ABSTRACT

FEBRANSYAH, ADE**.** *A Feature-based Approach to Automating High-Level Process Planning.* (Under the direction of Professors Ezat T. Sanii and Denis R. Cormier)

High-level process planning plays an important role in determining candidate process domains at the configuration design stage. Changing the processes domains later increases the product development cycle and the product development cost. Therefore, determining the most appropriate manufacturing processes at the beginning stages of the design process becomes critical. However, high-level process planning systems have traditionally lacked integration of design synthesis and design evaluation.

The main objective of this research has been to develop a FEature-Based Design And Process Planning (FEBDAPP) system that helps designers decide whether or not the design is worth pursuing by providing manufacturing advice to designers during the design process. In order to achieve the main objective, the following tasks have been accomplished: (1) developed a hybrid system incorporating design by feature and feature recognition approaches capable of reducing the complexity of feature recognition algorithms without sacrificing flexibility in creating a part design, (2) developed a comprehensive set of feature mapping algorithms capable of transforming a primary part representation into a secondary part representation that is required as input to downstream applications, and (3) developed a CAD-based interface capable of integrating a current CAD system with a high-level process planning system.

This research contributes significantly to the availability of early design tools that enhance and at the same time shorten the design process cycle. The implementation of feature technology in this research will support the development of applications such as tooling cost estimation and manufacturability analysis. Finally, the FEBDAPP system is intended to be an effective concurrent engineering tool that bridges the gap between design and manufacturing.

A FEATURE-BASED APPROACH TO AUTOMATING HIGH-LEVEL PROCESS PLANNING

by

ADE FEBRANSYAH

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

INDUSTRIAL ENGINEERING

Raleigh

2001

APPROVED BY:

Dr. Michael G. Kay

Dr. Fen Wu

Dr. Ezat T. Sanii
Co-chair of Advisory Committee

Dr. Denis R. Cormier
Co-chair of Advisory Committee

BIOGRAPHY

Ade Febransyah was born on February 2, 1965 in Jakarta, Indonesia. He attended Bandung Institute of Technology and received a Bachelor of Science degree in Mechanical Engineering in 1988. Following his graduation, he spent three years in industrial sector mostly working as a mechanical engineer. In 1992, he pursued a Master's degree in mechanical engineering at Oklahoma State University. After receiving his degree in 1994, he went back to Indonesia and worked as a research associate in the energy industry. In 1997, he started his Ph.D. study in the Department of Industrial Engineering at North Carolina State University.

ACKNOWLEDGEMENTS

I would like to thank the Co-Chairs of my Advisory Committee, Dr. Ezat T. Sanii and Dr. Denis R. Cormier for their advice, assistance, patience, and encouragement during the course of this research. I am deeply thankful for being introduced to the fascinating area of research pursued for my dissertation. I would also like to thank Dr. Michael G. Kay and Dr. Fen Wu for their roles on my advisory committee. Also, I would like to thank Dr. Kristin A. Thoney who represented the Graduate School in my Ph.D. oral examination.

Special thanks are due to Dr. Ronald E. Giachetti for providing me the Material and Manufacturing Process Selection (MaMPS) software. Without it, this dissertation would have taken a different and less efficient approach.

I dedicate this dissertation to my parents who always reminded me the importance of education. Their endless encouragement is very much appreciated.

Finally, I would like to express my most sincere thank to my wife, Rita, for her love, understanding, sacrifices, and support. I am really indebted to my daughters, Megan and Adia, for their love, cheerfulness, and their understanding of my not being able to spend a lot of time with them, especially during the final stage of this research.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Background

It is widely believed that a large percentage of product costs are determined during engineering design. One analysis shows that at least 70% of the product's final costs are set by the decisions made during the design stage [Ullman 1992]. This implies that once a final design is completed, there is little opportunity to reduce the product cost since a significant portion of its cost is already determined. Based on this fact, companies strive to consider economic factors as early as possible in the design process.

In practice, the so-called *over the wall approach* is still commonly used, where designers work on product design and then throw the design over the wall to the manufacturing engineers [Boothroyd 1994]. The design engineers are only responsible for designs of products that can function properly without considering the manufacturability of their designs. A *back and forth* process usually occurs when the manufacturing engineers find that the designed products cannot be manufactured or are difficult to manufacture and thus they have to send the designs back to the design engineers for revision. This is a lengthy and expensive process that unnecessarily extends the product development cycle.

Lack of manufacturing knowledge among designers is considered the main cause of the current inefficient product development process. An experienced designer repeatedly chooses the manufacturing processes he or she is familiar with, without considering alternative processes and materials. For example, a designer with experience in die-casting tends to choose die-casting again and again for most design problems. In the current manufacturing industry where product designs change very rapidly and products are often made in small batches, this habit-driven mentality could be harmful to the process of innovation [Smith 1999].

To avoid this problem, many companies have adopted some form of concurrent engineering. One important element of concurrent engineering is *Design for Manufacture* (DFM). DFM aims to bridge the gap between the design and manufacturing stages. The ultimate goal of DFM is that once a design is completed, it is ready for manufacture. But the DFM analysis typically focuses on a specific manufacturing process domain, i.e. machining, die-casting, injection molding, stamping, etc. It is assumed that certain candidate processes have already been designated for a given part design before the DFM analysis can be made. The other fact is that DFM guidelines are employed when the part is in the detailed design stage. One important task preceding design for manufacture analysis is the selection of manufacturing processes at the preliminary stages of design.

Figure 1.1 shows the typical steps taken in a concurrent engineering environment using Design for Manufacture and Assembly (DFMA) techniques [Boothroyd 1994]. As designers complete the DFA analysis, the sketch of each part is already made. Before the designers go further into detailed design, they need to make decisions as to what processes and materials should be used to produce the part. This is not an easy task since there are numerous combinations of processes and materials that need to be considered.

Fortunately, there are material and process relationships that can eliminate some materials or processes from consideration. Dixon and Poli [Dixon 1995] propose *either the material first approach* or *the process first approach* to select process candidates. Both approaches should end up with the same result. In practice, a certain process is typically selected because of cost constraints, so an early cost estimate should be made in order to select the most economical process.

1.2 Process Planning

The term *process planning* is widely defined as a family of planning tasks that must be completed before a designed product can be manufactured [Shah 1995]. Process planning systems can be grouped into two levels: low-level process planning and high-level process planning. In general, low-level process plans contain detailed instructions on how to create the part after the detailed design has been completed. For instance, in the metal cutting process domain, low-level process plans include operation sequencing, fixture selection, machine selection, tool selection, and cutting parameters selection. A lot of effort has been made in the area of low-level process planning systems, and most of the research in this area has focused mainly on machined parts [Alting 1989]. A wide range of systems, from computer-aided process planning to fully automated process planning systems, has been developed.

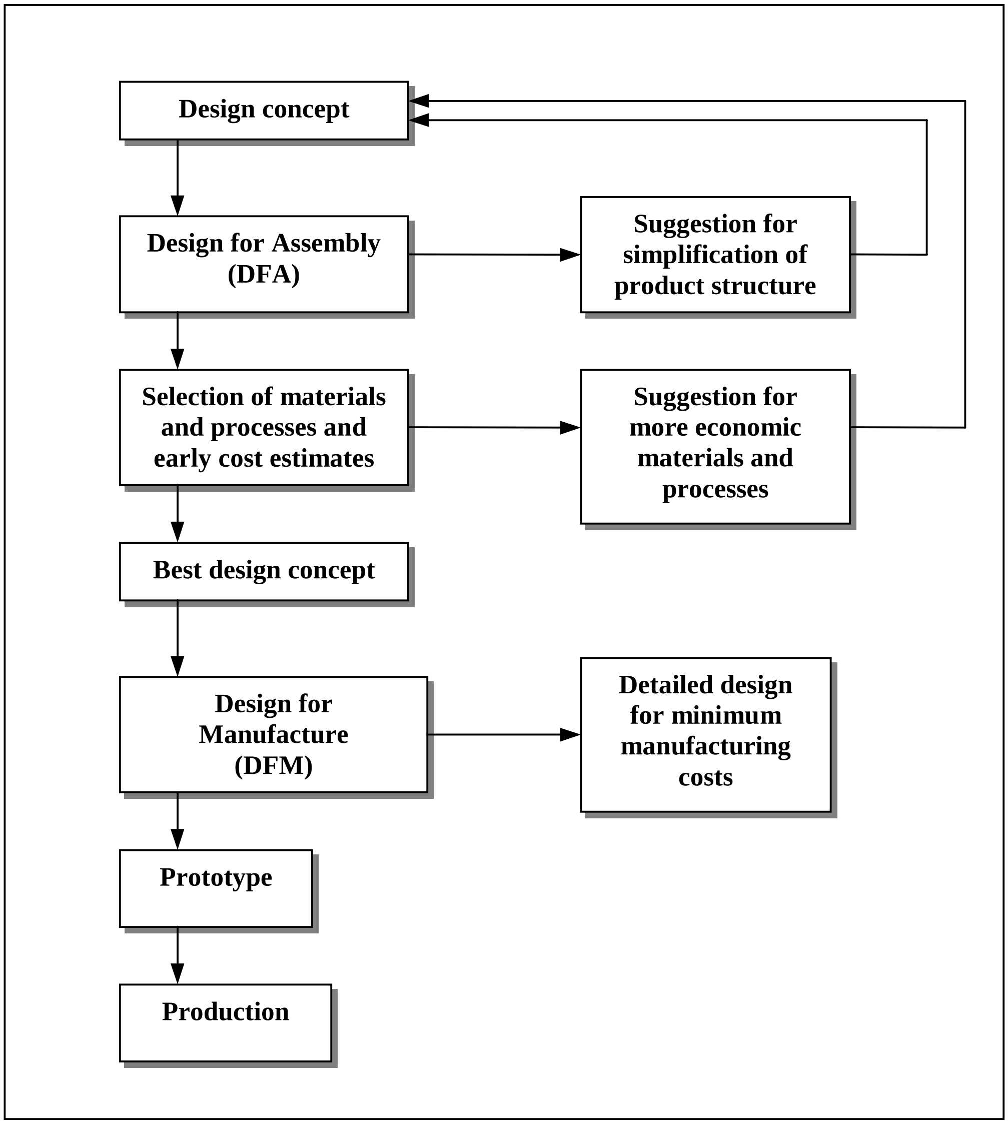


Figure 1.1 Typical Steps in Concurrent Engineering [Boothroyd 1995]

On the other hand, the main goal of high-level process planning is principally to select the manufacturing processes and materials that can be used to form the part at the early stage of design. The early stage of design is associated with the stage where products have been conceptually designed. At this point, it is helpful to define what is meant by conceptual and detailed design.

1.1.1 Conceptual Design

Practically speaking, a conceptual design focuses on product structure rather than individual parts. At the conceptual stage, a product designer usually transforms his or her ideas by sketching a product configuration consisting of each individual part. At this stage, the emphasis is on the functionality of the configuration that will likely satisfy the product's functional requirements. After a product has been conceptually defined, then designers create preliminary designs for its constituent components. At this stage, depending on the level of expertise of the designer, a specific combination of processes and materials is already on the designer's mind. In some instances, process selection is not too challenging during conceptual design. The more challenging task is in determining how to obtain a reliable cost estimate for each part at the early design stage. The cost estimate affects the decision making process that determines whether a part will be made or purchased, or whether a detailed design is worth developing. As previously stated, the process and material selection is an easy task if the designer ignores any analysis and just selects the same process repeatedly. However, if one would like to consider all possible processes and materials, this task becomes much more difficult to properly accomplish.

1.1.2 Detailed Design

Based on the product configuration that emerges from the conceptual design stage, designers elaborate on each individual part at the detailed design stage. The purpose of detailed design is to modify the conceptual design into a shape that is compatible with the selected process and to establish detailed specifications. In an effort to obtain good detailed designs, DFM techniques are introduced. Since DFM guidelines are specific to process domains, a list of process candidates should be provided prior to any detailed design activity. It is too late if a designer has not selected a process domain for each part prior to working on a detailed design. In good design practice, the designers should avoid making changes to the process domain after completing the detailed design. For example, suppose that injection molding is selected as the process domain during the preliminary part design, but that when the designer tries to elaborate on detailed specifications, it is found that injection molding is not suitable. The designer must then choose another process domain, such as die-casting, and the whole process is repeated. This practice can be harmful and may create a highly inefficient product development cycle [Smith 1999].

After a designer determines the process candidates for a preliminary part design, DFM techniques for the specific process domains are applied. If these DFM results suggest that the preliminary part design is difficult to manufacture due to certain features, then the design is revised in accordance with DFM techniques.

There are relatively few research activities related to high-level process planning. The lack of work in this area is perhaps due to the fact that the decision making activity of process and material selection is trivialized by adopting the past decisions or hastily selecting a seemingly obvious candidate proposed by an experienced designer. However, it should be noted that an experienced designer tends to repeatedly select the same processes and materials that he or she is familiar with. In fact, there are many process and material combinations that should be considered for producing a part, but most are often ignored.

1.3 Current Process Planning Systems

Most work on process planning systems pertains to low-level process planning. Most of the research in the field of process planning focuses on the low-level process planning, where a detailed process plan is automatically generated. Numerous Computer-Aided Process Planning (CAPP) systems have been developed that deal with a single process domain such as machining or forming. Only a handful of research activities have addressed process planning at the higher level. In Chapter 2, a review of high-level process planning systems is presented.

Figure 1.2 depicts the current practice in design. Designers start the design process by creating a conceptual design. Traditionally, this stage is done manually by hand drawing 2-D sketches or carved foam models. As mentioned earlier, a conceptual design is created to the approximate shape, size, and location. Once this conceptual design stage is done, a process planner (high-level) starts to consider candidate processes that can be used to produce the part. Currently, there is no CAD-based high-level process planning system and high-level process plans are still developed manually or with some kind of computer-assisted systems. After candidate processes are determined, DFM may be applied to develop the more detailed design, which includes exact dimensions, relations, and positions. This part information is then represented in a CAD model.

1.4 The Role of Feature Technology in Process Planning

In recent years, feature technologies such as design by feature and feature recognition have been used in an effort to realize the integration of CAD and CAM. Chapter 2 discusses these two approaches in feature-based systems. Design by feature or feature-based design has been mostly used in low-level process planning systems [Shah 1995, Mäntylä 1989, Cutkosky 1990]. However, these low-level process-planning systems mainly consider machining processes, and the features included are called machining or process planning features. If the design features represent the geometry construction units, then the machining or process planning features would represent the volume of material that needs to be machined. One of the shortcomings of designing with machining features is that only negative features, such as holes and slots, are considered [Shah 1995, Smith 1997]. The other shortcoming of this approach is that it only allows designers to create standardized or frequently used parts due to the limited of number of predefined features that can be stored. On the other hand, in the feature recognition based systems, designers work on an existing CAD system and feature recognition methods are implemented in the CAD system to recognize and extract the feature information from the CAD model.

When compared to the previous approach, the feature recognition approach gives more flexibility to designers to create more complex parts, but the main disadvantage of this approach is that it requires complex algorithms, even for a simple feature, and the searching process can be very time consuming. Chapter 2 presents a review of existing design by features and feature recognition approaches.

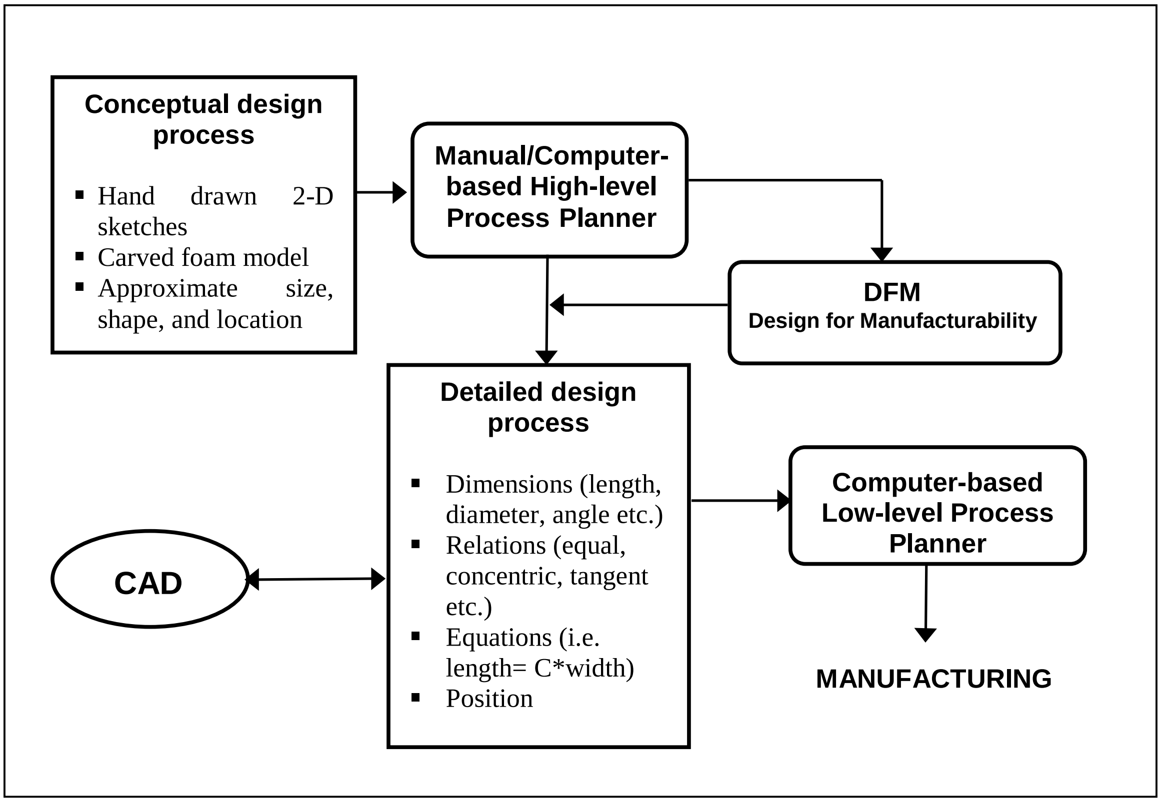


Figure 1.2 Current Practice in Design

1.5 Problem Definition

This research focuses on the following specific problems:

(1) *The inefficiency of current practice in design*

As previously discussed, the product development cycle using traditional practice in design can be time consuming since there is often a lack of communication between design and manufacturing engineers. There may be numerous design revisions where manufacturing engineers find that the design is too difficult or expensive to manufacture. Product development can be shortened if concurrent engineering efforts, where design and manufacturing engineers work simultaneously, are followed from the beginning. Alternatively, design engineers may be provided with a tool that can provide downstream information for their design. Therefore, an intelligent CAD-based solution interface system is recommended for design improvement purposes.

(2) *Recognizing application-dependent features for non-machining process domains*. Machining has consistently been used as the process domain in feature technology research. Several techniques for recognizing both negative and positive features of machined parts have been developed. Even though in non-machining process domains such as net-shape manufacturing processes, features to be recognized are quite different than those in machining processes, feature mapping still needs to be done in order to create a more intelligent CAD system.

1.6 Organization of the Dissertation

This chapter has presented the background of design practice, process planning systems and the implementation of feature technology. A basic problem definition has also been provided. Chapter 2 presents a literature review of high-level process planning systems and feature technology. The research objectives are stated in Chapter 3. Chapter 4 presents the framework of the feature-based design and process planning (FEBDAPP) system. It covers three subjects: (1) part creation, (2) feature mapping, and (3) system integration. Chapter 5 presents detailed procedures of how application-based features are derived from the CAD model. Chapter 6 presents the implementation and testing of the FEBDAPP system. Finally, Chapter 7 summarizes the methods and their advantages, and offers suggestions for further research.

