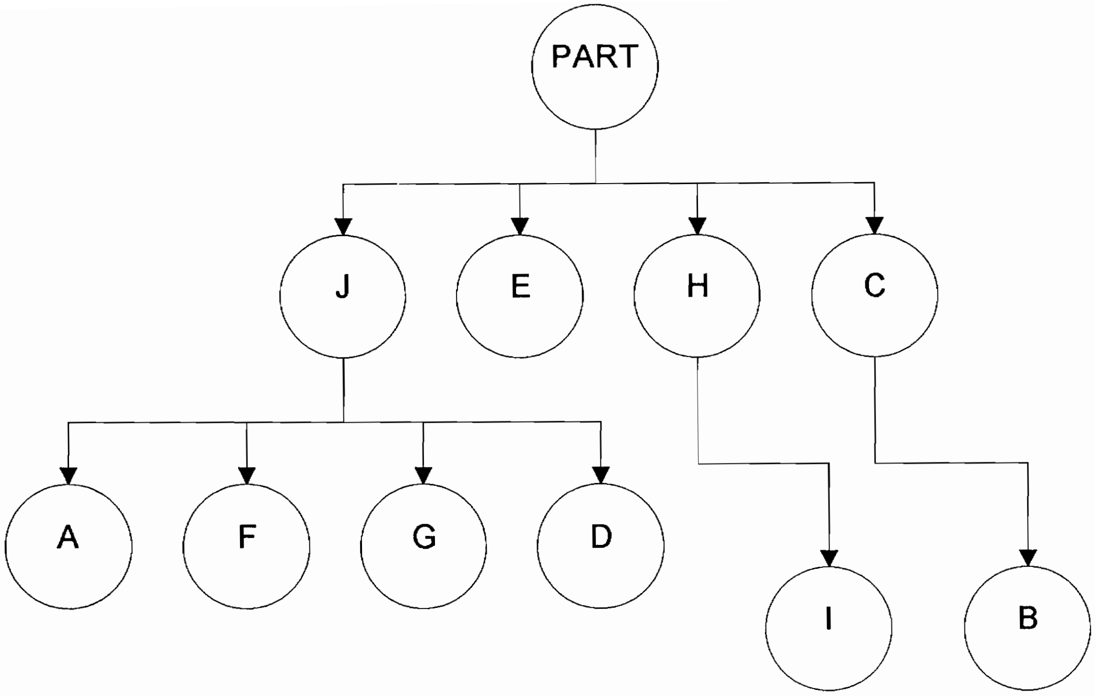


Figure 4.3. Precedence Relationship Of Sample Part

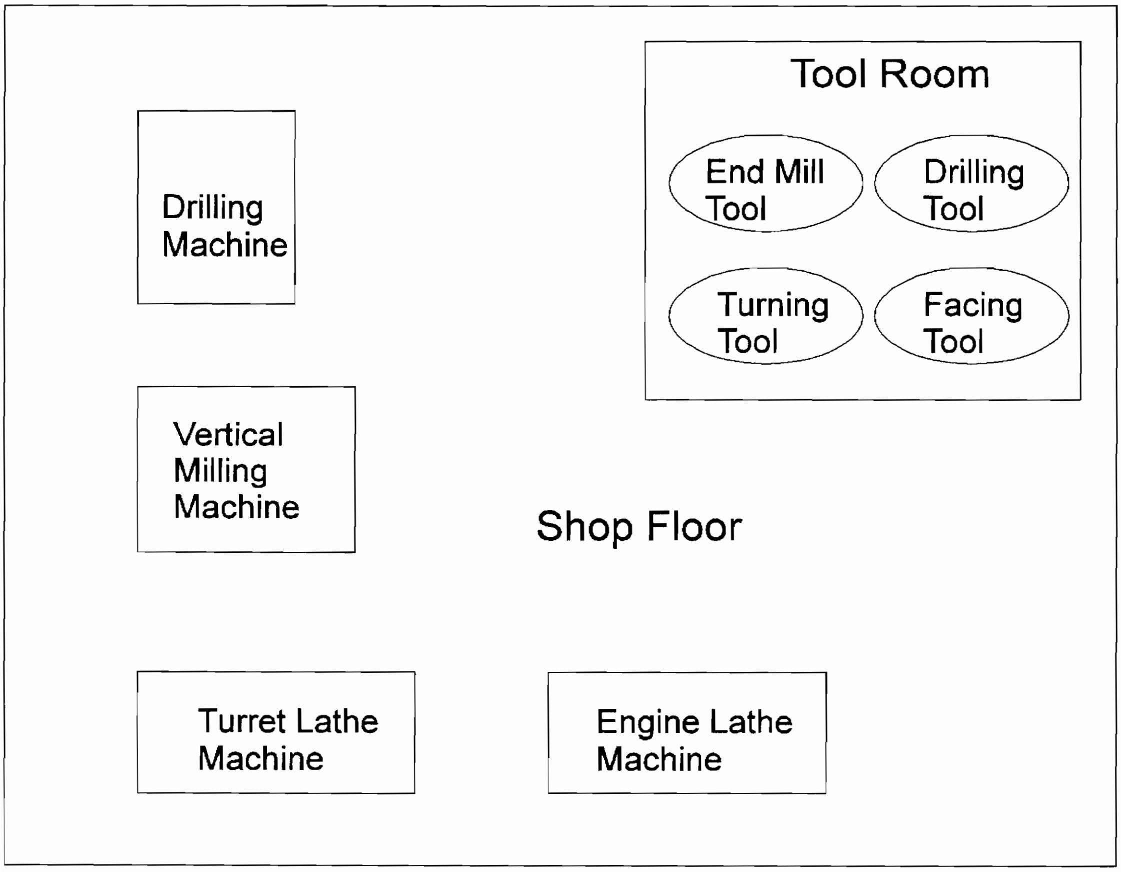


Figure 4.4. Simplified Factory For Case Study

The engine lathe on the other hand is capable of low volume production, is used moderately, and has a low operating cost. The drilling machine is a low volume, low cost, and infrequently used piece of equipment. The vertical milling machine is medium volume, medium cost, and is utilized more often than the drilling machine and less often than the lathe machine.

In this case study, we assume that the manufacturing facility has the right size of tools for the required operations. There are two types of tool materials: (1) high speed steel (HSS) for low production rate and (2) tungsten carbide (WC) for longer tool life (higher production rate) at a higher cost. Generalized data, production rate, utilization rate, and cost information, of the machines and tools shown in Figure 4.5, will be used to select appropriate resources depending on the criteria. Cost information is represented by a cost index, where the higher value indicates higher costs. Production facility status, which provides availability information on the manufacturing resources, is not included in this case study.

4.1.1 Process Selection

Process selection assigns feasible manufacturing processes for each surface/feature. The process assigned depends on whether the part is rotational or non-rotational. The processes assigned to the case study surface/features are listed in Table 4.1. For example, to generate face "J", a facing process and a milling process are assigned. For a circular surface "E", a turning process is assigned. In the case of thru-hole feature "F", a drilling process is assigned.