2. LITERATURE REVIEW

2.1 High-Level Process Planning

A great deal of research has been performed in the area of process planning. Most of this research is concentrated on the development of manufacturing processes with detailed specifications such as jigs and fixtures, machine tools, tooling, machining parameters, etc. The Computer-Aided Process Planning (CAPP) systems rely on detailed part specifications. The proposed research aims to provide an early determination of process and cost estimates while the part is being designed. This is referred to as high-level process planning (HLPP). The output of HLPP may influence the detailed design specifications. No effort is made here to provide a review of "low-level" process planning research. Research on low-level process planning is well documented by Alting and Zhang [Alting 1989].

There is a large body of research related to low-level process planning including the development of various CAPP systems for machined parts. However, there has been relatively little work done in the area of high-level process planning. In this chapter, some important research activities that have made major contributions to the development of process and/or material selection systems are presented. Since the research employs feature modeling as the basis for the part description, a separate section is devoted to a review of research related to feature modeling.

2.1.1 Computer Aided Process/Material Selection

One of the earlier research efforts in the area of combined material and process selection was conducted by Wilson et al. [Wilson 1982]. This work resulted in the development of a Fortran-based computer program for material and manufacturing process selection. This research was then continued by Shea and Dewhurst [Shea 1988] who developed a system using a commercially available relational database system. The result was a primary material/process selector called *Computer-Aided Material and*

Process Selection (CAMPS). In CAMPS, process candidates are identified based on the inputs of part shape, size, production parameters, mechanical properties, thermal properties, electrical properties and physical properties. A comprehensive set of processes is covered in this work including solidification processes, bulk deformation processes, material removal processes, and sheet forming processes. This research focuses on primary process selection, and as a result, the possible combinations of material and process generally decrease as the specification of the part becomes more precise. The CAMPS system does not provide early cost estimates for each process candidate.

2.1.2 Expert Processing Sequence Selector

Farris [Farris 1992] developed an *expert processing sequence selector* in an effort to overcome the associated problem with the previous work, which is the omission of some appropriate combinations of primary processes and materials. The procedures in this system are divided into four categories: geometry input, process selection, material selection and system update. In describing the geometry of the part, the user classifies it according to size, shape, cross section and features. *Pattern-matching* is used for process-shape relationships and process candidates are selected based on rules that consider the geometry of the part. Primary process selection is made with regard to the restrictions on the size of the enclosing envelope, the size and shape of the fundamental envelope and the cross-section of the part. Each feature on the part is assessed as to whether the primary process can also form the feature. If the primary process cannot make a feature, then the system finds the primary/secondary processes to make the feature. The material selection procedure utilizes fuzzy logic to model the imprecise material constraints and to select appropriate materials. The final step is to provide an early cost estimate for each process and material combination. Process coverage includes injection molding, plastic extrusion, blow molding, thermoforming, foam molding, rotational molding, thermoset compression molding, transfer molding, and compression molding.

This work can be considered comprehensive since it covers a wide range of high-level process planning functions including process and material selection and early cost estimates for several processes. This system seems to lack a CAD environment, which is a shortcoming. Principally, the system provides rules for determining primary, primary/secondary, and tertiary processes. If these rules are applied within a CAD-based design environment, a much more powerful high-level process planning system would result. With regard to fuzzy logic application, this system does not provide support for weighting of the criteria used. Also, it does not support comparisons among the combined process and material candidates.

2.1.3 Computer-Aided Design for Manufacturing Process Selection

Yu et al. [Yu 1993] developed the Computer-Aided Design System for Manufacturing Process Selection. Their work focuses on net-shape manufacturing processes such as injection molding, die casting and forging. In determining the process candidates, the system uses an index that is calculated by design compatibility analysis (DCA), which measures the compatibility between the decision factors in process selection and the process candidates.

The process candidates are ranked based on the so-called *match index.* This system gives valuable information to designers in terms of ranking the process candidates. Since in this work, the part geometry is described based on shape classification and its envelope size, the process-shape relationship is not as complete as the Expert Process Sequence Selector developed by Farris [Farris 1992]. Computer-Aided Design for Manufacturing Process Selection does not provide information on early cost estimate.

2.1.4 Cambridge Materials and Process Selector (CMS and CPS)

The Cambridge Materials Selector (CMS) was introduced mainly as a material selection system [Esawi 1998]. This system concentrates on the data modeling aspect by presenting the data in chart format. This data is then used in a process selection system, called the Cambridge Process Selector (CPS). The CPS database contains records of 125 processes and their attributes, which makes this system among the most comprehensive in terms of process coverage.

The CPS approach consists of two steps. The first step screens out processes that cannot meet the design requirements. The second step ranks the selected candidates by economic criteria. A typical screening process chart is shown in Figure 2.1.

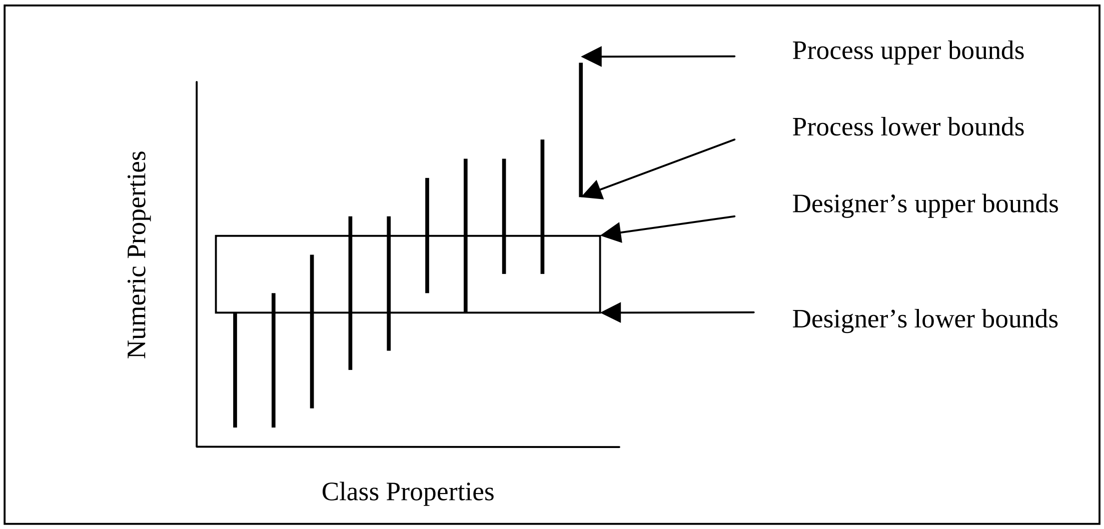


Figure 2.1 CPS screening stage

The vertical lines in Figure 2.1 represent the various manufacturing processes. The CPS typically consists of a 3-stage screening. The numeric properties represent tolerance, size, and minimum section. The class properties represent the process class, material class, and shape class. Each numeric property is plotted for a selected class property. For example, tolerance is plotted for process class, size for material, and minimum section for shape.

Each screening identifies a set of feasible processes. After all screening steps are taken; a final set of process candidates is obtained by taking the intersection of all the possible sets. The process candidates are then ranked based on their cost. A cost model is used for calculating cost estimates.

In this system, the part description must be qualitatively input by the designers that may be considered a shortcoming. Also, users obtain a final set of solutions only after all screenings are complete. The other limitation is that a chart format limits the designer to simultaneously consider only two, at most three, criteria.

2.1.5 Computer Oriented Materials, Processes, and Apparatus Selection System (COMPASS)

COMPASS, developed as a *Meta* planner, is intended to bring manufacturing issues upstream by generating timely and appropriate feedback to design engineers [Chan 1998]. The COMPASS system is perhaps the first work in this area that uses the term *high-level process plan* as an outcome of a Meta planner. In contrast to many existing *low-level* process planning systems that cover *depth*, COMPASS covers *width*. The term depth here refers to the outcome of the low-level process planning system that is very specific, i.e. including tool selection, operation sequence, cutting parameters, etc. On the other hand, the term width refers to the outcome of COMPASS that mainly focuses on selecting feasible process candidates.

A high-level process plan in COMPASS contains complete coverage of all feasible processes for the given design. While many CAPP systems focus heavily on machining processes, this system is aimed at covering a wide spectrum of process domains. The COMPASS system is considered more *dynamic* than most others, since it considers the real-time shop floor status in the decision making process. Not only does the system select feasible processes, but it also considers equipment and tools availability at the manufacturing facility.

Even though the system lacks implementation details, it provide a fundamental framework of how a complete high-level process planning system can be developed. It is said that the system will receive a CAD file as input and convert it into a standardized format for the system. The authors do not state how a part and the features on the part should be recognized and extracted from a CAD drawing. It is not also clear how design features are mapped to process features.

2.1.6 Material and Manufacturing Process Selection (MaMPS)

Giachetti [Giachetti 1998] developed a decision support system for Material and Manufacturing Process Selection, MaMPS. The system is divided into three modules: the material selection module, the process selection module, and the aggregation module. In this system, fuzzy set theory and relational database technology is used in each module for the related decision making procedures.

The material selection module and the process selection module are independent from each other. The material selection module evaluates the compatibility between possible candidate materials and the input material requirements. Likewise, the process selection module evaluates the compatibility between the characteristics of the alternative processes and the input design specifications. An aggregation module joins the two aggregated compatibility ratings. The outcome is a final ranking of possible combinations of materials and manufacturing processes.

While other systems typically provide either decision support or database support, MaMPS offers a combination of decision-making theory with database management. MaMPS, like the other systems described in this chapter, is not CAD-based. The system uses a rudimentary method of describing part geometry, and it does not provide any estimate of cost. Design synthesis and design evaluation cannot be made simultaneously, which lengthens the design development cycle.

2.1.7 Manufacturing Advisory Service (MAS)

Manufacturing Advisory Service (MAS) was developed by Smith [Smith 1999] as a manufacturing process and material selection tool. MAS generates a dialogue with the designer, inquiring about batch size, typical tolerances, size, overall shape, and cost requirements. At each step along the way, the user is presented with an updated, ranked list of manufacturing possibilities. A similar method is used to define the attributes for material selection such as yield strength and density to generate material rankings. The final result is a ranked list of viable combinations of materials and processes, obtained through a process/material pair optimization.

MAS is the newest and perhaps the most comprehensive high-level process planning system, since it accommodates almost all of the capabilities of the previous systems. Giachetti's work is adopted in ranking all possible combinations of materials and processes. In addition to performing primary process selection, MAS also sequences processes. However, since a part is described based on a geometry-based group classification, the misclassification of part shape is frequently encountered.

2.1.8 Process First and Material First Approaches

There are several textbooks and handbooks that provide guidelines in material and manufacturing process selection. One is "Engineering Design and Design for Manufacturing a Structured Approach: Text and Reference for Mechanical Engineers" from Dixon and Poli [Dixon 1995] that uses two approaches: the *process first approach* and the *material first approach* in solving the material and process selection problems. In the process-first approach, the input is the part information about production volume, size, and shape. Using the part information, feasible processes are identified. Once a process has been selected, the next step is to find a material class that is associated with the selected process. Application information is then applied to identify the final feasible material class(es). In the material-first approach, the search starts with application information. Feasible material classes are determined based on the application information. From the list of feasible material candidates, associated processes are then selected. After considering part information, the final feasible processes are listed. It is stated that both approaches should end up with the same results. The schematic representations of both approaches are shown in Figure 2.2 and Figure 2.3.

A comprehensive set of charts and tables is provided that can be used to evaluate design and materials. However, the approaches used in selecting materials and processes do not explicitly describe the process-shape relationship. Also, as with any reference handbook, the user is required to manually locate relevant information, evaluate it, and compare the alternatives.

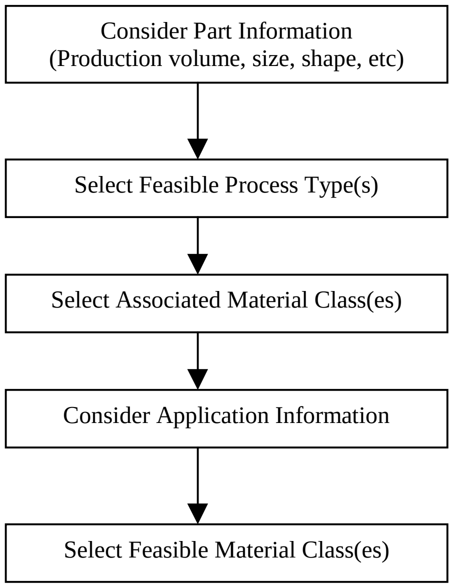


Figure 2.2 Process-First Approach

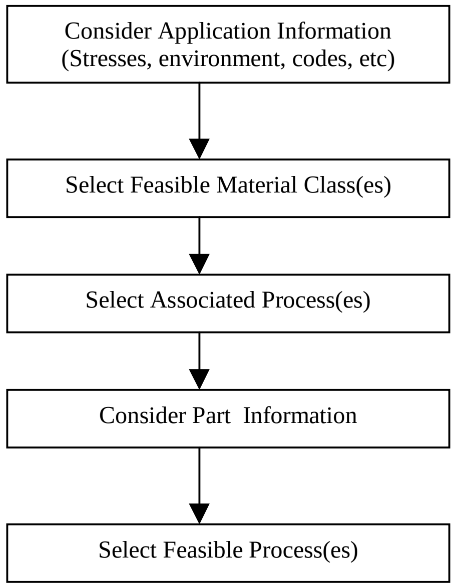


Figure 2.3 Material-First Approach

2.1.9 Others

Several other systems have been developed that consider just material selection or process selection. Goel and Chen [Goel 1996] applied the application of neural networks for material selection in engineering design. Parkan and Wu [Parkan 1998] use multiple objective and subjective attributes for process selection. Liao [Liao 1996] uses fuzzy set theory to select materials. All of these systems do not adequately address part description but they provide for matching of the part attributes to material and process requirements. However, they can only match the part attributes with some dynamic capabilities of process domains by introducing advanced techniques such as fuzzy logic and neural network. The research efforts focus primarily on developing tools rather than high-level process planning methodologies.

Table 2.1 summarizes the extent of the part information details allowed by each of the systems reviewed in this chapter. Table 2.2 summarizes all the design attributes required for the reviewed process selection systems. It should be noted that none of the existing systems, except COMPASS, uses a CAD file as input or creates the part model in a CAD-like environment. However, COMPASS has proposed a framework for a meta planner that will receive CAD files as input to the system. It is not clearly explained how the system extracts information from the CAD file.

Table 2.1Geometry Information Comparison

Features	**CAMPS**	**MAS**	**EPSS**	**DCA**	**MaMPS**	**CPS**	**COMPASS**	**Others**
Part Volume	X	X	X		X		X	
Part Weight	X				X	X		
Shape Descriptor	X	X	X	X	X	X	X	
Secondary Geometric Features	X	X	X	X	X	X	X	
Tolerance	X	X	X	X	X	X	X	X
Surface Finish	X	X	X	X	X	X	X	X
Wall Thickness		X			X	X		
Has Measure of Complexity					X	X		
Check for Uniformity of Wall	X	X						
Use CAD-based environment	-	-	-	-	-	-	X	-
Use CAD file as input	-	-	-	-	-	-	X	-

One likely explanation for the absence of CAD input to high-level process planners is that CAD systems are typically used for detailed design rather than early conceptual design. The next section provides a review of the research related to part description input, specifically as related to CAD systems.

Table 2.2 Summary of Existing Process Selection Systems

SYSTEM	DEVELOPER	MATERIAL REQUIREMENTS	DESIGN REQUIREMENTS
Manufacturing Advisory Service (MAS)	Charles S.Smith, UC at Berkeley 1999, Ph.D Thesis	1. Cost per Pound 2. Yield Strength 3. Density 4. Process Compatibility 5. Thermal Expansion 6. Elastic Modulus 7. Hardness	1. Batch Size 2. Shape : - Prismatic → Milling - Surface of Rotation → Turning - Planar → Sheet Metal Blanking - Thin Walled → Injection Molding - Constant Cross Section → Extrusion - Draped Free Form → Forging - General Freeform → Sand Casting 3. Bounding Box 4. Tolerance 5. Surface Finish 6. Wall Thickness 7. Production Rate 8. Setup Time 9. Setup Cost 10. Per part Cost
Material and Manufacturing Process Selection System (MAMPS)	Ronald E. Giachetti 1998 J.of Intelligent Mfg 9,265-276	1. Hardness 2. Stiffness/density 3. Str/density 4. Density 5. Cost/kg 6. Thermal condition 7. Corrosion	1. Undercuts : Yes or No ? 2. Wall Thicknes : Min and Max 3. Overal Dimensions : Length, Width 4. Weight 5. Shape : - Cone axial - hollow axial - prismatic - round flat - rounded box - solid axial - square flat - squared box - tank 6. Surface Finish 7. Tolerance 8. Production Rate : part/ht 9. Production Volume 10. Time to Market : # months
CMS Process Selector	Esawi, AMK & Ashby M.F. , Univ of Cambridge	1. Material Class : - Light Metal - Ferrous - Fine ceramic	1. Process Class : - Primary, discrete 2. Shape Class: - Prismatic - Sheet - 3-D 3. Size (kg) 4. Minimum section (mm) 5. Precision (mm) 6. Surface Finish (μm) 7. Batch Size

Table 2.2 Summary of Existing Process Selection Systems (continued)

SYSTEM	DEVELOPER	MATERIAL REQUIREMENTS	DESIGN REQUIREMENTS
Computer Oriented Materials, Processes and Apparatus Selection System (COMPASS)	Chan, K et al. 1998. J.of Mfg Systems Vol. 17 No.4	Name : 6061T6 aluminum alloy etc.	1. Shape : - Rotational - Non-Rotational 1. Tolerance 2. Surface Finish 3. Production Volume 4. Bounding Box
Computer-aided Design for Manufacturing Process Selection	Yu, J.-Cheng et al. 1992. J. of Intelligent Mfg, 4, 199-208	1. Mechanical properties: - Stiffness - Hardness - Strength etc. 2. Physical properties: - Erosion - Resistance - melting temperature etc	1. Shape 2. Bounding Box 3. Weight 4. Tolerance 5. Surface Finish 6. Production Quantity
Computer-aided Process/Material Selection	Shea, C and Dewhurst, P 1989	1. Mechanical Properties 2. Thermal Properties 3. Electrical Properties	1. Shape: - Depressions - Uniform wall - Uniform cross section - Axis of rotation - Regular cross section - Captured cavity - Enclosed cavity - No draft 2. Size 3. Production parameters
Expert Processing Sequence Selector	Farris, Univ. of Rhode Island 1992, Ph.D Thesis		1. Bounding box 2. Basic shape: rotational and non-rotational 3. Tolerance 4. Surface finish 5. Cross section of the part: - constant - constant, if some features are ignored - not constant 6. Functional features: - depressions - projections

2.2 Feature Technology

The review of the major research work in the area of high-level process planning reveals that the systems developed are generally not CAD-based. Even though some are computer-assisted systems, designers are still required to input information manually. For example, the part geometry in Giachetti's system is described using the following terms:

- Presence of undercut (yes/no)
- Shape: cone axial, hollow axial, prismatic, round flat etc.
- Weight (g)
- Wall thickness (cm)
- Enclosing envelope

In practice, specifying design attributes manually can be time consuming and subject to error and inaccuracies. In this research, it is of great interest to automate the input procedure related to geometrical features.

Currently, geometric modeling techniques such as Constructive Solid Geometry (CSG) and Boundary Representation (B-Rep) are widely used in solid CAD systems. However, the current geometric modeling systems represent geometry models at a low, very detailed level. For example, B-rep models are described in terms of edges, faces, curves, etc. CSG models are described in terms of solid primitives and Boolean set operators. Consequently, it can be difficult to integrate CAD with high-level process planning.

In an effort to overcome this shortcoming, feature technology has been introduced. Feature technology has been a central topic for years mainly in the field of CAD/CAM integration. Feature technology is essentially grouped into two approaches, namely feature recognition and design by feature. The feature recognition approach examines the topology and geometry of a part and matches them with the appropriate definition of predefined features. The design by feature approach builds a part from predefined features where their attributes are attached. Both approaches are advantageous in some ways, but each approach also brings some problems.

One advantage of the feature recognition approach is that the designers can work directly on the current CAD system. However, in order to get information from the CAD model, feature recognition algorithms need to be developed, and this is not an easy task. Even recognition of a simple feature may require a complex algorithm. So the main deficiency of the feature recognition approach is that it can be a time consuming process due to the complexity of algorithms and the need for extensive search to find the information from a CAD model. On the other hand, one benefit of using the design by feature approach is that features can capture the functional intent on which a designer's activity is based within its geometry-based representation [Cunningham 1988, Dong 1990]. In feature-based modeling, a "cylindrical hole" feature has a much clearer meaning than just a "cylinder" primitive in CSG. Also feature-based modeling has become more popular in the field of design and automation since it can facilitate high-level communications between design and manufacturing systems [Shah 1996]. However, the design by feature approach limits designers' ability to create complex parts due to the limited number of predefined features that can be stored in the feature library.

The term "feature" has been used in many different ways and for different purposes. Since form features are the most commonly used types of features in process-planning systems, when talking about features, one usually means form features. There is no single formal definition of a "feature". The various definitions of a feature are summarized in Table 2.3. However, the following definition perhaps can represent the general definition of a "feature":

> *"A feature is a partial form or a product characteristic that is considered as a unit and that has a semantic meaning in design, process planning, manufacture, cost estimation or other engineering discipline"* [Wierda 1991].

Practically speaking, features can be thought of as building blocks in product definition or geometric reasoning [Shah 1995]. The characteristics of a feature are described as follows:

Table 2.3 Definition of Features [Kim 1994]

Definition	Reference
"a specific geometric configuration formed on the surface, edge, or corner of a workpiece intended to modify outward appearances or to aid in achieving a given function"	[CAM-1 1981]
"a geometric form or entity whose presence or dimensions are required to perform at least one CIM function, and whose availability as a primitive permits the design process to occur	[Luby 1986]
"a single face or set of connected faces which has certain characteristic combinations of topology and geometry"	[Sakurai 1988]
"any geometric form or entity that is used in reasoning in one or more design or manufacturing activities"	[Cunningham 1988]
"a region of interest in a part model"	[Wilson 1988]
"a part of a formed object that is physically differentiable from the rest of the object and performs certain functions"	[Gadh 1989]
"objects to which problem-solving knowledge such as process planning will refer. They serve to classify geometric and topological patterns as being manufacturable by one process or another"	[Hummel 1989]
"a geometrical form and a set of specifications for which a process planning process exists and this process is almost independent of the other features of the part"	[Tsang 1989]
"a primitive in a high-level language or representation scheme for defining mechanical parts"	[Requicha 1989]
"regions of a part having some manufacturing significance in the context of machining"	[Joshi 1988]
"recurring patterns or information related to a part's description"	[Shah 1989]
"an abstraction of lower-level design information"	[Pinilla 1989]
"the modeling entities that satisfy the specific needs of various design disciplines"	[Zamanian 1991]
"any geometrical attribute that is necessary for the part to fulfill its intended function"	[Boothroyd 1992]
"higher level entities that model the correspondence between design information and manufacturing activities"	[Regli 1995]

- A feature is a physical constituent of a part.
- A feature is mappable to a generic shape.
- A feature has engineering significance.
- A feature has predictable properties.

As previously mentioned, there are two main approaches that are currently employed in feature-based modeling:

(1) *Feature recognition*. A geometric model is created first, then a computer program processes the resulting model to automatically identify features.

(2) *Design by features*. The geometric model of part is created directly in terms of features.

2.2.1 Feature recognition

Feature recognition is a post-processing approach; meaning that some procedures need to be applied to the CAD model in order to recognize its features. This is the first generation technique developed in feature technology. With this approach, designers can still work with the existing CAD systems. A significant amount of work has been done in feature recognition, and the approaches can be categorized into three groups as follows:

1. Ruled-based approaches
2. Graph-based approaches
3. Volume decomposition approaches

2.2.1.1 Rule-based recognition approach

In this approach, sets of rules are written to define features. Feature recognition is done by matching certain patterns and relationships to the rules of features. The general form of rules can take the following form:

If $(A_1, A_2, A_3, \ldots, A_n)$ then F

Where A_1, A_2, A_3, …,A_n are conditions that define the feature F.

The following is an example of a generic rule for defining a hole:

If a hole entrance exists and a cylindrical face is adjacent to the entrance and there is a bottom to the cylinder, Then
The entrance face, cylindrical face and bottom form a cylindrical hole [Chan 1994]

The rule above still needs to be elaborated in more details, at a lower level. The CAD system still needs knowledge of how to define a hole entrance, a cylindrical face, its position to the entrance, and whether or not there is a bottom to the cylinder. The geometric knowledge of the CAD systems is normally due to its storing of low-level information based on B-rep such as vertices, edges, and faces.

Rule-based recognition is useful in classifying secondary information for feature recognition, but is inadequate in defining feature topology and separating intersecting features [Chen 1998]. The apparent disadvantage of this technique is that it requires a large number of rules since each rule looks only for one specific feature.

2.2.1.2 Graph-based recognition approach

With the graph-based approach, the topology of the part is converted into edge-face graphs, which represent the adjacency relationship of the bounding surface in the form of connecting nodes and arcs. The graph-based approach matches the predefined feature graphs to sub-graphs from the part. The advantage of the graph-based approach is that the well-established techniques of graph algorithms can be adapted to feature based modeling [Shah 1995]. The disadvantages of this approach are as follows:

(1) The graph matching ensures only topological equality.
(2) The matching algorithm is NP complete.
(3) Intersecting features may bring some problems to this approach since feature topologies are altered.

2.2.1.3 Volume decomposition approach

While the previous two approaches use a boundary representation approach that evaluates the faces, edges and vertices, the volume decomposition approach operates on three-dimensional volumes. The convex hull decomposition was originally developed by Woo [Woo 1982]. This approach uses the convex hull of the part volume to subtract the part volume, and then uses the new convex hull of the resulting volume to subtract the resulting volume until a null volume is met. The resulting volumes are known as alternating sum of volumes (ASV), from which convex components, or sub-volumes, can be generated in the form of a DSG (Destructive Solid Geometry) tree. The main problem with Woo's approach is that it requires a convex hull to start so that it may lead to the problem of non-convergence in some cases.

Overall, since feature recognition approach is a post-processing approach, it allows different CAD systems as input data. This is perhaps the main advantage of feature recognition. Recognition procedures are used to obtain the recognized procedures. However, the approach may lead to infeasible results due to:

1. Imprecise or incorrect geometry in the designed part due to a designer's mistake. For example, in defining a solid axial for part shape, all cylindrical features must share the same axis of revolution. But, if there is at least one cylindrical feature shifted a small amount from the axis of revolution, the part shape becomes undefined.
2. Incomplete feature recognition algorithms. In some situations, there could be some features that cannot be recognized, because those features are not predefined. In some cases, recognition algorithms fail to recognize due to the interacting features.

2.2.2 Design by Feature

In contrast to the feature recognition approach, the design by feature approach involves pre-processing. In the design by feature approach, predefined features are created before the part creation. All attributes of the features are given and stored in the feature library. In creating a part, a designer uses a "pick and drop" technique; that is the user picks a predefined feature from the feature library and then drops it on the screen. Since attributes are attached to each predefined feature, feature recognition is no longer needed once a part is created.

However, the design by feature approach itself has deficiencies. The main weaknesses of this approach are that, feature recognition algorithms are still required, since features are very application dependent. The other weakness, practically speaking, is that this approach makes it difficult for designers to create complex parts.

2.2.3 Feature-Based Design and Process Planning

The topic of feature-based design has been investigated by many researchers in the field of CAD/CAPP integration. Most of the work has focused on developing feature-based process planning systems. Until now, automated feature recognition has been the most common approach used to extract manufacturing features from CAD systems. Almost all feature-based process planning systems that have been developed to date focus primarily on machining processes. Consequently, when a researcher refers to the process of extracting manufacturing features from the CAD system, he or she is usually referring to machining features.

As previously discussed, a problem of feature recognition is that it needs to infer a lot of information from the CAD product model and usually at a high cost when this information has already been generated during the design process [Salomons 1993]. Also, technical information is unavailable in the source geometric database and, consequently, the user has to obtain the information from other sources [Xiang 2000].

In an effort to overcome the problem of feature recognition, the design by features approach has been widely adopted. In feature-based process planning systems for machined parts, a *destructive solid geometry* approach is employed [Smith 1996]. The part creation starts with the selection of the initial stock of raw material that is also called the base feature, i.e. rectangular block, cylindrical block etc. After that, form features or machining features are placed sequentially on the part or features. The machining features here represent material volumes that need to be removed from the material stock.

The First-Cut system by Cutkosky and Tenenbaum [Cutkosky 1990] is a feature-based process planner in which the designer creates the part in terms of manufacturing features. HutCAPP is a similar system [Mäntylä 1989] that provides a facility for changing the feature specification of the part.

One major drawback of the design by manufacturing feature approach is that only negative features (holes and other features that subtract material from base) are available. Positive features such as bosses, ribs and other projection features are difficult to represent directly with machining operations. Therefore, they must be mapped to equivalent material removal features [Shah 1994].

Since a high-level process planning system should cover a wide spectrum of process domains, if the design by feature approach is adopted, then the main challenge becomes how to establish a correct and complete transformation from the primary design feature representation to the secondary manufacturing feature representation that is required for the application [Cunningham 88]. Applications here may include process planning, cost analysis and manufacturability.

2.3 Summary of the Existing High-Level Process Planning Systems

The review of the existing research related to high-level process planning systems points out that virtually none of the systems allow CAD-based input. All material and design attributes are defined manually via text-based methods. In defining the shape of the part, designers need to select a single overall description of the part. In doing so, there are three inputs that help designers make a selection: a textual description of what kind of shapes fit into the category, sample illustrations of typical parts, and a list of components that would fall into each category [Smith 1999]. Unfortunately, classification and coding systems, which use a similar method to describe part geometry, have long suffered from problems of ambiguity and misclassification. Simply put, it is not easy to describe complex geometries with a small set of textual descriptions.

Most of the current work in the area of process and material selection describes the shape of a part in a qualitative form, i.e. prismatic, constant cross section, thin wall, hollow, round box, etc. Virtually none of the existing systems use CAD files as their input. With regard to the nature of the preliminary design itself where a designer creates a part design by just adding, deleting and manipulating features, a feature-based environment will significantly help the designer create preliminary designs with an emphasis on ease of manufacturability.

In practice, determining the cost of making a part at the early stages of design is more crucial than low-level process planning. Most of the reviewed existing systems do not provide early cost estimates. For designers, early information on the cost of the part is very important in determining whether or not a company should proceed to the detailed design stage. In the existing systems, the cost estimation is non-existent or difficult to implement. Usually the design attributes associated with the processes are manually input, which is time consuming and subject to human error. For example, in defining the complexity of the part shape of an injection molded part, the reviewed systems may detect whether or not there is an undercut. However, in the cost analysis for injection molding process, not only is the presence of an undercut important, but also the selection of the parting plane, the number of undercuts and the types of undercut (i.e. internal or external) are crucial. In order to obtain a good early cost estimate, feature information from the part needs to be available. In some net shape manufacturing processes such as injection molding and die-casting, the complexity of the part's shape will significantly affect the manufacturing cost, in particular, the tooling cost. So, it is believed that a good cost estimate can only be obtained if the part shape can be well described.

Feature-based modeling, in particular the design by feature approach, is considered a new way of creating a part design. The feature-based design approach will make traditional CAD systems more intelligent so that designers can create and evaluate designs concurrently.

3. RESEARCH OBJECTIVES

In each of the reviewed process selection systems that have been discussed in Chapter 2, users or designers manually input the material and design attributes [Yu 1993, Farris 1992, Giachetti 1998, Eshawi 1998, Smith 1999]. The candidate processes are then obtained by matching all the design attributes with the process capabilities of each process domain. Some systems, such as the Expert Process Sequence Selector [Farris 1992], Manufacturing Advisory Service [Smith 1999] and Material and Manufacturing Process Selector [Giachetti 1998] use advanced decision-making processes such as fuzzy logic to select and rank the candidate materials and/or processes. These systems, however, do have drawbacks. They are not CAD-based systems, and they provide very limited support for downstream applications such as cost estimation.

The main objective of this research has been to develop a FEature-Based Design And Process Planning (FEBDAPP) system that helps designers improve their designs by providing manufacturing advice while they are in the early stages of the design process. In order to achieve this goal, the following specific objectives were set for this research:

1. Develop a hybrid system incorporating design by feature and feature recognition approaches capable of reducing the complexity of feature recognition algorithms without sacrificing flexibility in creating a part design.
2. Develop a comprehensive set of *B-Rep* based feature-mapping procedures that are capable of transforming the primary part representation into *application-based features*. The application-based features are basically design parameters used in applications, in this case, the high-level process planning system.
3. Develop CAD-based procedures and interface software capable of interfacing the current CAD systems and the relational database management system used in the existing high-level process planning system.

It is believed that this dissertation contributes significantly to the availability of early design tools that enhance and at the same time shorten the design process cycle. The implementation of feature modeling in this research will support the development of applications such as tooling cost estimation. Finally, the FEBDAPP system is intended to be an effective concurrent engineering tool that bridges the gap between design and manufacturing.

4. FRAMEWORK OF THE FEBDAPP SYSTEM

In this research, a CAD-based interface that extracts or creates design information that is needed for a high-level process planning system has been developed. It implements functions that utilize the feature information to provide designers with information on such factors as process selection and cost estimation that are valuable for an efficient design process. In order to create an interface that is capable of providing this kind of information, a *hybrid* system of feature recognition and design by feature approaches has been proposed and implemented in this research.

Figure 4.1 shows the overall framework of the Feature-Based Design and Process Planning (FEBDAPP) system. As shown in the figure, the existing high-level process planning systems have focused primarily on process selection. Although they are computer-aided systems, most of them call for manual input of design attributes and are not CAD-based. In order to develop an intelligent CAD-based system that can be integrated with downstream applications, this research focuses on (1) part creation, (2) feature mapping, and (3) application development. In this research, that is a high-level process planning and cost estimation system.

In part creation, designers use either predefined features or a sketch in creating a part. When dealing with the most frequently used parts, the design by features approach will be more favorable, since feature information can be directly derived. However, designers will frequently have to create more complex parts from sketches for which predefined features cannot be used. Even though the design by feature approach tries to eliminate the task of feature recognition, a search algorithm is still needed for certain information that is required for downstream applications.

In this research, a hybrid system of design by feature and feature recognition approaches is employed for the following reasons:

1. Designers tend to model parts from sketches rather than predefined features, unless they are dealing with standardized or frequently used parts. In other words, a *pure* design by feature approach alone is only suitable for standardized parts.

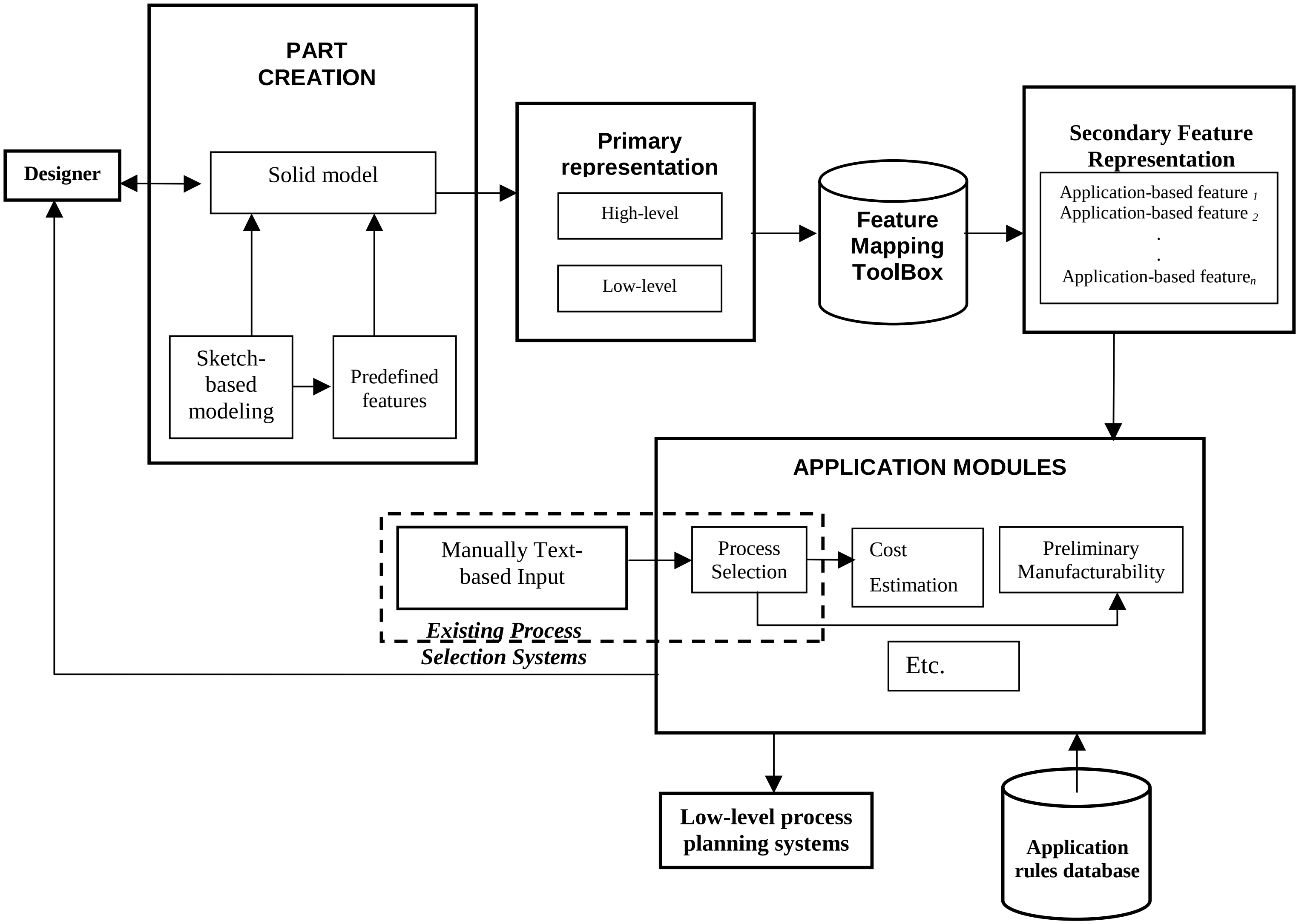


Figure 4.1 The Framework of the FEBDAPP System

2. Even though a whole part is built from predefined features, the feature recognition approach is still needed for recognizing application-dependent features, such as undercuts in injection molding and die-casting and part shape in process selection.

This research is intended to bring together the advantages of both design by feature and feature recognition approaches. It should be noted here that the feature recognition approach implemented in this dissertation is different from what is found in existing research on extracting machining feature using the feature recognition approach. It is of great interest in this research to focus on net-shape manufacturing process domains. For machined parts, the major challenge lies in determining how to convert the solid model or design features into machining features, in particular how to recognize and extract the interacting feature and to map the positive feature into the sequence of machining features that need to be taken away from the stock.

In high-level process planning systems, there are decision factors that need to be defined. They can be classified as related to *geometrical features*, *technological features*, and *production features* [Giachetti 1998]. Since this research is intended to select the candidate primary processes for a given part, the focus is on geometrical features. Geometrical features here include part shape, wall thickness, part size, weight, and undercuts. Technological features, which include surface finish and tolerance, are important factors, but they are not *driving* factors for primary processes. For example, even though a part requires a tight tolerance and fine surface finish, sand casting can be a candidate primary process. Although sand casting results in poor surface finish, an improved finish can be obtained through secondary processes. Production features, which include the production volume, are also not driving factors for primary process selection.