Turret Lathe Machine	Engine Lathe Machine	Milling Machine	Drilling Machine
Production Capability: High	Production Capability: Low	Production Capability:Medium	Production Capability: Low
Utilization Rate: High	Utilization Rate: Medium	Utilization Rate: Medium	Utilization Rate: Low
Cost_index: 10	Cost_index: 2	Cost_index: 5	Cost_index: 1

Drilling Tool	Facing Tool	Turning Tool	End Milling Tool
HSS	HSS	HSS	HSS
Production Capability: Low	Production Capability: Low	Production Capability: Low	Production Capability: Low
Utilization: high	Utilization: high	Utilization: high	Utilization: high
Cost Index: 5	Cost Index: 5	Cost Index: 5	Cost Index: 5

Drilling Tool	Facing Tool	Turning Tool	End Milling Tool
WC	WC	WC	HSS
Production Capability:Medium	Production Capability:Medium	Production Capability:Medium	Production Capability:Medium
Utilization: high	Utilization: high	Utilization: high	Utilization: high
Cost Index: 8	Cost Index: 8	Cost Index: 8	Cost Index: 8

Figure 4.5. Machine And Tool Data Of Factory Resources

Table 4.1. Process Selection Of Sample Part

Identification	Type	Process 1	Process 2
J	flat surface	face	mill
E	circular surface	turn	
H	circular surface	turn	
C	circular surface	turn	
A	flat surface	face	mill
F	thru-hole feature	drill	
G	groove feature	turn	
D	groove feature	turn	
I	chamfer feature	turn	
B	chamfer feature	turn	

4.1.2 Machine/Tool Selection

The machine/tool pairs assigned to each process of each surface/feature are listed in Table 4.2. A machine/tool pair is assigned to each feasible process based on the surface/feature information (i.e., tolerance, size, overall part size). The feasible machine/tool pair denote a general machine and tool type. For example, in the case study, a feasible machine/tool pair assigned to the facing process of surface "J" is an engine lathe and a facing tool. A feasible machine/tool pair assigned to the milling process of surface "J" is a vertical milling machine and an end mill tool.

4.1.3 Setup Planning

The previous steps identified one or more feasible manufacturing resources that can be used for creating the surface or feature. The objective of setup planning is to select the most appropriate manufacturing resource (depending on criteria). In this case study, two criteria will be used to demonstrate the generation of alternative process plans: (1) minimize cost and (2) minimize time (maximize production rate). Another criterion such as selecting underutilized resources could also be used.

Setup is determined based on tool access directions, precedence constraints, and machine commonality. The setup algorithm first assigns machines and tools that best meet the criteria. Additional process plans are generated by assigning machines and tools not already selected.

Table 4.2. Machine/Tool Selection Of Sample Part

Identification	Process	Machine	Tool
J	face	turret lathe	facing tool
J	face	engine lathe	facing tool
J	mill	vertical mill	end milling tool
E	turn	turret lathe	turning tool
E	turn	engine lathe	turning tool
H	turn	turret lathe	turning tool
H	turn	engine lathe	turning tool
C	turn	turret lathe	turning tool
C	turn	engine lathe	turning tool
A	face	turret lathe	facing tool
A	face	engine lathe	facing tool
A	mill	vertical mill	end milling tool
F	drill	turret lathe	drilling tool
F	drill	engine lathe	drilling tool
F	drill	drilling machine	drilling tool
G	turn	turret lathe	turning tool
G	turn	engine lathe	turning tool
D	turn	turret lathe	turning tool
D	turn	engine lathe	turning tool
I	turn	turret lathe	turning tool
I	turn	engine lathe	turning tool
B	turn	turret lathe	turning tool
B	turn	engine lathe	turning tool

Minimize Cost

The setup planning algorithm for minimizing cost (section 3.1.3) attempts to arrange as few setups as possible to minimize the manufacturing resources used. In situations where two resources can be chosen, the cost minimizing algorithm assigns the resource with the lower cost index.

All the surface/features are first organized into tool approach directions. In the case study, all surface/features of the rotational sample part are grouped into tool approach directions: +X, -X, and Y. Surface/features "J" and "F" are clustered into tool approach direction +X. Surface/features "A" and "F" are clustered into tool approach direction -X. Surface/features "J", "I", "H", "G", "E", "D", "C", "B", and "A" are clustered into tool approach direction Y. Any clusters that violate the precedence are removed, although in this case study, no precedence violations occurred.

The tool approach direction Y has surface/features with the most common machine type (Lathe Machine). Engine Lathe is assigned to be the resource of setup #1 to meet the criteria of minimizing cost. Any surface/features appearing in a setup group are removed from the remaining clusters. In the case study, "J" is removed from the remaining "+X" and "-X" tool approach clusters. "F" is the remaining surface/feature to be grouped in a setup. Tool approach direction "-X" is selected and the machine assigned is the drilling machine because it has the lowest cost index. Since all surface/features belong to a setup group, tool approach direction "+X" cluster is dropped. The tool material type selected for each surface/feature is HSS. The results of this setup generation are shown in Table 4.3.

Table 4.3. First Output Of Setup Planning To Minimize Cost

Identification	Process	Machine	Tool
		Iteration #1	
J	face	engine lathe	HSS facing tool
I	turn	engine lathe	HSS turning tool
H	turn	engine lathe	HSS turning tool
G	turn	engine lathe	HSS turning tool
E	turn	engine lathe	HSS turning tool
D	turn	engine lathe	HSS turning tool
C	turn	engine lathe	HSS turning tool
B	turn	engine lathe	HSS turning tool
A	face	engine lathe	HSS facing tool
		Iteration #2	
F	drill	drilling machine	HSS drilling tool

Machines and tools not selected for the first process plan are considered again on the second pass of the setup generation algorithm. Following the procedure detailed above, the turret lathe is assigned for setup #1 and the engine lathe is assigned for setup #2. Since all feasible equipment assigned to surface/features "E", "H", "C", "A", "G", "D", "I", and "B" has been selected to a process plan, the algorithm ends. The result of the setup planning using the criteria of minimizing cost is shown in Table 4.4.

Minimize Time

The setup planning algorithm for minimizing production time attempts to create as many setups as possible to minimize the processing time at each machine (ideally, the operation time at each machine should be balanced). In this simplified case study, line balance is not considered. Precedence constraint among surface/features is maintained. Following the minimizing time algorithm in section 3.1.3, four machines are first identified. The average number of surface/features per machine is calculated to be 3 (rounded up from 2.5). The drilling machine is identified as having the fewest number ("F", one thru-hole feature) of feasible surface/features assigned. Since there are no precedence violations, the feature "F", the drilling machine, and the WC drilling tool are selected to a setup.

The algorithm is repeated. This time, three machines remain. The average number of surface/features per machine is 3. The vertical milling machine is identified as having the fewest numbers of feasible surface/features assigned (with two, "J" and "A"). Since the number of surface/features does not exceed the average of 3 surface/features per machine and the precedence

Table 4.4. Second Output Of Setup Planning To Minimize Cost

Identification	Process	Machine	Tool
		Iteration #1	
J	face	turret lathe	WC facing tool
I	turn	turret lathe	WC turning tool
H	turn	turret lathe	WC turning tool
G	turn	turret lathe	WC turning tool
E	turn	turret lathe	WC turning tool
D	turn	turret lathe	WC turning tool
C	turn	turret lathe	WC turning tool
B	turn	turret lathe	WC turning tool
A	face	turret lathe	WC facing tool
		Iteration #2	
F	drill	engine lathe	WC drilling tool

constraint is maintained, "J", "A", the vertical milling machine, and the WC facing tool are selected to a setup.