In order to be compatible with the existing process selection systems, the system must fully accommodate all design specifications required by the existing systems. Therefore, the primary CAD representation is enhanced with additional data that is needed for high-level process planning. In high-level process planning for net-shape manufacturing processes that do not require material stock removal, design features are of principal interest. The feature mapping in this research is mainly concerned with obtaining *application-based features* from a CAD model. These application-based features are basically design parameters that are required for applications such as process selection, manufacturability analysis, and cost estimation. It is of great interest to develop a *toolbox* that contains procedures for the mapping process. It is intended that these tools are generic enough so that they can be used for many downstream applications. Both feature mapping and application-based features will be discussed in detail in the next section.

This research has aimed at developing a framework for a feature-based high-level process planning system. As discussed in Chapter 1, the main interest at the conceptual stage is to find the candidate processes and cost estimates. For demonstration purposes, Giachetti's high-level process planning system is adopted to select the candidate processes. Once the processes are selected, the next task is to find the cost. Tooling cost estimation procedures for net-shape manufacturing processes such as injection molding and die-casting have been developed and are presented in Chapter 5.

Before proceeding further, it is necessary to look at the proposed product development process as shown in Figure 4.2. Instead of using a hand-drawn sketch, a 3-D volumetric feature is used for building a part. Due to the nature of design at the preliminary stage, a part is created to the approximate size, shape, position, location etc. Part creation is done in a feature-based CAD environment. The primary part representation in a CAD modeler then becomes an input to the high-level process planning system. Once a candidate process is selected, the associated cost estimate can be determined, and the detailed design process can begin. In the detailed design process, the CAD model from conceptual design stage can be used and the process needs not start from scratch.

4.1 Concept of a Hybrid System

As previously discussed in Chapter 2, one likely explanation for the absence of CAD input to high-level process planners is that CAD systems are typically used for detailed design rather than early conceptual design. At the conceptual design stage, designers traditionally transform their idea by sketching a part on a piece of paper. Since a CAD system becomes a common design tool, the CAD system may also be used at the conceptual design stage.

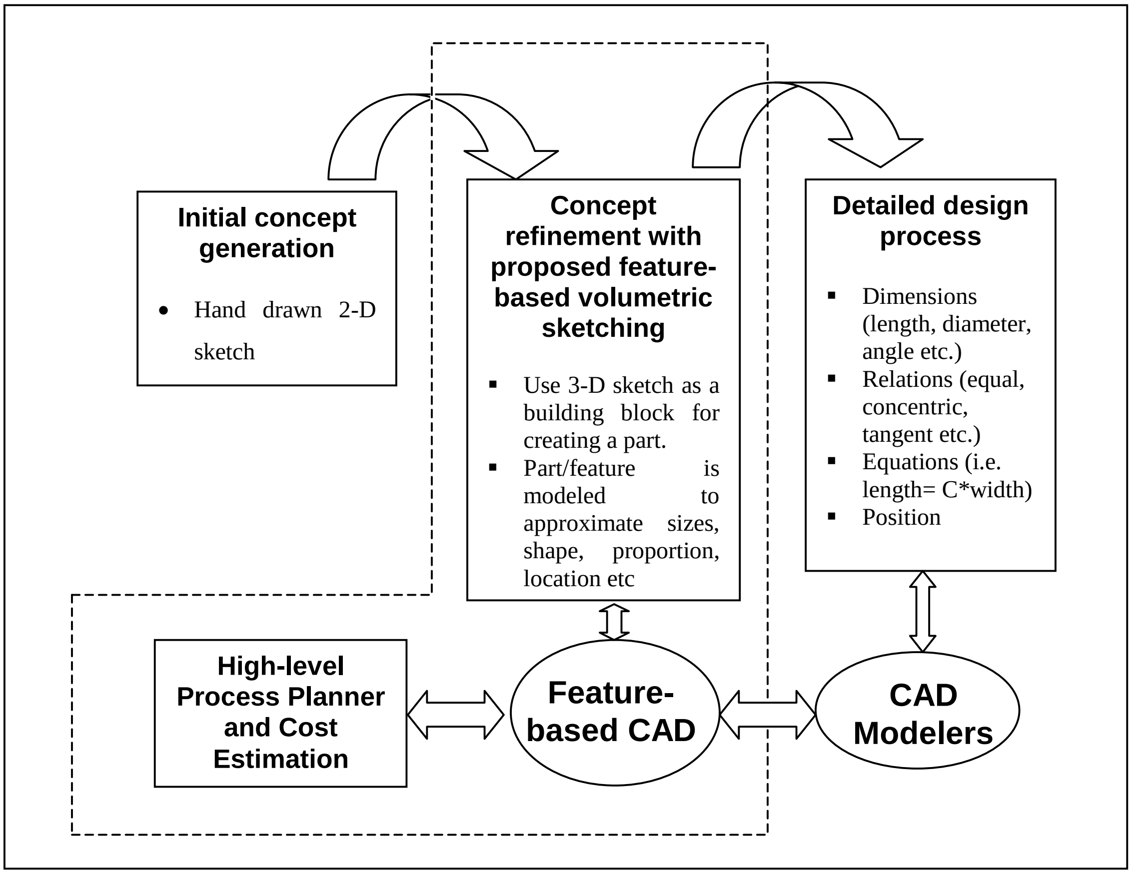


Figure 4.2 Proposed product development process

It is argued that the design by feature approach is more appropriate for the detailed design stage, since it is assumed that a process domain is selected prior to this stage. Once a process is known, it becomes apparent that designers tend to choose features that are "compatible" with the process domain. The term "compatible" can be interpreted as easy to manufacture. So, by employing the design by feature approach at the detailed design stage, designers can avoid the use of features that may be difficult to manufacture. Nevertheless, the design by feature approach itself can be applied at the conceptual stage. Even at the conceptual stage, designers have already in their mind the "standardized", "familiar", or "known" features that will be used.

In a design by feature system, a part **P** is represented in terms of features, $\mathbf{F_{1,..,n}}$ as follows:

$$P = F_1 \cup F_2 \cup F_3 \cdots\cdots \cup F_n$$

Each feature $F_1, F_2, \ldots F_n$ has its own attributes. These attributes can be names of features, feature types, surface finishes, tolerances etc. So, once a part is created, all these features can be directly obtained. To illustrate this idea, consider the following simple rotational part consisting of two cylindrical bosses and one cylindrical hole as seen in Figure 4.3.

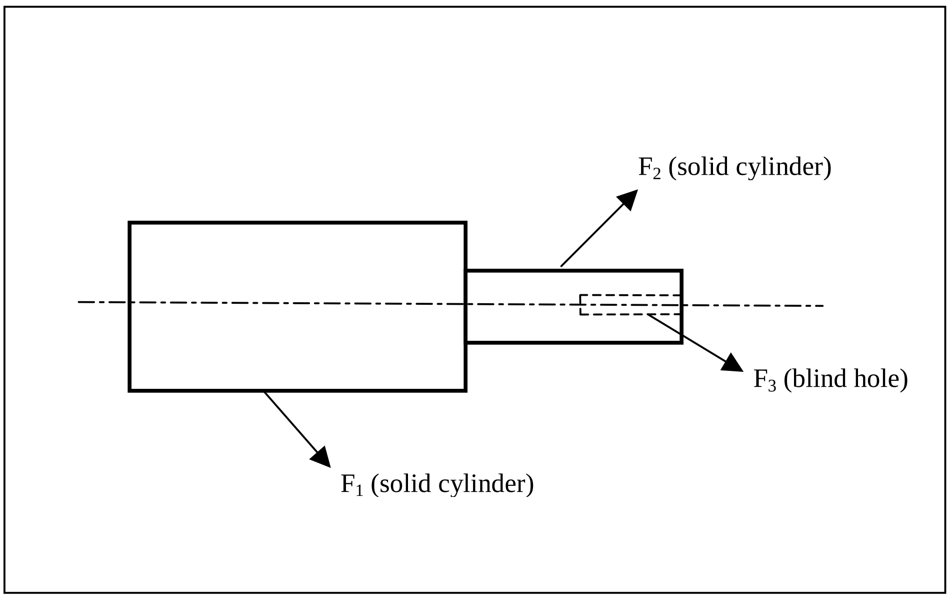


Figure 4.3 A simple rotational part

The feature set for the part, P above can be represented as follows:

P → F_1 *{cylindrical boss, cylinder 1, fine finish, tight tolerance etc.}*

→ F_2 *{cylindrical boss, cylinder 2, medium finish, good tolerance, etc}*

→ F_3 *{cylindrical blind hole, blind hole 1, fair finish, good tolerance etc.}*

All information about the types of features (i.e. cylindrical boss and blind hole), surface information (i.e. fine, fair), and tolerance information (i.e. tight, good) can be easily obtained, since they are already attached and stored in the feature library. On the other hand, when a part is created in a *non-design by feature* CAD environment, the part is represented at the microscopic level, i.e. points, edges, and faces. In order to recognize the features, feature recognition procedures need to be developed. Chapter 2 has discussed several techniques that have been developed in feature recognition. Pros and cons to feature recognition have also been discussed previously.

The hybrid system used in this research is intended to synergize both approaches so that the shortcomings of each approach can be minimized. This system is essentially an improved "pure" design by feature system such that a part may not necessarily be built from predefined features. In some circumstances, when designers have to deal with a novel feature that is not listed in the feature library, they must build it from sketch(es). The feature recognition approach is then employed to recognize this unknown feature.

Adopting the concept of the design by feature approach, volumetric features are used in creating a part instead of a 2-D sketch that is commonly used in practice. Examples of volumetric features are boxes, cylinders, slots, pockets, ribs, etc. Unlike the creation of detailed designs on most CAD modelers, in the volumetric sketching approach, features are drawn to approximate size, shape, and location.

Several CAD systems such as SolidWorks, Pro/Engineer, and Unigraphics already employ the design by features approach. However, these systems still have some limitations as follows:

1. The design by feature approach facilitated in the existing systems is primarily suitable for only orthogonal parts. New features must be placed orthogonally to the planar surface of only the existing feature(s).
2. Even though an orthogonal part has been created, the feature recognition approach is still needed since some features are application dependent.

The feature recognition approach is essentially a *post-processing* approach. It is executed after a CAD model has been created. On the other hand, the design by feature approach is a *pre-processing* approach. Predefined features are created and stored in the feature library before a part model is created. In this research, it is of great interest to do both pre and post processing approaches. By doing so, the designer still has the flexibility to create non-standardized features. At the same time, feature recognition procedures need not be executed for most frequently used parts.

4.2 Part Creation

In the existing high level process planning systems, it is assumed that the designer has already completed the part design with all the design attributes before he or she starts working with the process selection system. These design attributes, which include materials and processes attributes, are manually entered into the system. One of the specific objectives of this research is to simplify and automate this input procedure, in particular the input of geometrical features that include part shape, wall thickness, part size, undercuts, and weight.

Instead of prompting a designer with a set of questions to describe a part, this research has developed a procedure for the part to be represented in terms of volumetric design features. The developed system allows designers to create a part design by adding and manipulating features. Compared to typical text-based classification, the feature-based modeling approach has the following advantages:

1. It is feasible to derive information required for downstream applications.
2. Concurrent performance of both design synthesis and design evaluation using the feature-based volumetric sketching for engineering design results in a faster product development cycle.

Part creation in the design by feature approach consists primarily of two stages: (1) the preparation stage and (2) the modeling stage. In the preparation stage, a feature library that contains a set of predefined volumetric features is built. During the design process, a designer browses this library and selects and manipulates them to design a part.

Two common methodologies for the design by feature approach are *destructive solid geometry* and *synthesis by design features* [Shah 1995]. The destructive solid geometry approach is sometimes called *destruction by machining features,* since it is commonly used in process planning systems for machined parts. As with machining processes, the destructive solid geometry starts with a model of the initial stock from which a part is to be machined, and then material represented by machining or process planning features are removed from the stock. With destructive solid geometry, only feature subtraction is allowed. The synthesis by design feature approach, on the other hand, differs from the destructive solid geometry in that a part can be modeled by adding or subtracting features, and it is not necessary to start the model from the raw stock.

Since the high-level process planning developed in this research is intended to cover a wide spectrum of process domains, the destructive solid geometry approach is not used for this purpose. Instead, the synthesis by design feature approach is used.

4.2.1 The Preparation Stage: Feature Library Creation

In the design by features approach, a part model is built from features. This requires a feature library in which predefined features are stored and can be instanced by specifying their dimensions and the feature/face/edge on which they are to be located and oriented [Luby 1986, Chen 1993]. The most complicated and the least understood issue in feature-based design is determining what features should be included in the feature library. It is apparent that as more features are available, the problems associated with constructing and managing the feature library arise [Kim 1995]. This research creates a set of predefined features that are flexible and can make a wide variety of parts. Since this research deals with the preliminary design of parts, it is intended that only the most common feature types be considered. However, if this limits the designer in creating the part, the system should provide for the designers to create additional features.

Features contained in the feature library are grouped into four categories: (1) *base* features, (2) *add-on* features and (3) *reference* features, and (4) *user-defined* features.

4.2.1.1 Base Features

The base feature refers to the feature that is initially picked from the feature library. The base features to be included in the initial feature library should be flexible so that they can be used as the first feature when creating a part model. The FEBDAPP system is intended as a high-level process planner that covers a wide spectrum of process domains. Realizing the fact that features are application dependent, deciding on a basic set of base features for the feature library becomes difficult. The set of base features considered in this research is presented in Table 4.1. The base features are grouped according to how they are created, namely *extrusion, sweeping, revolving, and lofting.*

It can be seen from the table that all the base features are created from a sketched profile. If the feature is a box, then it starts with a rectangular profile that is extruded a certain distance. A cylinder can be made by extruding a circular profile or revolving a rectangular profile around an axis of revolution. A cylinder can also be made by sweeping a circular profile along a guiding line. A simple loft part is created by using a sketched profile and a point as seen in Table 4.1. A more complex loft part is created by using several sketched profiles and several guiding curves.

4.2.1.2 Add-on Features

Add-on features create attachments or details that are added to the base features and/or to other add-on features [Luby 1986]. An add-on feature may be a child of the base feature or a parent or another add-on feature and in turn be a parent for another add-on feature. Add-on features allow the designer to modify the shape for functional purposes, for example strengthening or smoothing the part [Chen 1997]. Common examples of these features include holes, bosses, ribs, wedges, and pockets.

Similar to base features, add-on features are created by extruding, revolving, sweeping, or lofting. An add-on feature can be modeled as either a positive or a negative volume. Holes and slots are examples of negative features, while bosses and ribs are positive features. Table 4.2 shows an example of a positive add-on feature and a negative add-on feature.

Table 4.1 A Basic Set of Base Features

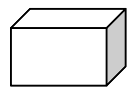
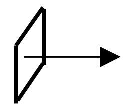
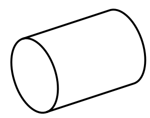
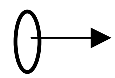
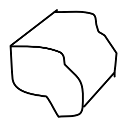
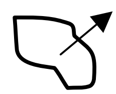
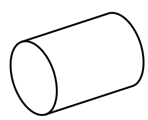
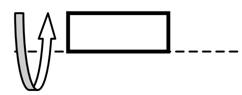
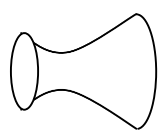
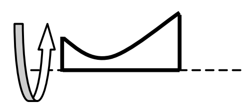
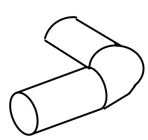
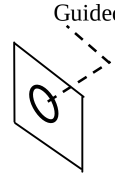
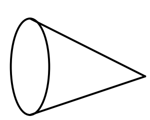
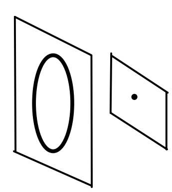
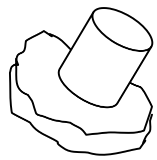
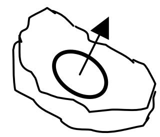
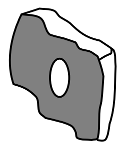
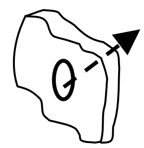
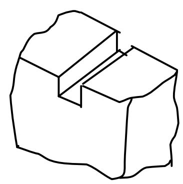
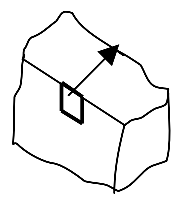
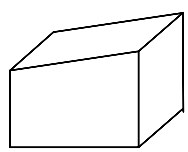
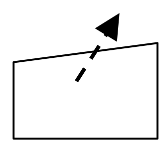
Base Feature Group	Typical feature	Feature Creation
Extruding	 Box	
	 Cylinder	
	 Any constant cross-section feature	
Revolving	 Cylinder	
	 Any revolved features	
Sweeping		Guideding line 
Lofting		

Table 4.2 Examples of positive and negative features

Add-on Feature Group	Typical feature	Feature creation
Extruding	 Boss (positive)	 A circular profile on the base feature is linearly extruded
	 Holes (negative) -blind hole -through hole	 A circular profile on the base feature is linearly cut
	 Slot (negative)	 A rectangular profile on a face of the base is linearly cut
	 Wedge(positive)	

4.2.1.3 User-Defined Features

The main concern of the design by feature approach is the nature and the number of features contained in the feature library. Instead of creating and storing a predefined feature, it is of great interest to create a predefined module that consists of several features, if the module is frequently used. The use of this predefined module, a so-called user-defined feature will, of course, simplify the designer’s task, which will in turn reduce the part creation time.

A user-defined feature is usually a combination of two or more primitives. It becomes obvious that when a compound feature consisting of two or more features is frequently used, a user-defined feature is needed. This can shorten the creation time of certain parts. Figure 4.4 shows an example of a user-defined feature that consists of two features, a base L-bracket and an add-on through hole. Therefore, the L-bracket feature is the parent of the through hole feature. Before creating the part, the L-bracket feature and the through hole feature must be created first and stored in the feature library. The L-bracket is created by linearly extruding an L-profile to a specified depth. For a negative feature such as a through hole, an arbitrary planar base feature must be created first, and then a through hole is created by linearly cutting a circular profile to a depth equal to the thickness of the base feature.

Figure 4.5 depicts the levels of feature creation for the L-bracket. The first level represents a feature that is built from primitives at the second level. The third level defines how those primitives are created. Data at the bottom level are parameters that define each primitive, such as the geometric dimensions.

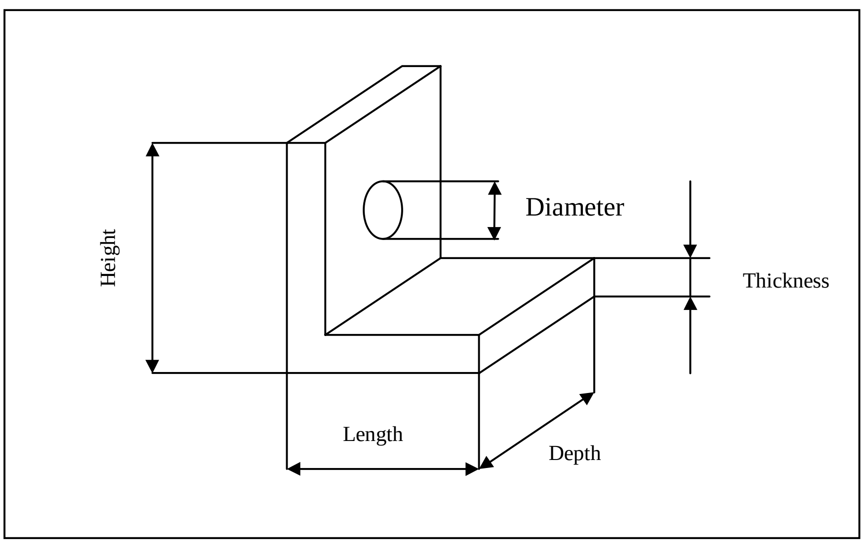


Figure 4.4 An example of a user-defined feature

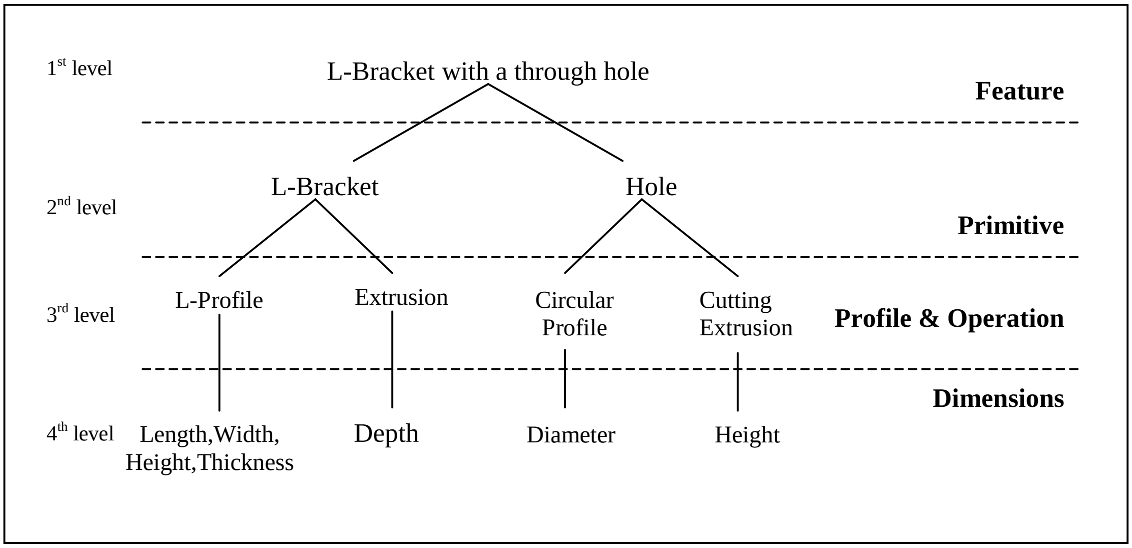


Figure 4.5 An Example of Feature Creation

4.2.2 The Modeling Stage: Part Creation using a Feature-based Volumetric Sketching

The base features and add-on features that are stored in the feature library are basically *volumetric* features. A volumetric feature is simply defined as an increment or decrement to the volume of shape [Shah 1995]. It is of great interest in this research to employ these types of features since they can sufficiently describe a preliminary part representation at the early stage of design. Transitional features such as edge transition and corner transition are not considered in this research, since they are mostly used in detailed design.

After a collection set of base features and add-on feature is stored in the library, the next step is to start creating a part. The part creation procedure, which is basically multiple "drags and drops", can be summarized as follows:

Step 1. Select a base feature from the feature library.

Step 2. Define the base feature attributes.

Step 3. Select an add-on feature(s) from the feature library to be attached on the base feature.

Step 4. Approximately position and orient the add-on feature relative the base or an existing features.

Step 5. Define the add-on feature's attributes.

Step 6. If there are any other add-on features to be attached, repeat from step 3.

Select a base feature from the feature library. From the feature library, pick one feature that will become a parent for all other features. It should be noted that a base feature can never be a child of another feature.

Define the base feature attributes. In the feature library, the geometry of features is defined, but their dimensions and other characteristics are considered as attributes. Any kind of base feature is created only after its attributes are fully instantiated. For example, in order to create a box, its dimension attributes such as length, width and height should be approximately defined.

Select an add-on feature(s) from the feature library to be attached on the base feature. Once a base feature is selected, the next step is to select the add-on features, if any. If the add-on feature to be placed is orthogonal to the face of the base feature, then the add-on feature can be directly placed on that face. But, if the add-on feature to be placed is at an angle relative to the face of the base feature, then a reference feature such as a plane needs to be created first.

Define position and orientation of the add-on feature on the base and/or existing features. Since the proposed system deals with the early stage of design, the location of the add-on feature to be placed only needs to be approximate. Designers would like to be able to later modify its position on the base feature later on after it has been dropped. The orientation of the add-on feature depends on which face of the base feature the add-on feature is placed. Figure 4.6 shows the orientation of a cylinder feature that is placed on a box base feature. From Figure 4.6, it is seen that when Cylinder 1 is placed on face 1, its orientation is parallel to the X-axis, as stated with its normal vector V_1 = (1,0,0). On the other hand, cylinder 2 that is placed on face 2 has a different orientation from that of Cylinder 1. It is parallel to the Z-axis and this is described by normal vector $V_2 = (0,0,1)$.

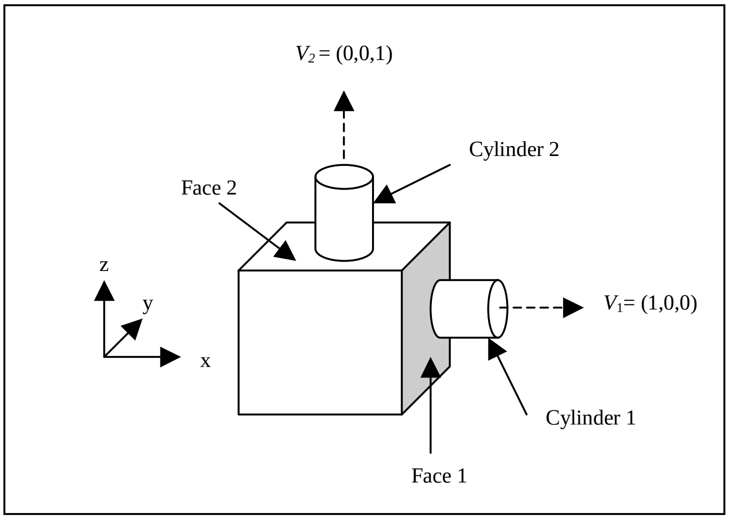


Figure 4.6 Feature orientation

Define the add-on feature attributes. Similar to defining the attributes for the base feature, the add-on feature is created only after its attributes are defined. The attributes for add-on features include dimensions. For example, cylinders 1 and 2 are described by their heights and diameters. The users and designers can modify the added feature by simply "stretching" it to its approximate size.

4.2.3 Part creation in sketch-based modeling

With the current feature-based CAD system, the design by feature approach works well for orthogonal parts only when feature placement is always normal to the base feature's face and the face is always planar. When the following situation occurs: (1) the base or parent face is not planar; and (2) the base or parent face is planar but the additional feature has to make an angle with the base feature, then feature placement in the design by feature approach cannot be done directly. In order to tackle this problem, reference features must be used. The most common reference features are the axis and plane.

Figure 4.7 shows a part where a design by feature approach cannot be used without the existence of reference features. The two horizontal cylinders can be built from predefined features. However, the vertical cylinder cannot be placed directly on the biggest horizontal cylinder. In this case, reference plane 1 needs to be created first. A circle sketch is then created on Plane 1. The next step is to extrude this sketch up to the next surface of the horizontal cylinder.

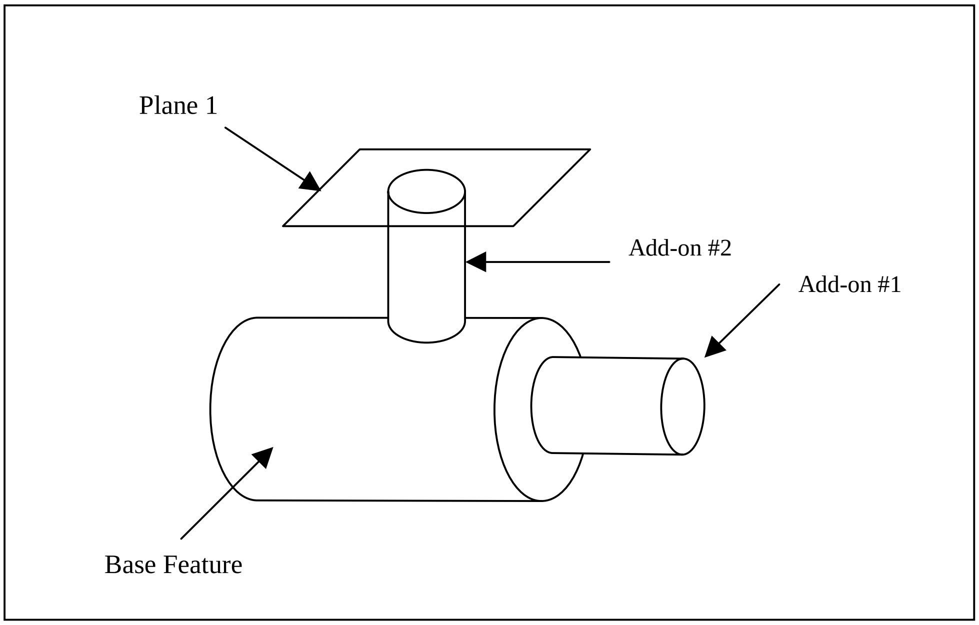


Figure 4.7 An example of the of reference plane

4.3 Primary Representation

Once a part is created, a primary representation in terms of design features is established. In Figure 4.1, the primary part representation can be in terms of low-level and high-level.

When a part is built from all predefined features, such as holes, slots, and bosses, the part is represented at the macroscopic or higher level. If there is at least one feature that is created from a sketch rather than from a predefined feature, then a part is represented in both high and low levels. In the primary representation, a part is the aggregation of features and feature relationships. Figure 4.8 shows how a part is built from a base feature and several add-on features that can be grouped in two categories: cavities and projections. A cavity is created by sketching a 2-D shape, extruding it along 3rd dimension, and then subtracting the extruded volume from the workpiece. Projections are created in a similar manner except that the resulting volume is united.

From the primary representation, a simple database describing the part in terms of all of its features can be obtained. The information from the primary representation cannot be used for downstream applications. However, the feature-based approach allows the necessary information to be extracted from the primary representation. This research described here implements feature mapping in order to convert the primary representation into the application-dependent secondary representation.

As shown in Figure 4.8, the base, depression, and projection features are considered to be *Family* features. Each family has several *Classes*. For instance, a base feature can be a *box*, *slab*, or an *L-bracket*. Depressions can be *slots* or *holes.* An *instance* is a representation of a particular class, while Slab-1, Slot-1 and Hole-1 are instances of Slab, Slot and Hole Classes, respectively.

4.4 Feature Mapping

The main goal of this research has been to make the existing CAD system more intelligent and thus capable of providing input data for different applications such as process planning, cost estimation, and manufacturability. In order to achieve this goal, feature mapping is implemented to obtain feature information required by applications from the primary representation generated at the part creation stage.

The result of this feature mapping is called the *secondary representation*. It should be noted that this secondary representation is application-dependent. This means that the secondary representation for cost analysis is different from that for manufacturability; the secondary representation for cost analysis of die-casting is different from that for cost analysis of stamping and so on. However, it is intended that the feature mapping mechanism be generic so that the system can be customized for different applications with minimal effort [Shah 1988].

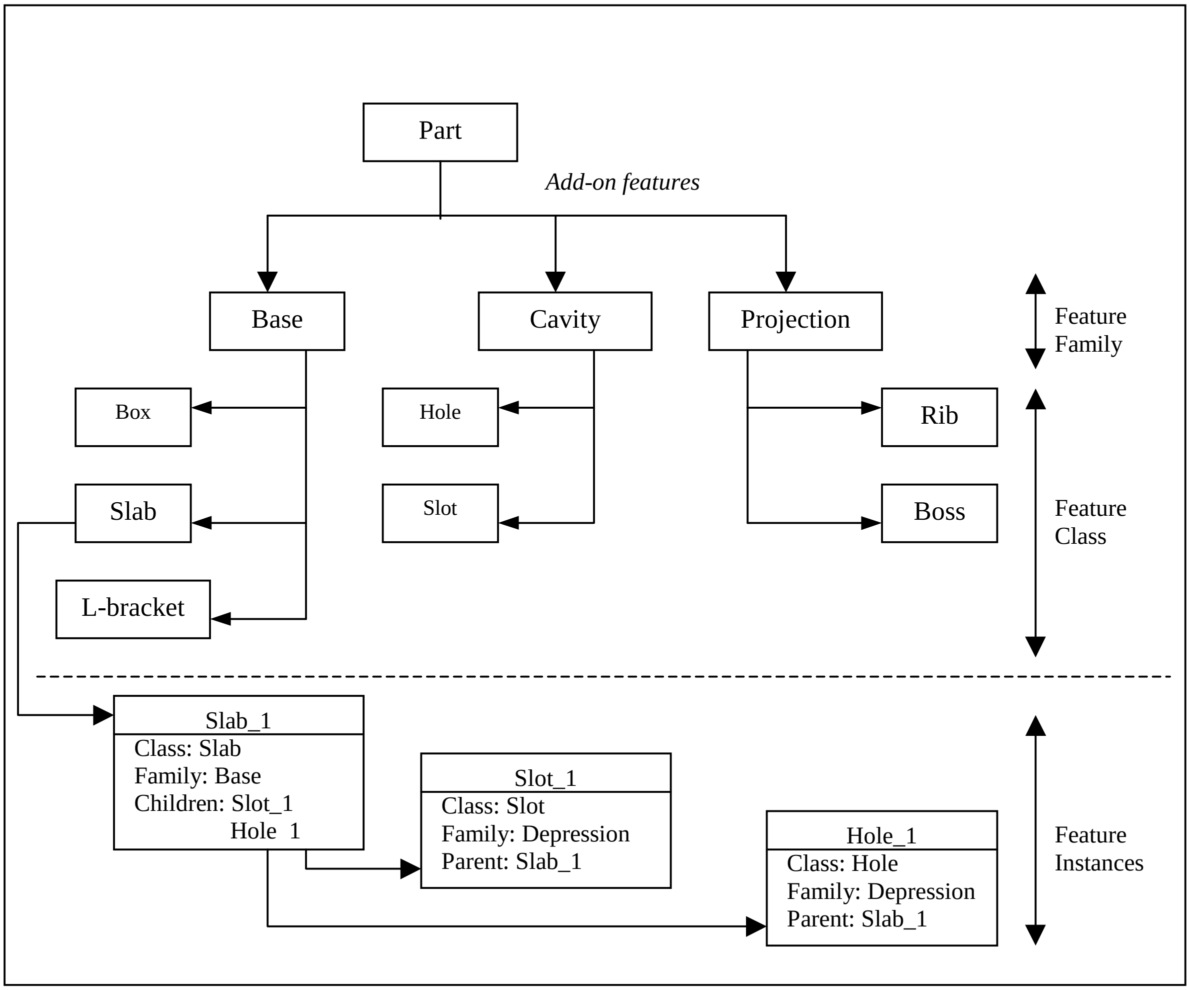


Figure 4.8 Part Structure [Latif 1993]

In the current research of feature technology, feature mapping is intended to convert the design features into manufacturing features. Most of the existing research deals with machined parts, therefore the manufacturing features tend to be the machining features that are always negative features such as hole, slot, pocket, or step. Since this research focuses on net-shape manufacturing processes and not machining processes, the feature mapping in this research is quite different. In the net-shape manufacturing processes, there is no material to be removed from the initial raw stock. So, the main task of feature mapping for these process domains is principally to convert design features into the so-called *application-based features* that are used as inputs in downstream applications.

4.4.1 Application-based features

One of the challenging tasks in this research has been to extract the application-based features from the CAD model. Before the procedures for application-based features are developed, it is necessary to determine which design parameters are involved in the applications. In injection molding cost analysis, for instance, features such as *undercuts, surface finish, tolerance, wall thickness, basic envelope, projections, depressions, and type of parting surface* are amongst the application-based features that determine the tooling cost.

Based on how the part information is obtained, we can classify them into two generic groups as follows:

- *Global type:* This type of feature is determined directly by the user. Direct type features are not derived from the primary representation. Among this type of feature is material, tolerance, surface finish, production volume and so on. This research does not cover this kind of feature, as it is typically given by the user.

- *Application-based type:* This type of feature needs to be derived from the primary CAD representation. The bounding box, shape definition, wall thickness, surface area, and volume are examples of this type of feature.

The global type of attributes are basically non-feature dependent, and their values are not defined by how features are placed and what the feature relationships are. For example, when material properties are given, their values normally apply to all features in the part. The application-based type of attribute is the main consideration in this research since this type of attribute is not provided by traditional CAD systems for use by the downstream applications. In this research, it is of great interest to focus on the application-based type of features.

In an effort to obtain application-based information from the CAD model, it is important that the primary representation of the part be easily convertible to the required application-based features. In Chapter 2, several approaches for feature recognition have been discussed. In this research, the boundary-based feature recognition approach is adopted for the following reasons:

1. The Boundary Representation (B-Rep) provides complete low-level information of the part that can be used for feature recognition purposes.
2. The Constructive Solid Geometry (CSG) representation creates some problems for recognition due to its *non-uniqueness* and *global nature* [Lee 92].

As discussed in the previous section, a hybrid system of design by feature and feature recognition approaches is used in this research. In this research, even though a part is completely built from predefined features, feature recognition is still required for obtaining application-based features. A ruled-based feature recognition approach is applied for obtaining the indirect type of attributes.

4.4.2 Procedures for extraction of application-based features

As discussed in the previous section, application-based features are classified into two types: (1) global type and (2) application-based type. In this research, the application-based type can be differentiated as *explicit* and *implicit* types. Explicit type features are features that can be obtained by a built-in GET method in a feature-based CAD system. For example, SolidWorks and Pro/Engineer provide a comprehensive set of built-in methods for obtaining the explicit type of information. However, most application-based features are implicit. In the high-level process planning system, features such as wall-thickness, undercuts, and shape cannot be obtained simply by using the built-in GET methods. Feature mapping algorithms are required to obtain the application-based features.

In developing the feature mapping algorithms, three-level objects are interchangeably used: (1) PART, (2) FEATURE, and (3) FACE. Figure 4.9 shows the basic procedures of extracting the application-based features from a CAD model. At the PART level, information such as the number of features can be obtained. At the FEATURE level, important steps are performed such as finding the parents of each feature, checking whether one of the parents is a sketch, determining the number of arcs or lines, and determining how a feature is created. Based on this information, one can obtain the very important information pertaining to the feature object, feature shape, and feature relationship. At the FACE level, it is of interest to triangulate an object, get the vertices of each triangle, and get the normal vectors of each vertex. Chapter 5 describes in detail how the algorithms for application-based features are developed.

4.5 System Integration

Once a comprehensive set of feature mapping algorithms has been developed, the next objective that needs to be accomplished is the development of a CAD-based interface that is capable of interfacing the existing CAD system and the relational database management system that is used in existing manufacturing process and material selection systems.

Figure 4.10 shows how the integration of the CAD system and process selection system takes place. First, the designer creates a part in a feature-based working environment. The feature mapping toolbox contains procedures that map the CAD model into the application-based features. After all required application-based features are obtained, this information is then passed to the current high-level process planning system to select the candidate processes. After the candidate processes are selected, designers can execute a cost estimation module to obtain the tooling cost estimate.