There are still eight surface/features unresolved, so the algorithm is repeated. The engine lathe and the turret lathe machines remain. The average number of surface/features is 4 (rounded up from 3.5). Both machines have the remaining surface/features assigned. The turret lathe, with higher production capability, is chosen to break the tie. All seven of the remaining features from a single tool access direction, "Y" are feasible and do not violate the precedence constraint. Since the value seven is greater than the average number of surface/features per machine, tool commonality is determined. However, all seven surface/features use the same type of tool. Therefore, four surface/features meeting the precedence constraint, "H", "E", "C", and "B" are arbitrarily selected to a setup with the turret lathe and WC turning tools. The algorithm is again reiterated and the remaining 3 surfaces, "I", "G", and "D", are selected to the final setup with the engine lathe and HSS turning tools.

Alternative assignments of machines and tools are performed by considering feasible machine/tools selected the first time. The HSS drilling tool and the turret lathe, selected over the engine lathe because of its high production capability, are assigned to feature "F". Surfaces "J" and "A" are also assigned with the turret lathe and the WC facing tool. Surfaces "H", "E", "C", and "B" are selected to a setup with the engine lathe and HSS turning tools. Finally, surfaces "I", "G", and "D" are selected to the turret lathe and WC turning tools. The results of the setup generation for minimizing time are shown in Tables 4.5 and 4.6.

Table 4.5. First Output Of Setup Planning To Minimize Time

Identification	Process	Machine	Tool
		Iteration #1	
F	drill	drilling machine	WC drilling tool
		Iteration #2	
J	face	vertical mill	WC end mill tool
A	face	vertical mill	WC end mill tool
		Iteration #3	
H	turn	turret lathe	WC turning tool
E	turn	turret lathe	WC turning tool
C	turn	turret lathe	WC turning tool
B	turn	turret lathe	WC turning tool
		Iteration #4	
I	turn	engine lathe	HSS turning tool
G	turn	engine lathe	HSS turning tool
D	turn	engine lathe	HSS turning tool

Table 4.6. Second Output Of Setup Planning To Minimize Time

Identification	Process	Machine	Tool
Iteration #1			
F	drill	turret lathe	HSS drilling tool
Iteration #2			
J	face	turret lathe	WC facing tool
A	face	turret lathe	WC facing tool
Iteration #3			
H	turn	engine lathe	HSS turning tool
E	turn	engine lathe	HSS turning tool
C	turn	engine lathe	HSS turning tool
B	turn	engine lathe	HSS turning tool
Iteration #4			
I	turn	turret lathe	WC turning tool
G	turn	turret lathe	WC turning tool
D	turn	turret lathe	WC turning tool

4.1.4 Operation Sequence Planning

Sequencing the setups is done by precedence relationship and machine type commonality. Using the results of the setup generation for minimizing cost, setup group of tool approach direction "Y" is sequenced first (because of precedence) followed by tool approach direction "-X". Operation sequencing within each setup is determined by precedence and tool commonality. In the case study, the surface/feature sequence of the first setup is: "J", "A", "E", "H", "C", "G", "D", "I", "B". The second setup consists of a single operation on "F". The result of operation sequence planning for setup 1 and setup 2 is shown in Table 4.7 and Table 4.8.

4.1.5 Process Plan Evaluation

At this stage, the user should be able to look at the output process plan and make refinements and/or adjustments. In the case study, we have demonstrated the process by which alternative process plans can be generated by the use of different criterion. The manner in which manufacturing resource information is attached to each feature allows an additional dimension of control by the user to select a specific feasible manufacturing resource. This module should be able to provide cost, utilization rate, and estimated production rate, based on the selected machines and tools in the process plan.

4.2 Software Demonstration

A software program was developed to demonstrate and validate the CAPP Module and the manufacturing resource representation scheme. The software uses the sample part introduced in section 4.1. The software

Table 4.7. Operation Sequencing Of Setup 1 For Minimizing Cost

Setup: 1		Machine: Engine Lathe	
Operation #	Surface	Process	Tool
10	J	Face	HSS Facing Tool
20	A	Face	HSS Facing Tool
30	E	Turn	HSS Turning Tool
40	H	Turn	HSS Turning Tool
50	C	Turn	HSS Turning Tool
60	G	Turn	HSS Turning Tool
70	D	Turn	HSS Turning Tool
80	I	Turn	HSS Turning Tool
90	B	Turn	HSS Turning Tool

Table 4.8. Operation Sequencing Of Setup 2 For Minimizing Cost

Setup: 2		Machine: Drilling Machine	
Operation #	Face	Process	Tool
10	F	Drill	HSS Drilling Tool

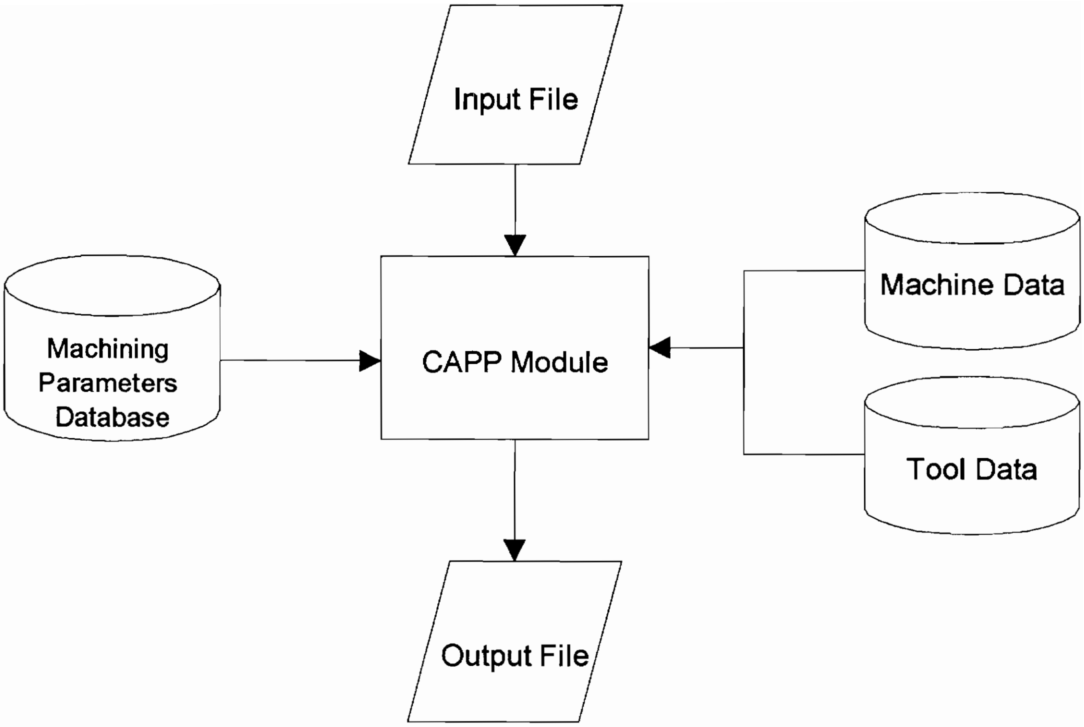


Figure 4.6. Software Architecture

architecture is shown in Figure 4.6. The main components of the implemented software are: (1) Input File, (2) CAPP Module, (3) Manufacturing Resource represented by the Machine Data and the Tool Data, (4) Machining Parameters Database and (5) Output File. The following subsections discuss specific implementation details of the software components. Section 4.2.1 discusses the input file. Section 4.2.2 discusses the CAPP Module. 4.2.3 discusses the representation of the manufacturing resources. 4.2.4 discusses the format of the machining parameters database. 4.2.5 discusses the details of the output file.

4.2.1 Input File

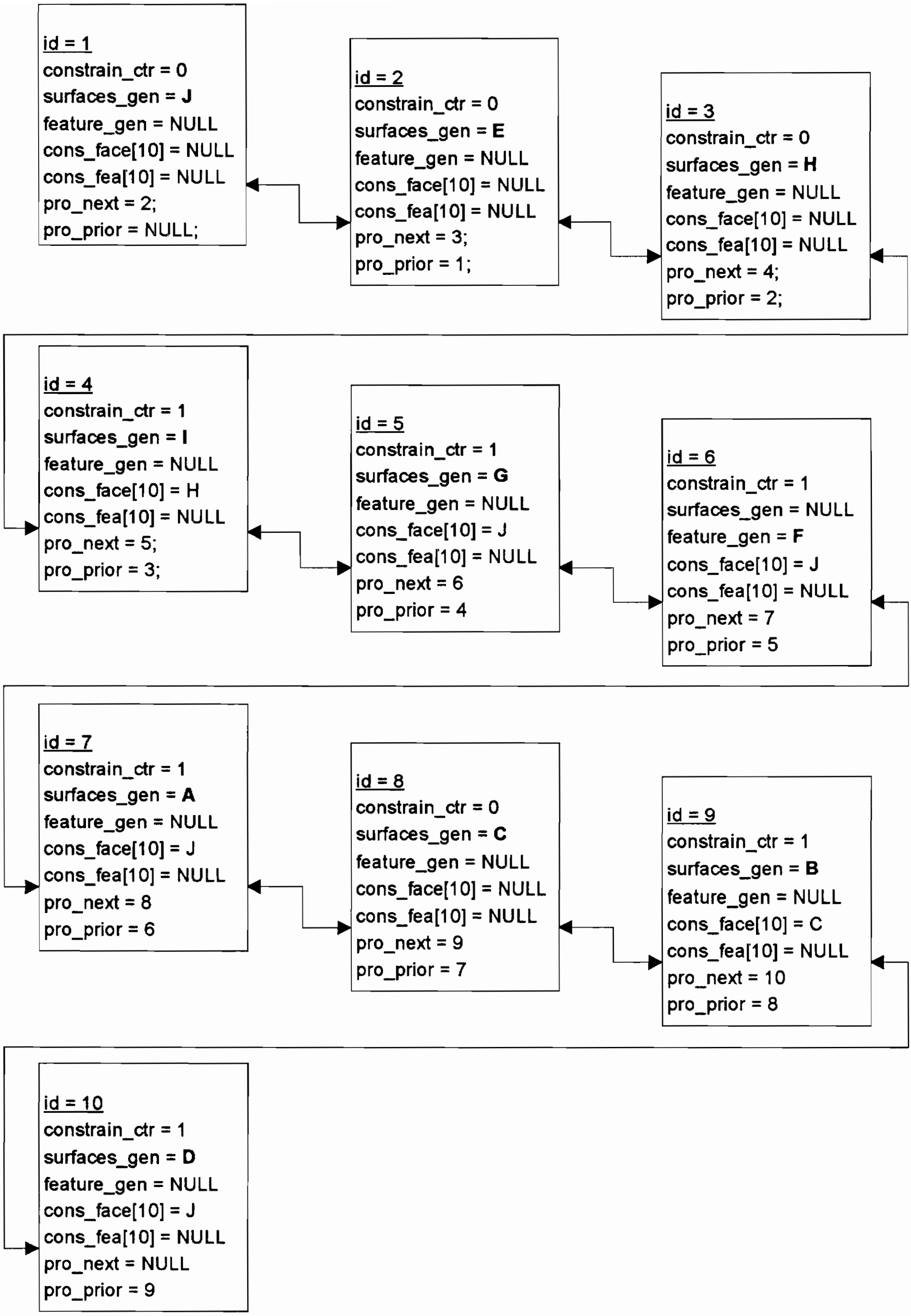
The input file to the implemented software is a sequence list. The structure of the sequence list is a doubly linked list (Figure 4.7). Each feature and/or surface is identified by its own identification (id) and its type pointer (surface_gen/feature_gen) along with the number of constraints (constraint_ctr) and constraint pointer (cons_face/cons_fea). The feature sequence list for the sample part is shown on Figure 4.8. The list is composed of 10 links, where each link represents a surface or a feature that needs to be generated.

4.2.2 CAPP Module

The main components within the CAPP Module follow the illustration shown in Figure 3.1. Detailed work flow procedure of the CAPP Module follows the illustration shown in Figure 3.2. The menu system is implemented in the C programming language. The various routines for creating the process plan are implemented in C++. The process selection and machine/tool pair selection involves linking process and machine/tool information onto the links of the sequence list. The setup planning utilizes a search routine and uses temporary

```
INT id

INT constrain_ctr;

FACE POINTER*  surfaces_gen;

FEATURE POINTER*  feature_gen;

FACE POINTER*   cons_face[10];

FEATURE POINTER*   cons_fea[10];

PROCESS_PLAN*   pro_next;

PROCESS_PLAN*   pro_prior;
```

Figure 4.7. Feature Sequence List Data Structure



id = 1
constrain_ctr = 0
surfaces_gen = **J**
feature_gen = NULL
cons_face[10] = NULL
cons_fea[10] = NULL
pro_next = 2;
pro_prior = NULL;

id = 2
constrain_ctr = 0
surfaces_gen = **E**
feature_gen = NULL
cons_face[10] = NULL
cons_fea[10] = NULL
pro_next = 3;
pro_prior = 1;

id = 3
constrain_ctr = 0
surfaces_gen = **H**
feature_gen = NULL
cons_face[10] = NULL
cons_fea[10] = NULL
pro_next = 4;
pro_prior = 2;

id = 4
constrain_ctr = 1
surfaces_gen = **I**
feature_gen = NULL
cons_face[10] = H
cons_fea[10] = NULL
pro_next = 5;
pro_prior = 3;

id = 5
constrain_ctr = 1
surfaces_gen = **G**
feature_gen = NULL
cons_face[10] = J
cons_fea[10] = NULL
pro_next = 6
pro_prior = 4

id = 6
constrain_ctr = 1
surfaces_gen = NULL
feature_gen = **F**
cons_face[10] = J
cons_fea[10] = NULL
pro_next = 7
pro_prior = 5

id = 7
constrain_ctr = 1
surfaces_gen = **A**
feature_gen = NULL
cons_face[10] = J
cons_fea[10] = NULL
pro_next = 8
pro_prior = 6

id = 8
constrain_ctr = 0
surfaces_gen = **C**
feature_gen = NULL
cons_face[10] = NULL
cons_fea[10] = NULL
pro_next = 9
pro_prior = 7

id = 9
constrain_ctr = 1
surfaces_gen = **B**
feature_gen = NULL
cons_face[10] = C
cons_fea[10] = NULL
pro_next = 10
pro_prior = 8

id = 10
constrain_ctr = 1
surfaces_gen = **D**
feature_gen = NULL
cons_face[10] = J
cons_fea[10] = NULL
pro_next = NULL
pro_prior = 9

Figure 4.8. Feature Sequence List Of Sample Part

links to store results of the setup clusters. The operation sequencing module utilizes a sorting routine to organize operations within each setup.