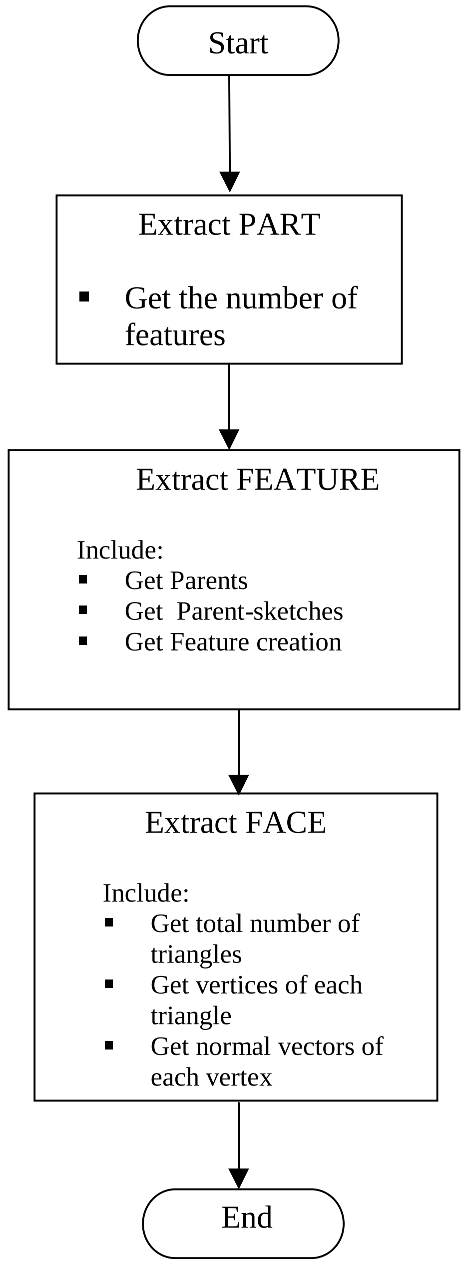


Figure 4.9 Procedures for extracting application-based features

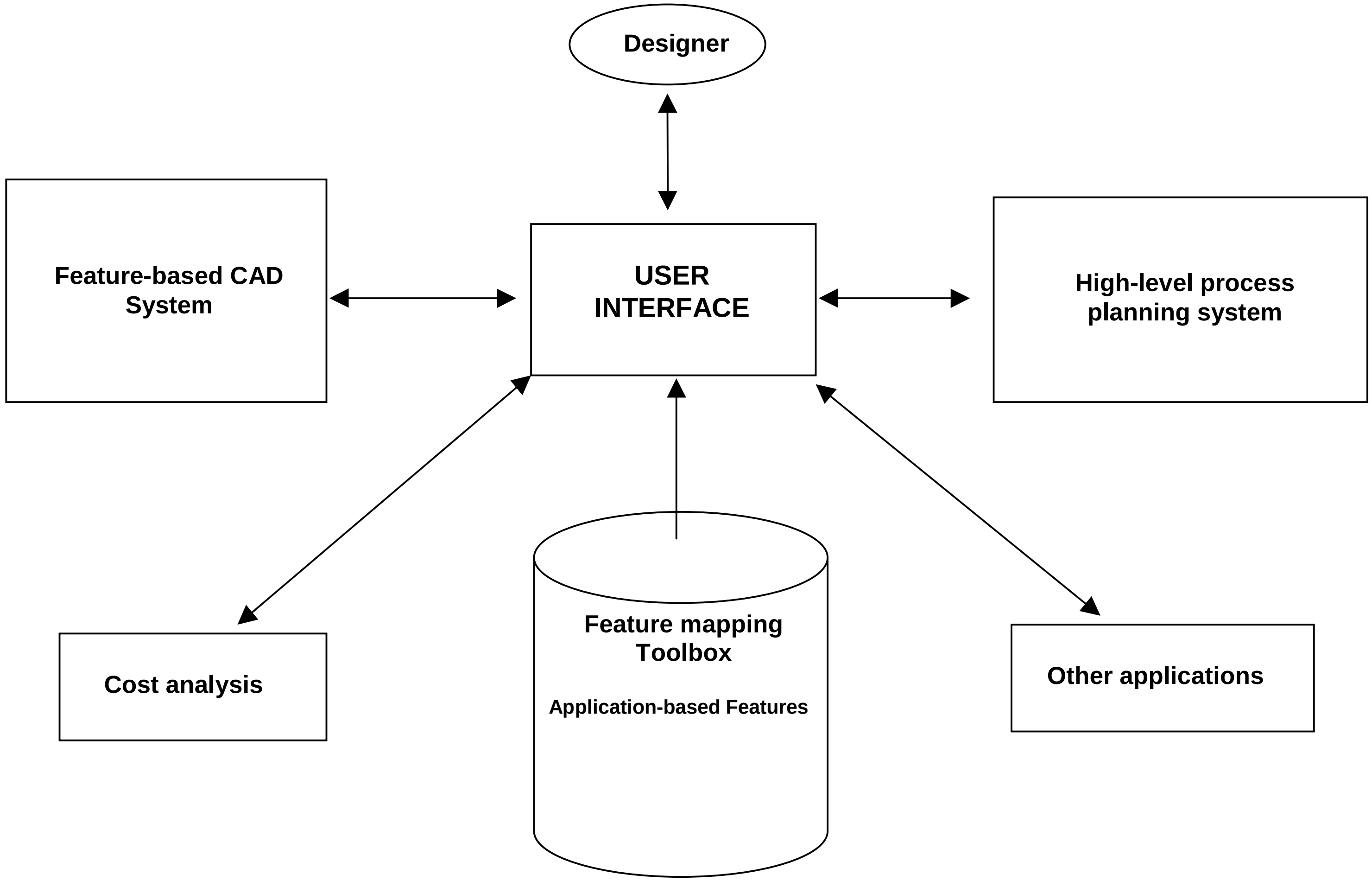


Figure 4.10 A scheme of CAD-Applications integration

The Manufacturing Process and Material Selection System (MaMPS) [Giachetti 1998] is adopted as the source for process selection in this research. MaMPS is implemented in the Microsoft-Access relational database management system. As pointed out in Chapter 2, MaMPS is considered the first process selection system to incorporate the database management system and fuzzy-based decision making process.

4.5.1 Object-oriented approach

An object-oriented approach is adopted in developing a CAD-based interface because of its characteristics of data abstraction, modularity, and inheritance. Figure 4.11 shows the part model represented in terms of the object-oriented approach. Object PART is composed of Features 1, 2, …n. Attributes of this object can be part NAME, TOLERANCE, SURFACE FINISH. Methods for PART consists of the feature mapping algorithm for obtaining application-based features, namely ENCLOSING ENVELOPE, UNDERCUT DETECTION, WALL THICKNESS, SHAPE DEFINITION, etc.

The first feature is the BASE that is parent for other features. The attributes for the BASE feature are NAME and FEATURE CREATION, which tells how the feature is created, i.e. *extrude*, *sweep, revolve, loft*, *and surface thickening*. Methods included in BASE are PARENT/CHILD RELATION, FEATURE SHAPE, AXIS ORIENTATION, NUMBER OF FACES, etc. Feature shape is determined by considering Feature Creation and/or the sketch of that feature.

For example, without looking at the shape of the sketch, if the feature is created through *revolve*, one possible feature shape classification is *Solid Axial*. However, a solid axial part can also be produced from an extruded circle. The method of axis orientation is important, in particular for placement among cylindrical features, to determine the incremental part shape. The attributes and methods for the add-on features, BOSS and CUT are almost the same as BASE feature, except for one additional method, the INCREMENTAL SHAPE DEFINITION. In this research, the part shape evolves during part creation. The incremental shape definition algorithm is presented in detail in the following chapter.

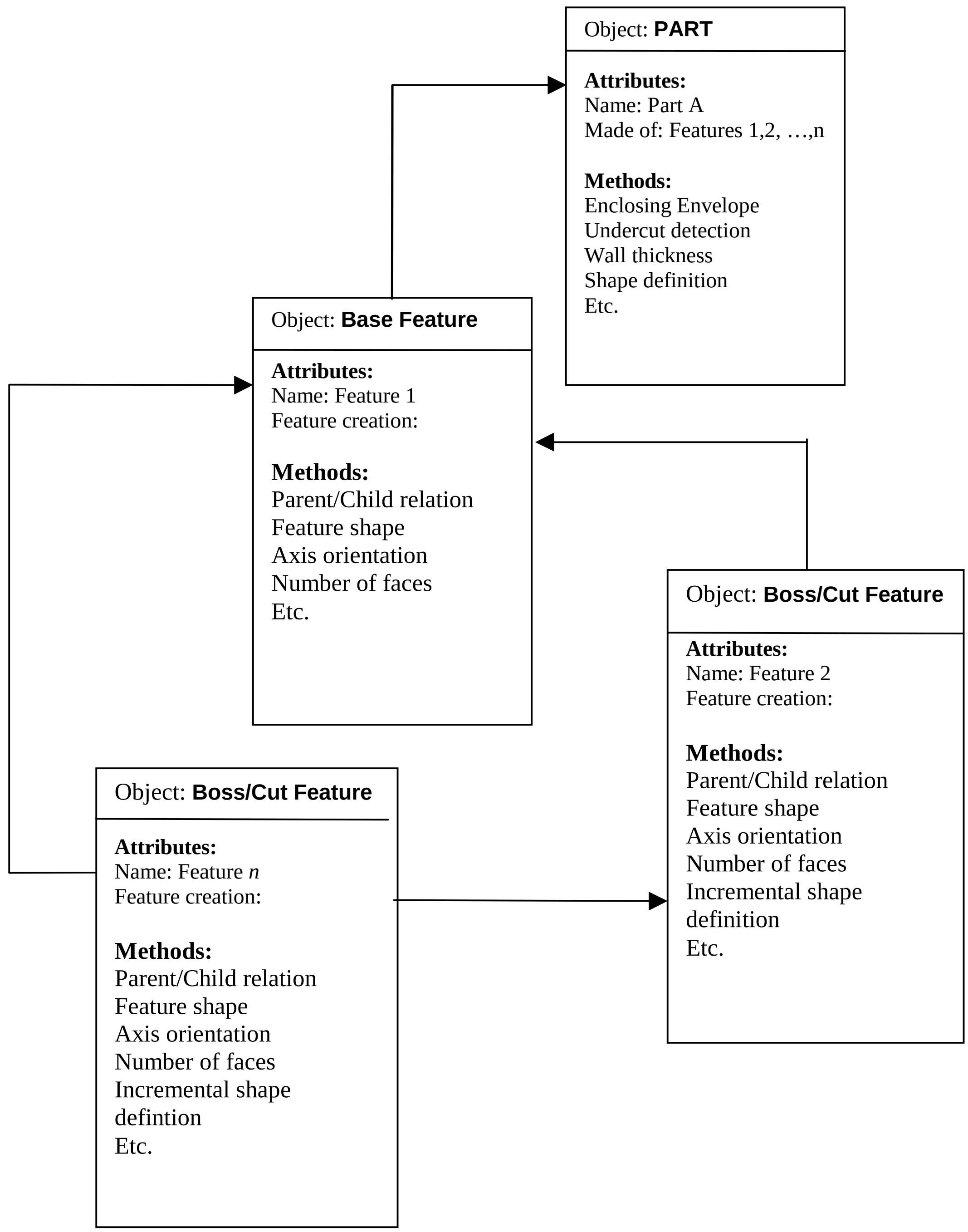


Figure 4.11 Object-oriented approach for a part model

4.5.2 Inside MaMPS

Figure 4.12 shows the MaMPS system architecture. There are three modules working together to support the decision-making task. The material selection module and the process selection module are executed separately. These modules search for the candidate processes and materials by evaluating the compatibility between the design requirements and the process or material capability. The aggregation module joins the ranked set of feasible materials and processes. The output of this system is the ranked feasible combination of materials and processes that can be used to produce a part design. A relational database management system is used to store the capability data of material and manufacturing processes. Figure 4.13 shows the data structure representation of MaMPS.

This research focuses on selecting candidate process based on mainly geometrical features. There are three types of inputs in the process selection modules: (1) geometrical features, (2) technological features, and (3) production features. Therefore, not all tables in Figure 4.13 are used in this research. The following are Tables used in the feature-based high-level process planning system:

1. Shape capability
2. Minimum and maximum wall thickness capability
3. Undercut capability
4. Length capability
5. Width capability

The shape capability table defines the relationships between process domain and the shapes that can be formed. The undercut capability defines the capability of each process domain due to the presence of undercuts. Based on two inputs, part shape and undercuts, the system is able to identify some candidate processes. The other inputs to the system are imprecise in nature. A fuzzy logic approach is then adopted in Giachetti's system to tackle design parameters that are somewhere in the boundary of different process domains.

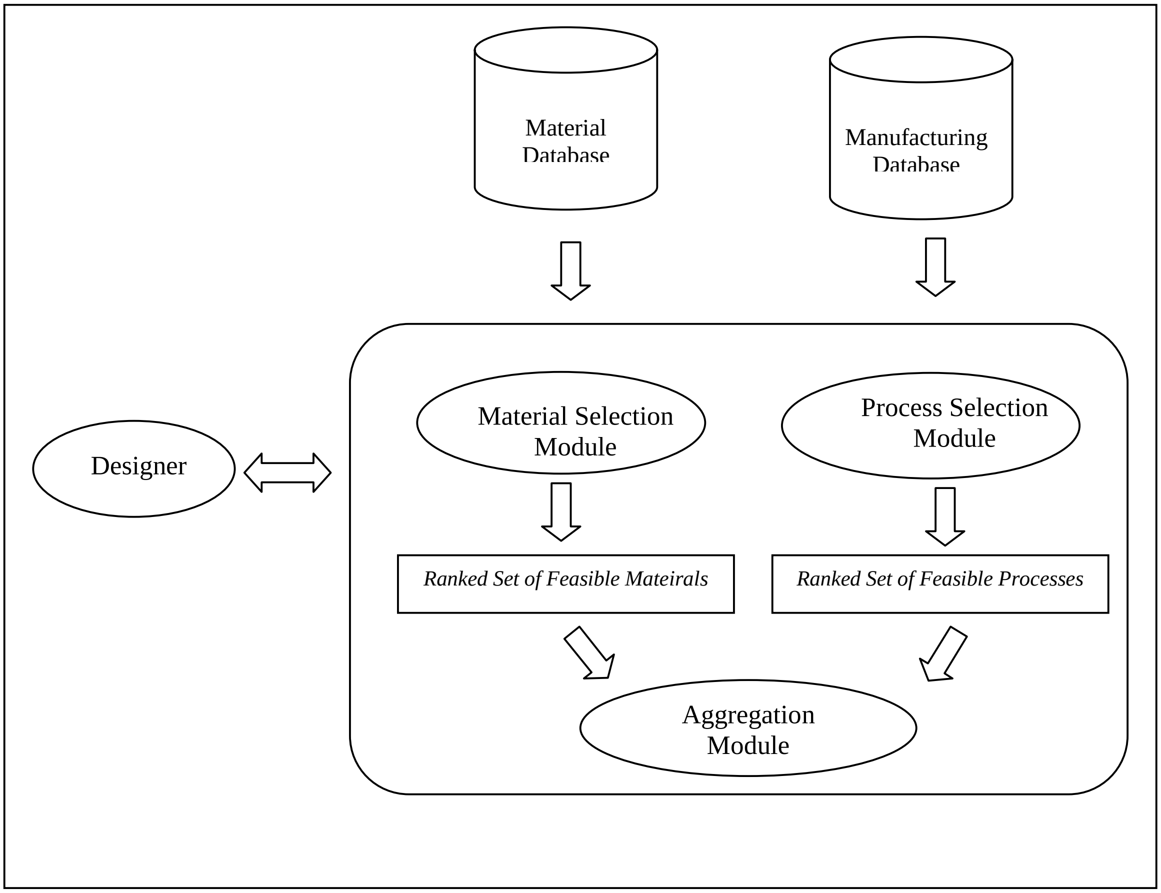


Figure 4.12 MaMPS' architecture [Giachetti 1998]

This chapter has presented the framework of FEBDAPP, starting from how a part is created using the design by feature approach and/or sketch-based modeling. Since this research is implemented in a commercial feature-based CAD system, a pure design by feature approach is not the main focus in this research. Instead, this research takes existing CAD systems that already employ the design by feature approach. It focuses on developing feature recognition algorithms for obtaining application-based features. When a hybrid system incorporating the design by feature and feature recognition approaches is used, it will reduce the complexity of feature recognition algorithms without sacrificing in the flexibility in creating a part.

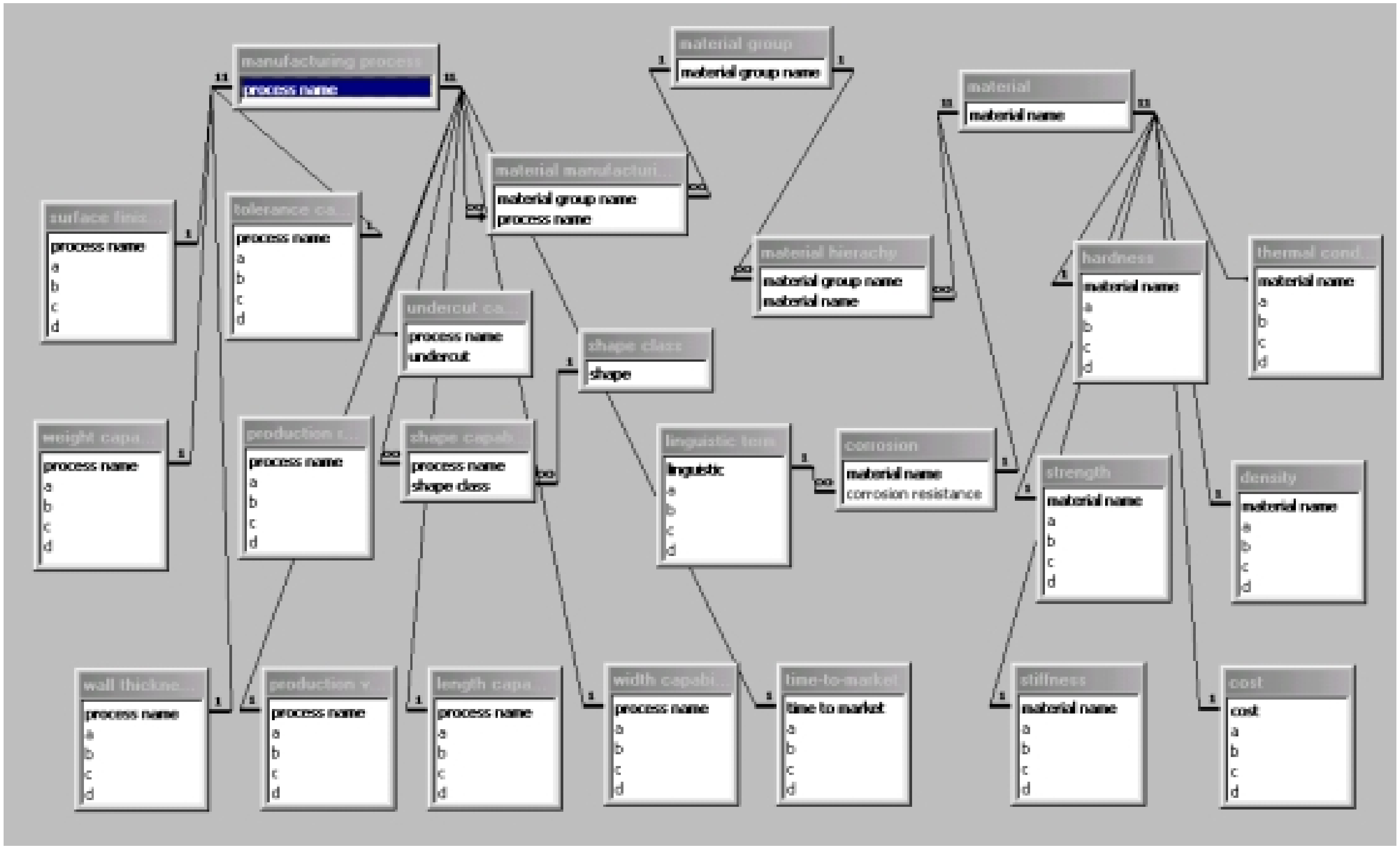


Figure 4.13 MaMPS' data structure representation

5. APPLICATION-BASED FEATURES

As discussed in the previous chapters, in order for a CAD system to be integrated with the application systems, the CAD systems should be able to deliver all information or attributes that are required by the downstream applications. In a concurrent engineering environment, a CAD system that can provide feedback to the designer in terms of cost analysis and manufacturability, for instance, is highly desirable. However, most of today's CAD systems are not able to produce a complete product or part definition due to their database limitations, and other factors. The major deficiencies of solid modelers can be summarized as follows [Shah 1988]:

- *Incomplete product definition*. Most of the CAD systems only define the nominal geometry of the part. Attributes such as tolerances and surface finish are not represented. Material specifications, surface treatments, and designer's instructions related to certain features cannot be stored, or if stored, they do not have any meaningful association. However, commercial CAD systems such as SolidWorks or Pro/Engineer do provide facilities whereby designers can give attributes to design entities.
- *Low-level product definition*. Parts are often represented in terms of low-level details, i.e. geometry and topology for B-rep and primitives and operators for CSG. We cannot simply query the CAD database to give the number of holes in a part, or to tell whether it is a through hole or blind hole. This is because higher-level information, such as form features, is missing.