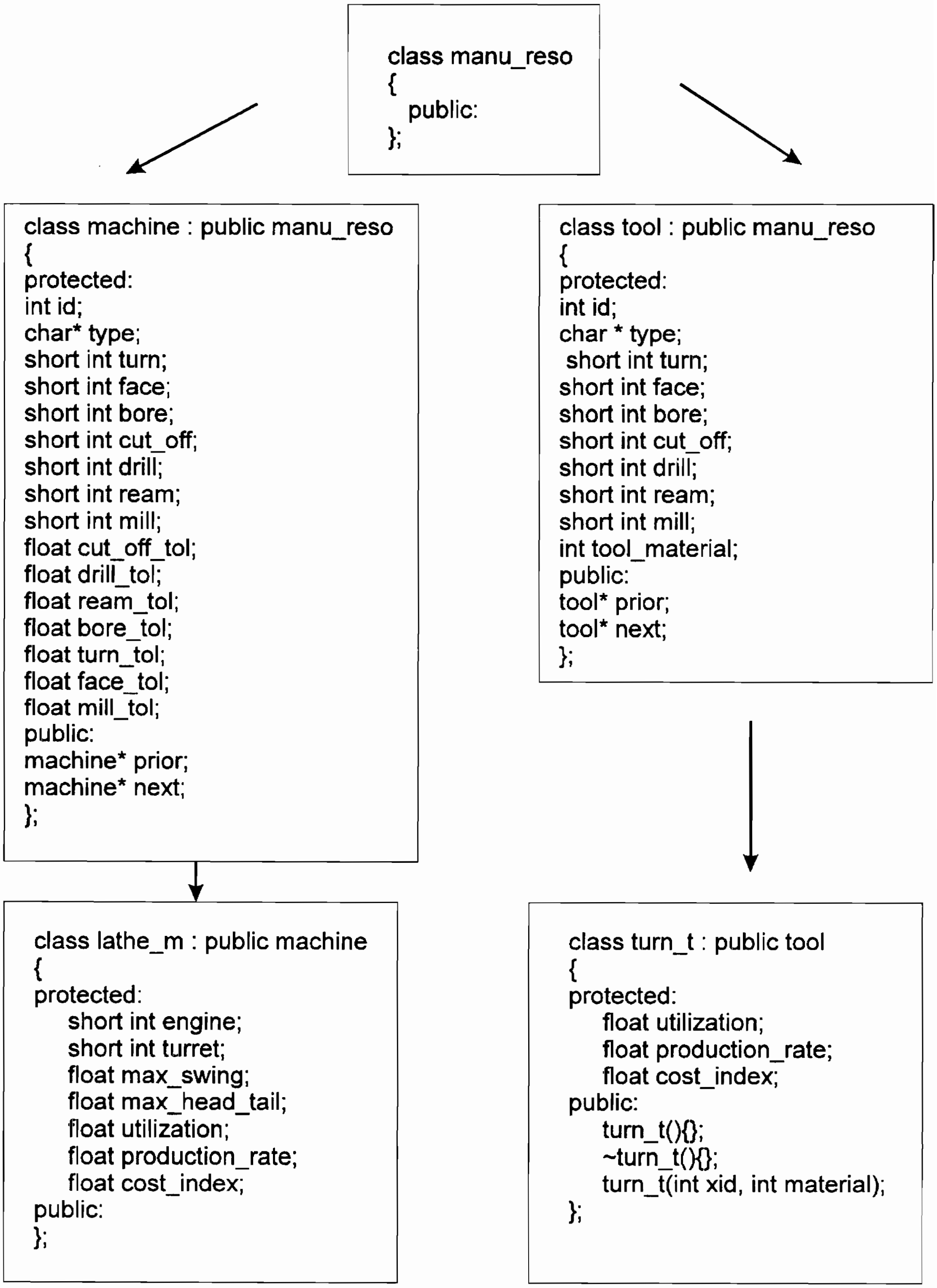
4.2.3 Manufacturing Resource Representation

The representation of manufacturing resources, i.e., machines and tools, is shown in Figure 4.9. Implementation of the data structure is done in an object oriented model. For example, the lathe machine class (class lathe_m) and turn tool class (class turn_t) are derived from their respective general machine class (class machine and class tool). Class machine and class tool are subclasses of the manufacturing resource class (class manu_reso). Lathe_m and turn_t classes contain information specific to the type of machine and tool. In Figure 4.9, class lathe_m contains parameters describing the physical attributes of the lathe. Their parent classes each encapsulate information generic to all types of machines and tools. Also, the class machine contains parameters describing all process capabilities and tolerances of material removal machines.

4.2.4 Machining Parameters Database

A database was created to provide machining parameters, cutting speeds, feeds, rough or finish cut, and type of operations relative to machining operations, workpiece materials, and tool materials. The data is stored in an ASCII file for easy future maintenance and additions. A sample section of the database is shown in Figure 4.10. The format of the database is structured for a parser to scan for: (1) operation type, (2) workpiece material, (3) workpiece material number, and (4) tool material.

The header of each section is the word "operation", followed by the number of operations the data describes. The following line(s) provides



```
class manu_reso
{
  public:
};
```

```
class machine : public manu_reso
{
protected:
int id;
char* type;
short int turn;
short int face;
short int bore;
short int cut_off;
short int drill;
short int ream;
short int mill;
float cut_off_tol;
float drill_tol;
float ream_tol;
float bore_tol;
float turn_tol;
float face_tol;
float mill_tol;
public:
machine* prior;
machine* next;
};
```

```
class tool : public manu_reso
{
protected:
int id;
char * type;
 short int turn;
short int face;
short int bore;
short int cut_off;
short int drill;
short int ream;
short int mill;
int tool_material;
public:
tool* prior;
tool* next;
};
```

```
class lathe_m : public machine
{
protected:
    short int engine;
    short int turret;
    float max_swing;
    float max_head_tail;
    float utilization;
    float production_rate;
    float cost_index;
public:
};
```

```
class turn_t : public tool
{
protected:
    float utilization;
    float production_rate;
    float cost_index;
public:
    turn_t(){};
    ~turn_t(){};
    turn_t(int xid, int material);
};
```

Figure 4.9. Implementation Of Manufacturing Resource Representation

```
OPERATION=2
TURN
FACE
FREE CUTTING STEELS [1100] [1199] [1200] [1299]
hss .015, 135, .007, 190
wc .020, 525, .007, 600
ceramic .007, 1000, .005, 1500
LOW CARBON STEELS [1010] [1025]
hss .015, 120, .007, 160
wc .02, 475, .007, 160
ceramic .007, 1000, .005, 1500
MED CARBON STEELS [1030] [1050]
hss .015, 75, .007, 100
wc .02, 375, .007, 475
ceramic .007, 700, .005, 1000
"HIGH CARBON STEELS [4100] [4199] [4300] [4399]
hss .015, 65, .007, 85
wc .015, 350, .007, 450
ceramic .007, 600, .005, 800
ALLOY STEELS [4100] [4399]
hss .015, 60, .007, 80
wc .015, 350, .007, 400
ceramic .007, 600, .005, 700
```

Figure 4.10. Sample Data Of Machining Parameter Database

identification to the operation(s) the section describes. In this example, the section of the database provides data on turning and facing operations. For each workpiece material such as, "FREE CUTTING STEELS", the material number is represented in one or more ranges. The lower material number range is represented by the first number enclosed in brackets "[" and "]". The next number represents the high material number range. The machining parameter data are represented after each tool material type in the following order: roughing feed, roughing speed, final feed, and final speed.

4.2.5 Output File

The output file is the finished process plan. Besides the information provided in Tables 4.7, 4.8, machining parameters, if found in the database, are also included. Multiple process plans may be generated for different circumstances such as minimizing cost or time. The process plan is stored in an ASCII file for easy retrieval.

Ideally, this file will be the input to the DFM Analysis Module (Section 3.1) where improvements to the design can be performed through configuration analysis and tolerance control. In the future, this information should be integrated to the scheduling software.

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions

This thesis discusses the importance of integrating CAPP within a DFM framework. Most CAPP systems assume feasible designs as input and lack proper design evaluation capabilities. A true generative CAPP system was developed to be an integral part of a DFM methodology [KRIS94]. The DFM methodology is composed of three components: (1) DFM feasibility, (2) process plan generation and (3) DFM analysis.

The major objectives of this research were to develop a CAPP system for machined components and to determine the appropriate representation schemes for production resources. The function of the CAPP system is performed by five process dependent modules: (1) Process Selection, (2) Machine Tool Selection, (3) Setup/fixture Planning, (4) Operation Sequence Planning, and (5) Process Plan Evaluation. The CAPP System has been validated using a case study. Demonstration software has been developed to further address the issue of data representation.

The main contribution of this work is the development of a true generative process planning system that is incorporated into a DFM framework. Process plans generated by the proposed CAPP Module take production facility status into consideration as well as additional criteria for evaluation. The second major contribution of this work is the identification of data required for representing machinery and tooling. The representation structure of process plans will aid in the automation of additional DFM analysis within the framework.

5.2 Future Research

The manufacturing resource representation can be extended to broaden the analysis capabilities of the CAPP system with respect to the production facility. Machining time can also be calculated to determine the machining time at various levels (i.e. operation, setup, part). This information can be useful during process planning analysis.

A fixture planning module is needed to determine the orientation of the part on the machine. The module would require the representation of fixtures and jigs and the part holding capability of machines. The method of fixturing can be addressed by prompting the user to define locating surfaces. This constraint can be included in step 2 of the setup generation algorithm (Section 3.1.3) as part of the precedence constraint. As additional information, such as fixture representation, becomes available to the CAPP Module, the procedure of the setup planning algorithm will need to be reevaluated accordingly.

Although the case study part data used in the demonstration software is PDES/STEP compatible, it was created manually. A feature-based modeler is needed to integrate the "design to manufacturing" concept as discussed in the thesis. This software application will be more complete with the integration of a feature-based modeler into the DFM Application of which the CAPP Module is an integral part.

Finally, future research should also be directed at other processes such as: net-shape, casting, and assembly. An area of interest is identifying critical DFM information and its representation for a process such as net-shaped components. Integration of DFM and process planning and process plan evaluation capabilities for other processes is also worthy of further investigation.

REFERENCES

[ABDO93] Abdou, G., Cheng, R., TVCAPP, Tolerance Verification In Computer-aided Process Planning, International Journal Of Production Research, Volume 31, Number 2, 1993, Pages 393-411

[AGER93] Agerman, E. Et Al, Evolution And Future Perspectives Of CAPP, Annals Of CIRP, Volume 42, Number 2, 1993, Pages 739-751

[ALTI89] Alting, L., Zhang, H., Computer-aided Process Planning: The State-of-the-art Survey, International Journal Of Production Research, Volume 27, Number 4, 1989, Pages 553-585

[ANDE90] Anderson, D. C., Chang, T. C., Geometric Reasoning In Feature-based Design And Process Planning, Computers & Graphics, Volume 14, Number 2, 1990, Pages 225-235

[ANSE93] Anselmetti, B., Bourdet, P., Optimization Of A Workpiece Considering Production Requirements, Computers In Industry, Volume 21, Number 1, 1993, Pages 23-43

[ARAI93] Arai, E., Iwata, K., CAD Systems With Product Assembly / Disassembly Planning Function, Robotics & Computer Integrated Manufacturing, Volume 10, Number 1-2, 1993, Pages 41-48

[BEN-94] Ben-Arieh, D., Kramer, B., Computer-aided Process Planning For Assembly: Generation Of Assembly Operations Sequence, International Journal Of Production Research, Volume 32, Number 3, 1994, Pages 643-656

[BERN94] Bernardi, A., Legleitner, R., Klauck, C., PIM-Skeletal Plan-based CAPP, Computers In Industry, Volume 23, 1994, Pages 87-97

[BOOT92] Boothroyd, G., Dewhurst, P., Knight, W.A., Selection Of Materials And Processes For Component Parts, Proceedings Of The 1992 NSF Design And Manufacturing System Conference, 1992, Pages 255-263

[CATR91] Catron, B. A., Ray, S. R., ALPS: A Language For Process Specification, International Journal Of Production Research, Volume 4, Number 2, 1991, Pages 105-113

[CHAM93] Chamberlain, M. A., Joneja, A., Chang, T. C., Protrusion-features Handling Is A Design And Manufacturing Planning, Computer Aided Design, Volume 25, Number 1, 1993, Pages 19-28

[CHAN90] Chang, T. C., Expert Process Planning for Manufacturing, Addison-Wesley, Massachusetts, 1990

[CHAN91] Chang, T. C., Wysk, R. A., Wang, H. P., Computer-Aided Manufacturing, Prentice Hall, New Jersey, 1991

[CHAN93] Chan, K. C., Nhieu, J., A Framework For Feature-based Applications, Computers In Engineering, Volume 24, Number 2, 1993, Pages 151-164

[CHEN94a] Chen, C. Y., A Framework For Concurrent Process Planning, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 456-464

[CHEN94b] Chen, C. L., LeClair, S. R., Integration Of Design And Manufacturing: Solving Setup Generation And Feature Sequencing Using An Unsupervised-learning Approach, Computer Aided Design, Volume 26, Number 1, 1994, Pages 59-75

[CHOO92] Choobineh, F., Shivani, M., A Dynamic Process Planning And Scheduling Algorithm, Proceedings Of The 1992 Japan - USA Symposium On Flexible Automation, Volume 1, 1992, Pages 429-432

[CHRY85] Chryssolouris, G., Chan, S., An Integrated Approach To Process Planning And Scheduling, Annals Of CIRP, Volume 34, Number 1, 1985, Pages 412-417

[COGU93] Cogun, C., Computer-aided System For Selection Of Nontraditional Machining Operations, Computers In Industry, Volume 22, 1993, Pages 169-179

[COLQ91] Colquhoun, G. J., Baines, R. W., A Generic IDEF0 Model Of Process Planning, International Journal Of Production Research, Volume 29, Number 11, 1991, Pages 2239-2257

[COLQ93] Colquhoun, G. J., Baines, R. W., Crossley, R., A State Of The Art Review Of IDEF0, Int. J. Computer Integrated Manufacturing, Volume 6, Number 4, 1993, Pages 252-264