It becomes obvious why the primary representation of the part built on most solid modelers cannot be used to promote applications such as process selection, process planning, manufacturability and cost analysis. It is because they do not provide the information needed for these applications. The modelers are built to enable geometric form design.

Furthermore, if feature information constitutes to the primary representation, it should be noted that they are application dependent.. Depression features such as holes, slots, steps, and pockets are meaningful in process planning for machined parts. Thin wall features are more applicable to sheet metal or injection molding than sand casting or machining. Undercut features are more of a concern in injection molding and die-casting than with sand casting. In this research, it is of great interest to develop generic feature mapping procedures that can be applied to many downstream applications, such as material selection or tooling cost estimation.

To facilitate use with downstream applications, the primary representation needs to be transformed into a secondary representation that includes *application-based features* that include design parameters required for each application. In the process selection system, for instance, design parameters may include material and process attributes. Each application developer may come up with different parameters. For purposes of this research, the focus is on design parameters that are needed for use with the existing systems. The highlights of each system in existence today are shown in Chapter 2, Table 2.2. In this research, MaMPS [Giachetti 1998] is adopted as a representative process and material selection system, since it is comprehensive in its coverage of processes and materials, and because it is available for integration with other systems. The proposed system will therefore use MaMPS for determining the candidate processes. Consequently, all the input of MaMPS is included in the secondary representation developed in this research.

Accordingly, this research focuses on the following application-based features:(1) part shape, (2) wall thickness, (3) undercuts, (4) parting surface types, and (5) part size. These five features are the most important features in determining the primary candidate processes. In Giachetti's system, there are other feature inputs, such as technological features and production features that are not geometry-based and are simply dictated by the system user.

5.1 Basic shape definition

Part shape is one important main input that needs to be defined in Giachetti's high-level process planning system. One goal of this research is to automate the manual input procedure of the MaMPS system. In such an automated system, the part shape is automatically defined from a CAD model rather than by the user. It should be noted here that part shape definition is required by numerous downstream applications such as process selection and Group Technology for Classification and Coding.

Giachetti's process and material selection system classifies the part shapes as follows:

1. Solid axial
2. Hollow axial
3. Rounded box
4. Squared box
5. Round flat
6. Square flat
7. Prismatic
8. Cone axial
9. Tank

One main task of this research has been to develop knowledge for each shape classification and then to store that knowledge in the CAD system so that the system automatically recognizes the shape of the part. In order to do that, the characteristics of each part shape need to be examined.

Solid axial

A solid axial or axisymmetric part is defined as a part that is built by revolving a 2-D sketch around an axis of revolution. In a feature-based CAD system, a solid axial part can be built in the following ways:

1. The whole part is built by revolving a sketch around an axis of revolution; and
2. The part is composed of more than one cylindrical feature, and those features must share the same axis of revolution.

Figure 5.1 shows how a solid axial part is created by either method.

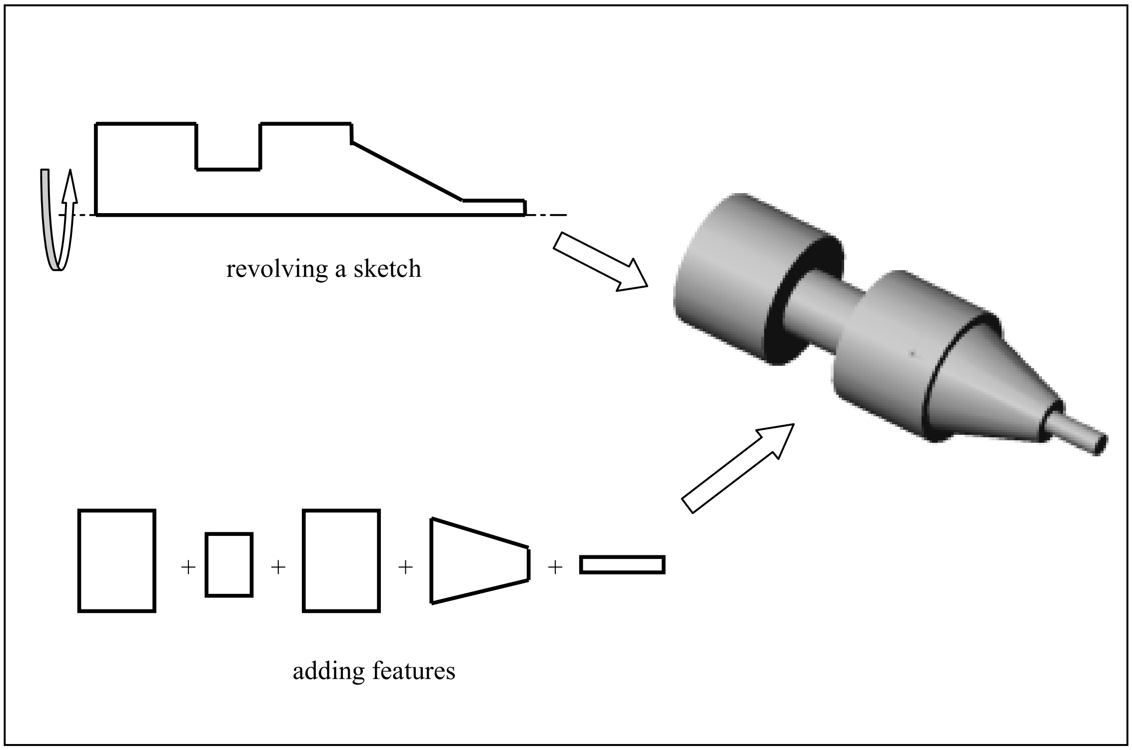


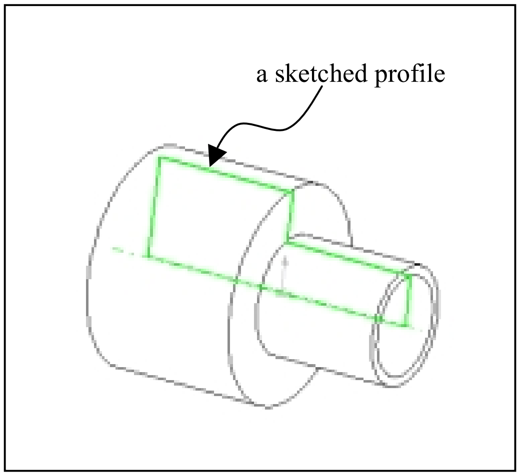
Figure 5.1 Solid axial part creation

Hollow axial

Similar to solid axial, a hollow axial part is also built by revolving a sketch around an axis of revolution. The difference is that the hollow axial part is characterized by a hollow interior region that results in uniform wall thickness. In a feature-based CAD system, hollow axial parts can be built using the following approaches:

1. Create the solid part as indicated in procedures 1 or 2 of creating solid axial parts, and then shell the solid part out;
2. Revolve a sketch of a thin wall part around an axis of revolution; and
3. Extrude a circle sketch to create a cylinder, and then create a through hole collinear to the cylinder.

Figures 5.2, 5.3, and 5.4 display how a hollow axial part is created.



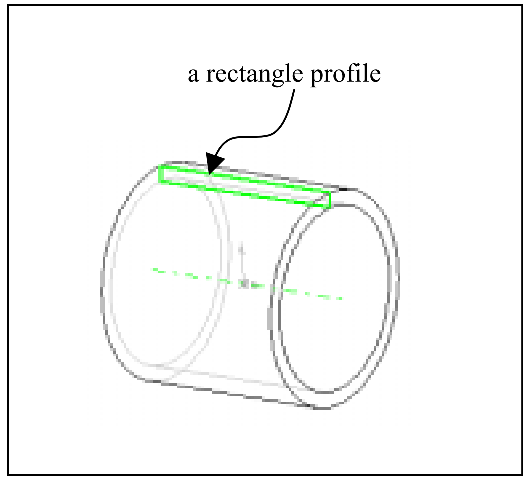


Figure 5.2 A part is created from revolving and shelling

Figure 5.3 A part is created from revolving only

In Figure 5.2, a designer creates a sketch and revolves it around an axis of revolution. A hollow part is created by shelling out the front and back faces. On the other hand, a hollow part can be created by just revolving a sketch around an axis of revolution as seen in Figure 5.3. Even though the hollow axial parts shown in Figures 5.2 and 5.3 are both created by revolving a sketch, in Figure 5.3 the axis of revolution is not on the sketch. Figure 5.4 shows another way of creating a hollow axial part. It becomes evident that in both solid axial and hollow axial parts, the parts must be composed of all rotational features.

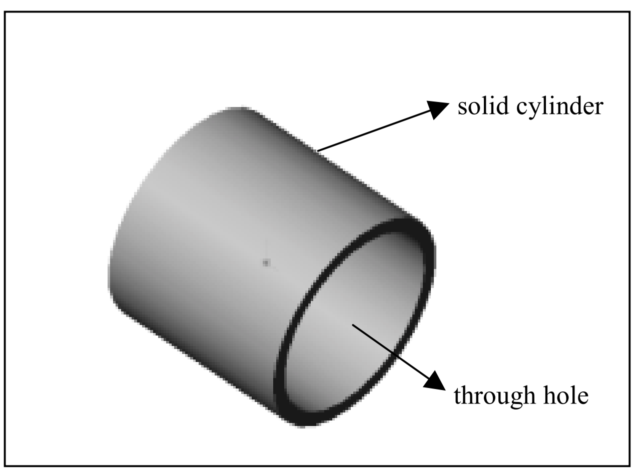


Figure 5.4 A hollow axial is created by extruding and cut-extruding

Prismatic

Prismatic parts are referred to as parts that can be produced by machining processes, i.e. milling and drilling. A prismatic part can be composed of the following features:

1. Positive features: rectangular blocks, cylindrical blocks, bosses, ribs, and extruded features
2. Negative features: holes, slots, pockets, and any extruded cut feature.

Figure 5.5 shows an example of prismatic part that is composed of both positive and negative features such as rectangular and cylindrical blocks, pockets, through holes and blind holes.

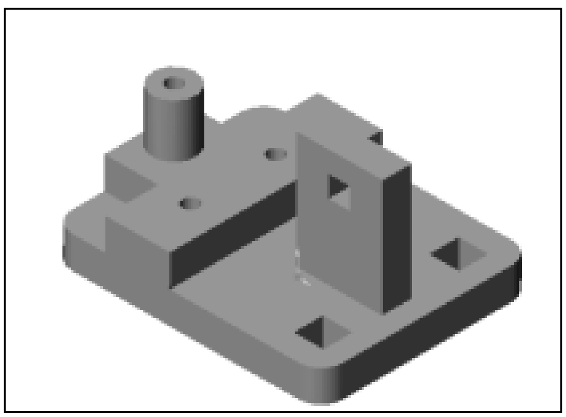


Figure 5.5 An example of prismatic part

Rounded and Squared box

A rounded box part is characterized by relatively thin walls compared with the bounding box dimensions. The other main characteristic of a rounded box part is that it has a large opening at the top face, which is basically a virtual surface. Figures 5.6 and 5.7 provide examples of how a rounded box part is built.

In Figure 5.6, a rounded box part is created using *Loft* and *Shell* operations. In the lofted part, several sketch profiles are created first and then using a guiding curve, the lofted part is built. Shelling hollows out the part, leaving open the faces that the modeler selects, and thin walls the remaining faces. It should be noted here that the selected faces should be planar.

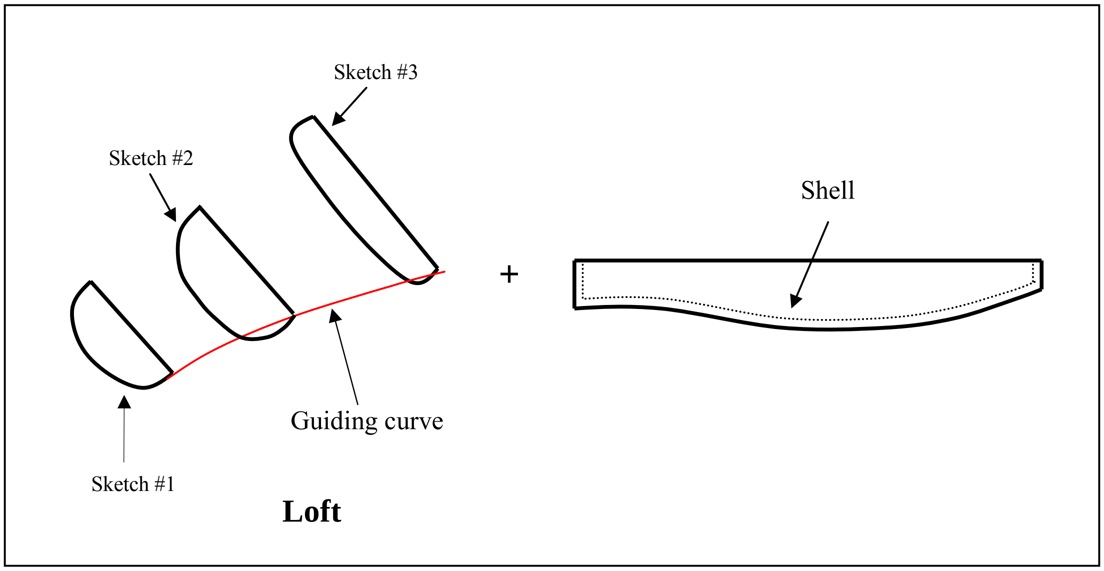


Figure 5.6 Creation of primary shape of rounded box

Rounded box parts are typically found as the lower and upper housings in electronic products. The base feature of a rounded box part can also be created via extrusion. Figure 5.7 shows an extruded cover. The rounded box part in Figure 5.7 is created by extruding a sketch profile, which is the combination of a curve and a line. The solid feature is then hollowed out to leave the top face open.

From the illustration in Figures 5.6 and 5.7, the rounded box parts are characterized as follows:

1. The base feature of rounded box parts can be created using either Loft or Extrusion operations in the CAD system.
2. Once the solid feature is created, a shelling operation hollows out the part, creating an opening on one face, and thin walls on the remaining faces.

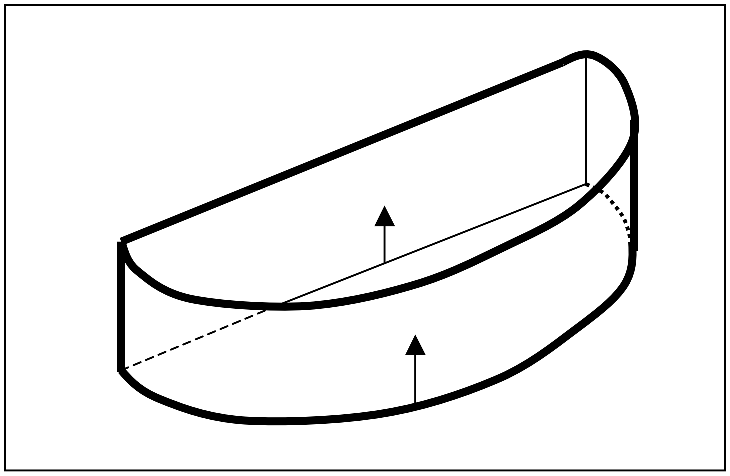


Figure 5.7 Example of the primary shape of a cover

Similar to the rounded box part, the squared box is also a thin-walled part with one pierced face. The difference is that while the rounded box has a sculptured profile, a squared box part has a rectangular profile.

Round and square flat

As implied by the name, these parts have a small thickness to length ratio. A part is typically created by extruding a 2-D sketch profile such that the depth is relatively small compared to other dimensions of the bounding box. When the profile is rectangular, the part produced is a square flat. On the other hand, when the profile is composed of closed arcs and lines, then a round flat is produced. Figures 5.8 and 5.9 show the base features of round flat and square flat parts, respectively.

Cone axial

A cone axial part is basically a solid axial part with a cone profile. Figure 5.10 shows an example of the primary shape of a cone axial part. The primary shape of a cone axial part is composed of at least 3 lines in which at least two lines are orthogonal to each other and one line that makes an angle to other two lines. This characteristic is used in developing algorithm for determining a cone profile.

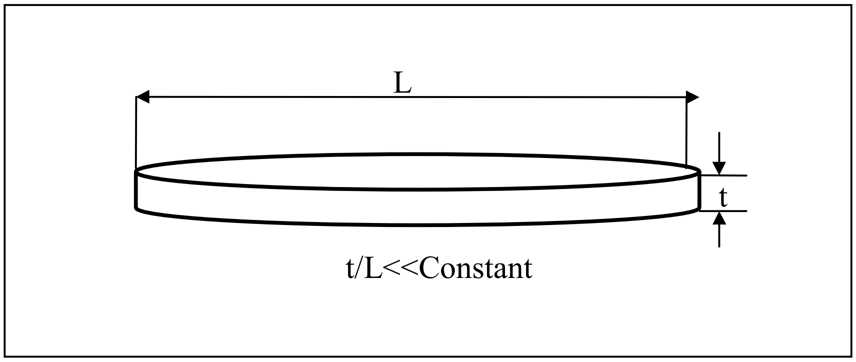


Figure 5.8 Example of a round flat primary shape

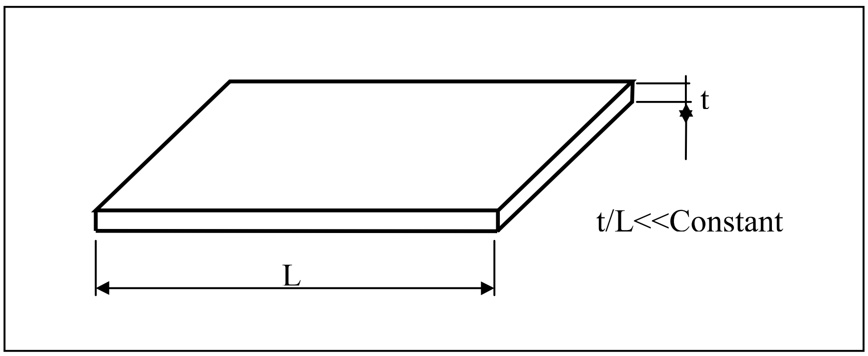


Figure 5.9 Example of a square flat primary shape

It is important to note here, that when the primary shape of a part is composed of multiple cylindrical, conical, and other rotational features that share an axis of revolution, the part is classified as a solid axial rather than a cone axial.

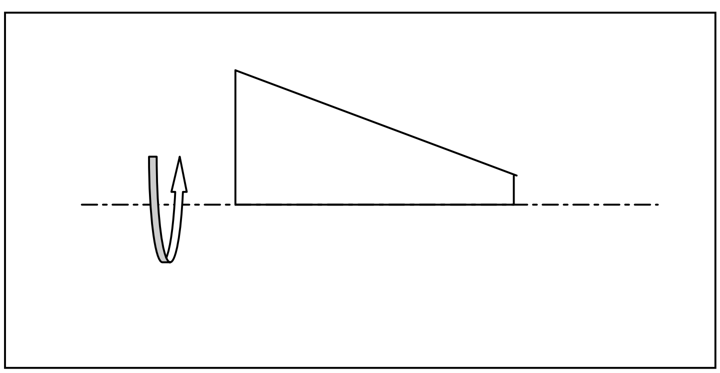


Figure 5.10 Primary shape of a cone axial part

Tank

A tank is defined as a closed-hollow part. Since it functions to contain fluid materials, there should be at least one opening, even though the openings are relatively small compared with the bounding box dimensions. The primary shape of a tank can be created using extrusion, sweeping, revolving, and lofting operations. The opening can be located on one or more faces of the base feature or any add-on features having less face area than that of the base feature. Figure 5.11 shows an example of a tank that is created by a sweeping operation. It should be noted here, that the main difference between tank and rounded box parts is in the size of opening and also the number of openings.