[DALL76] Dallas, G., Society of Manufacturing Engineer's Tool and Manufacturing Engineers Handbook, McGraw-Hill, New York, 1976

[DELB93] Delbressine, F. L. M., De Groot, R., Van Der Wolf, A. C. H., On The Automatic Generation Of Set-ups Given A Feature-based Design Representation, Annals Of CIRP, Volume 42, Number 1, 1993, Pages 527-530

[DONG91] Dong, Z., Hu, W., Candidate Machining Sequence Generation For Optimal Process Planning Using A Knowledge-Based System, IEEE Pacific Rim Conference On Communications, Computers, And Signal Proceedings, 1991, Pages 569-572

[DONG92] Dong, J., Hyeon, H. J., Parsaei, H. R., A Feature-Based Dynamic Process Planning And Scheduling, Computers & Industrial Engineering, Volume 23, Number 1, 1992, Pages 141-144

[DONG94] Dong, J., Parsaei, H. R., Feature-Based Automated Process Planning A Key To Concurrent Design And Manufacturing, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 723-734

[DUAN93] Duan, W., Zhou, J., Lai, K., FSMT: A Feature Solid-modelling Tool For Feature-based Design And Manufacturing, Computer Aided Design, Volume 25, Number 1, 1993, Pages 29-38

[EVER93] Eversheim, W., Schneewind, J., Computer-aided Process Planning-state Of The Art And Future Development, Robotics & Computer Integrated Manufacturing, Volume 10, Number 1-2, 1993, Pages 65-70

[GU, 94] Gu, P., A Feature Representation Scheme For Supporting Integrated Manufacturing, Computers & Industrial Engineering, Volume 26, Number 1, 1994, Pages 55-71

[GULE93] Gulesin, M., Jones, R. M., Knowledge Representation And Expert System Based Operation Selection In A Process Planning And Fixturing System, International Conference On Applications Of Artificial Intelligence In Engineering, 1993, Pages 565-580

[GUPT88] Gupta, T., Ghosh, B. K., A Survey Of Expert Systems In Manufacturing And Process Planning, Computers In Industry, Volume 11, 1988, Pages 195-204

[GUPT93] Gupta, T., Gali, T. R., Design And Implementation Scheme For An Alternate Process Planning System--A-CAPPS, Computers In Industry, Volume 22, 1993, Pages 15-23

[HERM93] Herman, A., Lawley, M., Lu, S. C., Mattox, D., An Opportunistic Approach To Process Planning Within A Concurrent Engineering Environment, Annals Of CIRP, Volume 42, Number 1, 1993, Pages 545-548

[HORV93] Horvath, L., Horvath, I., Dorozsmai, K., Gabor, Z., Feature-object-based Manufacturing "process" "planning" With The PRODES System, Proceedings Of 1993 IEEE International Conference On Systems, Man, And Cybernetics, Volume 4, Number 4, 1993, Pages 21-26

[IHAR91] Ihara, T., Ito, Y., A New Concept Of CAPP Based On FLAIR Of Experienced Engineers - Analyses Of Decision-Making Processes Of Experienced Pro, Annals Of CIRP, Volume 40, Number 1, 1991, Pages 437-440

[INUI94] Inui, M., Kimura, F., Using A Truth-maintenance System To Assist Product Model Construction For Design And Process Planning, Computer Aided Design, Volume 25, Number 1, 1994, Pages 59-71

[ISHI92] Ishii, K., Miller, R. A., Design Representation For Manufacturability Evaluation In CAD: Beyond Feature-based Design, Computers In Engineering, Volume 1, 1992, Pages 37-44

[JONE91] Joneja, A., Chang, T. C., Search Anatomy In Feature-Based Automated Process Planning, Journal Of Design And Manufacturing, Volume 1, 1991, Pages 7-15

[JOSH86] Joshi, S., Chang, T. C., Feasible Tool Approach Directions For Machining Holes In Automated Process Planning Systems, PED-Concurrent Engineering ASME, Volume 21, 1986, Pages 157-169

[JOVA93] Jovane, F., Results Of The EUREKA FAMOS Research Program: A European Contribution To Manufacturing Technology, Robotics & Computer Integrated Manufacturing, Volume 10, Number 1-2, 1993, Pages 13-19

[KANG93] Kang, T. S., Bartholomew, N. O., Feature Representation And Classification For Automatic Process Planning Systems, Journal Of Manufacturing Systems, Volume 12, Number 2, 1993, Pages 133-145

[KARI92] Karinthi, R., Nau, D., Yang, Q., Handling Feature Interactions In Process Planning, Applied Artificial Intelligence, Volume 6, Number 4, 1992, Pages 389-415

[KAST93] Kastelic, S., Kopac, J., Peklenik, J., Conceptual Design Of A Relational Data Base For Manufacturing Processes, Annals Of CIRP, Volume 42, Number 1, 1993, Pages 493-496

[KEST94] Kesteloot, P., Tilley, S., Sleeckx, E., IDAM: An Architecture For An Integrated DFM/DFA System, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 408-418

[KOJI93] Kojima, T., Hattori, M., Okazaki, T., Concept Sketching For The Next Generation Of Machine Production Information Systems, Computer Integrated Manufacturing Systems, Volume 6, Number 2, 1993, Pages 135-142

[KRIS94] Krishnan, K. K., DFM Methodology and Data Representation Framework for Machined Components, VPI&SU Department of Industrial and Systems Engineering Dissertation Proposal, May 1994.

[KRUT92] Kruth, J. P., Detand, J., CAPP System For Nonlinear Process Plans, Annals Of CIRP, Volume 41, Number 1, 1992, Pages 489-492

[LAKK90] Lakko, T., Mantyla, M., Mantyla, R., Neiminen, J., Sulonen, R. And Tuomi, J., Feature Models For Design And Manufacturing, Proceedings Of The Hawaii International Conference On System Science, Volume 2, 1990, Pages 445-454

[LEND93] Lenderink, A., Kals, H. J. J., The Integration Of Process Planning And Machine Loading In Small Batch Part Manufacturing, Robotics & Computer Integrated Manufacturing, Volume 10, Number 1-2, 1993, Pages 89-98

[LEVI92] Levin, J. B., Dutta, D., On The Effects Of Parallelism On Computer-aided Process Planning, Computers In Engineering, Volume 1, 1992

[LEVI92] Levin, J. B., Dutta, D., Computer-aided Process Planning For Parallel Machines, Journal Of Manufacturing Systems, Volume 11, Number 2, 1992, Pages 79-92

[LIAO94] Liao, T. W., Lee, K. S., Integration Of A Feature-Based CAD System And An ARTI Neural Model For GT Coding And Part Family Forming, Computers & Industrial Engineering, Volume 26, Number 1, 1994, Pages 93-104

[LOGA86] Logan, F. A., Evolutionary Cycle Of An Expert CAPP System, SME Technical Paper, Number 222, 1986, Pages 4/3-4/17

[MANT93] Mantyla, M., Representation Of Process Planning Knowledge For Part Families, Annals Of CIRP, Volume 42, Number 1, 1993, Pages 561-564

[NADI93] Nadir, Y., Chaabane, M., Marty, C., PROCODE-Automated Coding System In Group Technology For Rotational Parts, Computers In Industry, Volume 23, 1993, Pages 39-47

[NAGA94] Nagarur, N. N., Adhikari, S., Integration Of Computer-aided Design And Shape-based Classification And Coding: A Neural Networking Approach, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 427-436

[NARA93] Narang, R. V., Fischer, G. W., Development Of A Framework To Automate Process Planning Functions And To Determine Machining Parameters, International Journal Of Production Research, Volume 31, Number 8, 1993, Pages 1921-1942

[NAU85] Nau, D. S., Chang, T. C., A Knowledge-Based Approach To Generative Process Planning, Symposium Of Computer-Aided Intelligent Process Planning, 1985, Pages 65-71

[NGWE94] Ngwenyama, O. K., Grant, D. A., Enterprise Modeling For CIM Information Systems Architecture: An Object Oriented Approach, Computers & Industrial Engineering, Volume 26, Number 2, 1994, Pages 279-293

[PARK93] Park, J. Y., Khoshnevis, B., A Real-time Computer-aided Process Planning System As A Support Tool For Economic Product Design, Journal Of Manufacturing Systems, Volume 12, Number 2, 1993, Pages 181-193

[QIAO93] Qiao, L., Zhang, C., Liu, T., Wang, H. B., Fisher, G. W., A PDES / STEP-based Product Data Preparation Procedure For Computer-aided Process Planning, Computers In Industry, Volume 21, Number 1, 1993, Pages 11-22

[SALO93] Salomons, O. W., Van Houten, F. J .A .M., Kals, H. J. J., Review Of Research In Feature-Based Design, Journal Of Manufacturing Systems, Volume 12, Number 3, 1993, Pages 113-132

[SANI90] Sanii, E. T., Davis, R. E., Feature-based Distributed Computer Aided Process Planning System, Design Automation 1992 Winter Annual Meetings Of The ASME, 1992, Pages 163-179

[SCHU92] Schulte, R. M., Padmanabhan, S., Devgun, M. S., Feature-driven, Process-based Approach To The Integration Of CAD/CAM In Wireframe Models, International Journal Of Production Research, Volume 30, Number 5, 1992, Pages 1005-1028

[SHAH91] Shah, J., Sreevalsan, P., Mathew, A., Survey Of CAD Feature Based Process Planning And NC Programming Techniques, Computer-Aided Engineering Journal, Volume 8, Number 1, 1991, Pages 25-33

[SING91] Singh, N., Mohantly, B. K., Fuzzy Approach To Multi-Objective Routing Problem With Applications To Process Planning In Manufacturing Systems, International Journal Of Production Research, Volume 29, Number 6, 1991, Pages 1161-1170

[SRIH90] Srihari, K., Emerson, C. R., Petri Nets In Dynamic Process Planning, Computers & Industrial Engineering, Volume 19, Number 1-4, 1990, Pages 447-451

[SSEM93] Ssemakula, M. E., ICAPP: A Process Planning System For The CIM Environment, Computer Integrated Manufacturing Systems, Volume 6, Number 4, 1993, Pages 235-243

[SUBB90] Subbarao, K., Cutkosky, M., Approach Toward Incremental And Interactive Planning For Concurrent Product And Process Design, Winter Annual Meeting Of The ASME, Volume 29, 1990, Pages 1-8

[SWIF94]	Swift, B. F., Ahmad, M. M., Lewis, H. J., Linking CAD To CAPP, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 437-445
[TANG93]	Tang, G., Lan, H. F., Auto-placement Of Semi-graphic Flow Diagrams, Computers In Industry, Volume 21, Number 1, 1993, Pages 51-59
[THAK93]	Thakar, G., Shanker, K., Jain, V. K., An Integrated Process Planning And NC Part Programming System For Rotational Components, Computers In Industry, Volume 21, 1993, Pages 341-357
[TOHR93]	Tohru, I., Misao, O., Yoshimi, I., Evaluation Method Of Complexity And Difficulty Of Difficult To Machine Portions For CAPP Based On FLAIR, Journal Of The Japan Society Of Precision Engineering, Volume 59, Number 3, 1993, Pages 435-440
[TONS94]	Tonshoff, H. K., Baum, T., Ehrmann, M., SESAME: A System For Simultaneous Engineering, Proceedings Of The Fourth International FAIM Conference, 1994, Pages 380-389
[TSAN89]	Tsang, J. P., Brissaud, D., A Feature-based Approach To Process Planning, Computers In Engineering, Volume 1, 1989, Pages 419-430
[VAN 93]	Van Den Molengraaf, J. C., Van Mal, H. H., Wijnia, J., Van Den Molengraaf, J. C., Selection Procedures For Manufacturing Processes For Design Engineers, Robotics & Computer Integrated Manufacturing, Volume 10, Number 1-2, 1993, Pages 57-64
[WACO94]	Waco, D. L., Kim, Y. S., Geometric Reasoning For Machining Features Using Convex Decomposition, Computer Aided Design, Volume 26, Number 4, 1994, Pages 477-489
[WANG86]	Wang, H. P., Wysk, R. A., An Expert System For Machine Data Selection, Computers & Industrial Engineering, Volume 10, Number 2, 1986, Pages 99-107
[WANG88]	Wang, H. P., Wysk, R. A., AIMSI: A Prelude To A New Generation Of Integrated CAD/CAM Systems, International Journal Of Production Research, Volume 26, Number 1, 1988, Pages 119-131
[WANG91]	Wang, H. P., Li, J. K., Computer-Aided Process Planning, Elsevier, Netherlands, 1991
[WANG94]	Wang, M. T., Chamberlain, M. A., Chang, T. C., Manufacturing Feature Extraction And Machined Volume Decomposition In A Computer-integrated Feature-based Design And Manufacturing, Computers In Industry, Volume 23, 1994, Pages 75-86

[YANG92] Yang, H., Lu, W. F., Lin, A. C., A Framework For Using Case-based Reasoning In Automated Process Planning, PED-Concurrent Engineering ASME 1992, Volume 59, 1992, Pages 101-114

[YUT,94] Yut, G. A., Chang, T. C., A Five-Step Object-Oriented Architecture For Process Planning, International Journal Of Production Research, Volume 32, Number 3, 1994, Pages 545-562

[ZHAN92] Zhang, C., Wang, H. P., Li, J. K., Simultaneous Optimization Of Design And Manufacturing, Tolerances With Process (machine) Selection, Annals Of CIRP, Volume 41, Number 1, 1992, Pages 569-572

[ZHAN93] Zhang, H. C., IPPM - A Prototype To Integrated Process Planning And Job Shop Scheduling Functions, Annals Of CIRP, Volume 42, Number 1, 1993, Pages 513-518

[ZHAO93] Zhao, Z., Baines, R. W., Blount, G. N., Definition Of Generic Relationships Between Design And Manufacturing Information For Generative Process Planning, Computer Integrated Manufacturing Systems, Volume 6, Number 3, 1993, Pages 204-212

VITA

Charles Chi-Yu Chiang was born January 13, 1970, in Taipei Taiwan. He is the son of Chieh-Min Chiang and Teh-Yung Fang. He moved to Annandale, Virginia during the summer of 1979. Charles graduated from Falls Church High School in Falls Church, Virginia in 1988. He went on to receive his B.S. in Industrial and Systems Engineering from Virginia Tech in 1992. Upon completion of his Master's thesis under the Manufacturing Systems Engineering Program of the Industrial and Systems Engineering Department at Virginia Tech, Charles hopes to be contributing to society working as a manufacturing engineer.