The characteristics of the basic shape definition can be summarized and tabulated in Table 5.1.

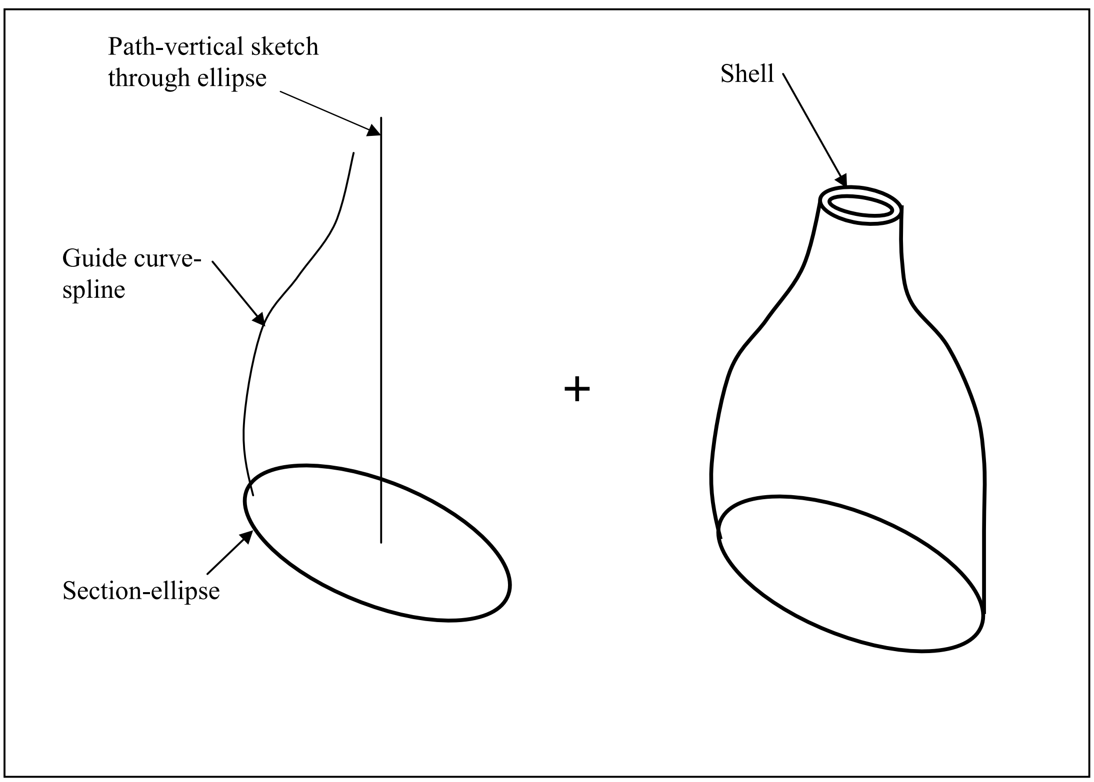


Figure 5.11 An example of the primary shape of a tank from sweeping

Table 5.1 Production rules for shape definition

Shape type and geometry	Production rule for shape recognition
1. Solid axial	If Number of feature is one Base feature is created from revolving a sketch around an axis of revolution Add-on features, if any, are relatively small compared to the base feature Then Shape type =solid axial OR If Number of feature is more than one Base feature is created either by revolving a sketch or by extruding a cylinder profile Add-on features are collinear with the base feature Then Shape type = solid axial
2. Prismatic	If There is at least one non-cylindrical feature All features are created by extruding rectangular or cylindrical profile Then Shape type = prismatic
3. Hollow axial	If Base feature is created by revolving a sketch around an axis of revolution Add-on feature is a shell OR Base feature is created by revolving a thin wall profile around an axis of revolution Then Shape type = hollow axial
4. Rounded box	If Base feature is created by either extruding a sculpture profile, lofting several sketches where at least one edge of each sketch is a line and these edges lie on one plane There is one big opening on one face resulted from shelling Then Shape type = rounded box

5. Squared box	If Base feature is created by either extruding a rectangular profile There is one big opening on one face resulted from shelling Then Shape type = squared box
6. Round flat	If Base feature is created by extruding a cylindrical profile at the depth relatively small compared to its diameter Add-on features are relatively small to the base feature Then Shape type = round flat
7. Square flat	If Base feature is created by extruding a rectangle or parallelogram at the depth relatively small compared to its length Add-on feature are relatively small to the base feature Then Shape type = square flat
8. Cone axial	If Base feature is created by revolving a cone-shaped profile Add-on features, if any, are relatively small compared to the base feature Then Shape type = cone axial
9. Tank	If Base feature is created by sweeping and revolving Add-on feature is shelling on a smaller planar face Then Shape type=tank

Table 5.1 (continued)

5.2 Incremental shape definition

The preceding section provides basic shape definition. Typically, a part is defined in several layers. This section shows how to classify complex shapes. Part shape is one of the main decision factors in process selection. In feature-based modeling, a part, P is built from a set of design features, F_i

$$\mathbf{P = \{F_1, F_2, \ldots, F_n\}}$$

In order for a CAD system to recognize the part shape, rules must be developed. For this purpose, an "incremental shape definition" technique is introduced. The purpose of using this technique is to avoid complex shape definition recognition that requires consideration of the whole part. Using the incremental technique, the part shape definition evolves as each new design feature is added.

Figure 5.12 illustrates the overall procedures that show the incremental shape definition. The designer starts with a cylindrical block, $\boldsymbol{F_1}$ as a base feature. This cylinder feature can be created by either extruding a 2-D circle sketch or revolving a 2-D rectangle sketch around an axis of revolution. At this stage, the current shape of design can be defined as either solid axial (revolving process) or constant cross-section (extruding). In Figure 5.12(b), a smaller cylinder, $\boldsymbol{F_2}$ is added. These two features are collinear since they both share the same axis of revolution. After this feature is added to the design that happens to be $\boldsymbol{F_1}$, constant cross-section is removed as a candidate shape. As a result, the current design is either a rotational part or a solid axial part. Now, suppose another smaller cylinder, $\boldsymbol{F_3}$ is added as seen in Figure 5.12(c). Since the axis of $\boldsymbol{F_3}$ does not lie on that of $\boldsymbol{F_1}$ and $\boldsymbol{F_2}$, the part can no longer be a solid axial. Instead it becomes a prismatic part.

The example in Figure 5.12 uses positive features. Now, consider when negative features are added to an existing shape. Instead of adding the cylinder $\boldsymbol{F_3}$ as in Figure 5.12.c, four blind holes are added as seen in Figure 5.13. Figure 5.14 depicts how the part shape evolves.

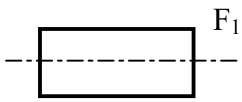
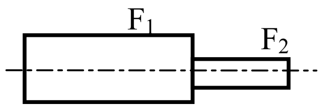
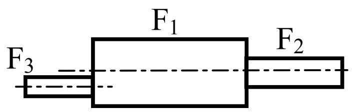
Incremental design process	Position & orientation	Current shape
a. 		***P={F_1}*** ***S= {solid axial, constant cross section}***
b. 	Collinear	***P={F_1, F_2}*** ***S={solid axial}***
c. 	Parallel	***P={F_1,F_2, F_3}*** ***S={prismatic}***

Figure 5.12 Incremental shape definition procedure

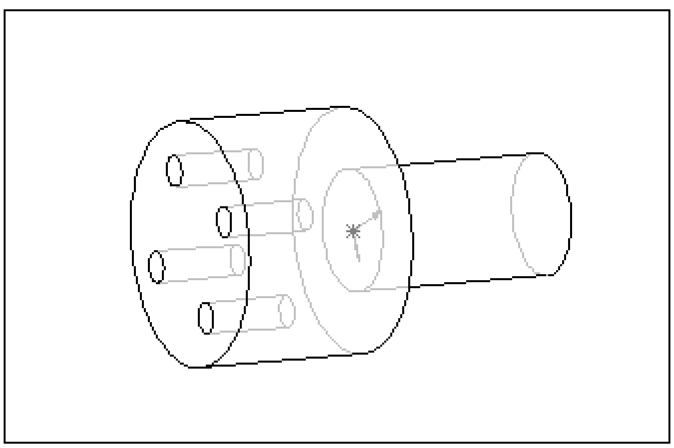


Figure 5.13 Effect of negative features

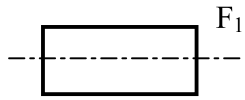
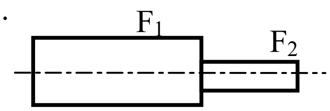
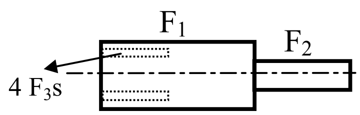
Incremental design process	Position & orientation	Current shape
a. 		***P={F_1}*** ***S= {solid axial, constant cross section}***
b. 	Collinear	***P={F_1, F_2}*** ***S={solid axial}***
c. 	Parallel	***P={F_1,F_2, 4 F_3s}*** ***S={solid axial}***

Figure 5.14 Effect of negative features in part shape definition

From Figures 5.12 and 5.14, it is shown that

(a) $S_1 = f(S_0, F_1, R_{1,}) = f(S_0)$ for the first feature

(b) $S_2 = f(S_1, F_2, R_2)$

(c) $S_3 = f(S_2, F_3, R_{3,})$

After adding *n* features,

$$\mathbf{S_n = f(S_{n-1}, F_n, R_n)}$$

where,

S_n is the shape after adding n^{th} feature

F_n is the n^{th} feature

R_n is the relationship between feature F_n and the existing design shape S_{n-1}

It should be noted here, that for rotational or solid axial parts, all features should be cylindrical features. For non-rotational parts such as prismatic parts, there must be at least one feature that is not cylindrical.

Rules for defining part shape are tabulated in the decision tables as shown in Tables 5.2 and 5.3. The first table shows the rules for transitional shape when a first add-on feature is added to the base feature. Both base features and add-on features are classified based on how they are created, i.e. extruding, sweeping, revolving, lofting, and surface thickening.

It is interesting to note, that in some cases, the shape of a feature only can define the final part, irregardless of how subsequent features are position and oriented. For example, if the base feature has a non-rotational shape, i.e. rectangular block, then solid axial or hollow axial shapes are out of consideration for the final part shape. However, if the base feature is rotational, then multiple part shape classification is possible, depending on the type of add-on features to be added, and how they are related to the base feature or existing features. In certain cases, the final shape can be determined just from the base feature. As an example, consider a lofted base feature. Once the base feature is created via lofting, the final part shape cannot be solid axial or hollow axial and choices will narrow down to rounded box or tank.

This illustrates how the incremental shape definition technique avoids the necessity to develop a more complex algorithm that must simultaneously consider all the part feature's interactions.

5.3 Extraction of low-level feature information

In order to implement the incremental shape definition rules developed in the previous section, it is necessary to extract low-level feature information from the CAD system. This information pertains to the feature object themselves as well as the relationships between features.

5.3.1 Feature Object

In this section, the pseudocode for the procedure as shown in Figure 5.15 used to identify each feature's type is presented. Once the feature types are identified, then the logic contained in the decision tables of Table 5.2 and 5.3 is applied in order to classify the overall part type.

Table 5.2 Decision table for incremental shape definition up to 2nd feature

Base Feature	Add-on Feature																							
	none	Positive Extrusion							Revolved-based Features						Lofted based Feature		Swept feature						Shell	
		Cylinder		box	O profiles												Cylinder		Box	CCSO	CCST	NCCS	no. of opening	
		Col	Non-Col		LS	LR	FS	FR	Solid Cyl.		Hollow Cyl.		Non-Cyl		FFG	FFG w/ one planar face	Col.	Non-Col.						
									Col	Non-Col	Col	Non-Col	Col	Non-Col									1	>1
Extruded-based features:																								
Solid cylinder	SA	SA	P	P	P	P	P	P	SA	P	SA	P	SA	P	O	O	SA	P	P	P	O	O	RB	HA
Solid box	P	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	SB	O
Long square	CCS	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	SB	O
Long round	CCS	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	RB	O
Flat square	SF	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*	SF*
Flat round	RF	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*	RF*
Revolved-based features																								
Solid cylinder	SA	SA	P	P	P	P	P	P	SA	P	SA	P	SA	P	O	O	SA	P	P	P	O	O	RB	HA
Hollow cylinder	HA	SA	P	P	P	P	P	P	SA	P	O	O	SA	P	O	O	SA	P	P	P	O	O	O	O
Cone	CA	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*	CA*
Other form	SA	SA	P	P	P	P	P	P	SA	P	SA	P	SA	P	O	O	SA	P	P	P	O	O	T	O
Lofted-based features																								
FFG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FFG w/ one planar face	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	RB	O
Swept-based features:																								
Solid cylinder	SA	SA	P	P	P	P	P	P	SA	P	SA	P	SA	P	O	O	SA	P	P	P	O	O	RB	HA
Solid box	P	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	SB	O
CCSO	P	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	RB	O
CCST	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FFG	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	T	O

CCS : constant cross section
NCCS : non-constant cross section
CCSO: constant cross section w/one direction
CCST: constant cross section w/two or more direction
O: Other

SA: Solid axial
SF: Square flat
RF: Round flat
RB: Rounded box
SB: Squared box

P: Prismatic
HA: Hollow axial
CA: Cone axial
T: Tank

LS: Long square
LR: Long round
FS: Flat square
FT: Flat round
FFG: free form general

*: shape remains the same assuming that add-on features are ignored

Table 5.3 Decision table for incremental shape definition after 2nd feature

Existing features, P={f1,f2,…fn}	Add-on feature																						
	Positive Extrusion							Revolved-based Features						Lofted based Feature		Swept feature						Shell	
	Cylinder		box	Other profiles												Cylinder		Box	CCSO	CCST	NCCS	no. of opening	
	Col	Non-Col		LS	LR	FS	FR	Solid Cyl.		Hollow Cyl.		Non-Cyl		FFG	FFG w/ one planar face	Col.	Non-Col.						
								Col	Non-Col	Col	Non-Col	Col	Non-Col									1	>1
Hollow axial	SA	P	P	P	P	P	P	SA	P	O	O	SA	P	O	O	SA	P	P	P	O	O	O	O
Solid axial	SA	P	P	P	P	P	P	SA	P	SA	P	SA	P	O	O	SA	P	P	P	O	O	RB	HA
Rounded box	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB	RB
Round flat	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF	RF
Squared box	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB	SB
Square flat	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF	SF
Prismatic	P	P	P	P	P	P	P	P	P	P	P	P	P	O	O	P	P	P	P	O	O	SB	O
Cone axial	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA
Tank	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

HA:Solid axial

SA:Solid axia

RB:Rounded box

RF:Round flat

SB:Squared box

SF:Square flat

P:Prismatic

CA:Cone axial

T:Tank

```
START
For each Feature, Fi
        Get FeatureCreation, FeatC

        IF FeatC  is  Positive-Extrude Then
                Get ParentSketch    ‘ get a sketch to create the feature
                Get LineSketchCount ‘ get the number of lines in a sketch
                Get ArcSketchCount   ‘ get the number of arcs in a sketch

                If ArcSketchCount=1 And OtherSketchesCount=0 Then
                        ParentSketch is a circle
                        FeatureObject is a cylinder
                ElseIf LineSketchCount=4 Then
                        For each Line
                                Get LineParams   ‘ get line information
                                Get LineVector   ‘
                        Next Line
                        Get LineRelations ‘ get relations between lines
                        If there are two lines parallel to each other AND
                                There are two lines orthogonal to two other two lines Then
                                ParentSketch is a rectangle
                                FeatureObject is a box
                        End If
                Else
                        FeatureObject is other constant-cross section
                End If
        End IF

        IF FeatC is Revolve Then
                Get ParentSketch
                Get LineSketchCount ‘ get the number of lines in a sketch

                If LineSketchCount=4 Then
                        For each Line
                                Get LineParams   ‘ get line information
                                Get LineVector   ‘
                        Next Line
                        Get LineRelations ‘ get relations between lines
                        If there are two lines parallel to each other AND
                                There are two lines orthogonal to two other two lines Then
                                ParentSketch is a rectangle
```

Figure 5.15 Pseudocodes for defining feature objects

```
                If ParentSketch is a rectangle and one line of the rectangle
                    lies on the axis of revolution Then
                    FeatureObject is a cylinder
                ElseIf ParentSketch is a rectangle and no line of the
                    rectangle lies on the axis of revolution Then
                    FeatureObject is a hollow cylinder
                End If ‘ ParentSketch
            Else
                    FeatureObject is a non-cylinder

            End If ‘ there are two lines parallel
        Else  ‘LineSketchCount=4
            FeatureObject is a non-cylinder
        End If
End IF

IF FeatC is Sweep  Then
        Get ParentSketch
        Get ParentPath
        Get ParentGuide

        If there is no ParentGuide Then

        If ParentPath is a Line Then
            If ParentSketch is a circle Then
                    FeatureObject is a cylinder
                Else If ParentSketch is a rectangle Then
                    FeatureObject is a box
                Else
                    FeatureObject is constant cross section with one
                    direction
            End If

        Else If ParentPath is a curve Then
            FeatureObject is constant cross section two or more direction
        End If
        End If

        If there is a ParentGuide Then
            FeatureObject is a free form general
        End If
End IF
```

Figure 5.15 Pseudocodes for defining feature objects (continued)

```
        If FeatC is Loft then
                Get ParentSketch
                Get ParentSketchCount
                For each Sketch
                        Get LineSketchCount
                        Get ArcSketchCount
                Next Sketch
                If there is at least one line on each sketch and each line lies on one plane
                        Then
                        FeatureObject is a free form general with at least one planar face
                Else
                        FeatureObject is a free form general
        End If
Next Feature
End
```

Figure 5.15 Pseudocodes for defining feature objects (continued)

5.3.2 Feature relationship

It becomes apparent that when adding one cylindrical feature to an existing cylindrical feature, the axis relationship between those features is important in defining the part shape. As discussed earlier, only positive features are considered in determining the axis relationship. There are four possible relationships between feature axes:

1. *Parallel*: Feature F_1 and feature F_2 are parallel if the feature axis of F_1 is parallel to the feature axis of F_2 and they are not collinear.
2. *Orthogonal*: Feature F_1 and feature F_2 are orthogonal if the feature axis of F_1 is perpendicular to the feature axis of F_2.
3. *Collinear*: Feature F_1 and feature F_2 are collinear if feature F_1 and feature F_2 have the same axis of revolution.
4. *Angular*: Feature F_1 and feature F_2 are angular if the feature axis of F_1 is oriented with the feature axis of F_2 at an angle not equal to 0, 90, or 180 degrees.

It is also of interest in this research to determine whether or not two features' axes are collinear, since a collinear relationship determines whether the part is solid axial or not. If two features are not collinear (i.e. they are parallel, orthogonal, or angular) the part can no longer be a solid axial part.

The following is the feature recognition procedure developed to determine the feature relationships in a feature-based CAD system. The procedure described here is built by adding some cylindrical features rather than revolving a sketch, since the latter case is a straightforward problem.

Problem definition. Given that a part is composed of ***n*** positive cylindrical features, determine the axial relationship between the features.

As discussed previously, one difficulty with the design by feature approach is that features must be located on planar surfaces. A reference plane must be used whenever a new feature is added to a non-planar surface. Figure 5.16 shows the cross section of a part composed of three cylindrical features, F_1, F_2 and F_3. Suppose F_1 is the base feature and F_2 is placed on one of the planar surfaces of F_1. In a feature-based CAD system, F_3 cannot be directly placed on a cylindrical surface of F_1 because it is not on a planar surface of F_1. Instead, a reference plane is needed to create F_3 that makes an angle α to F_1.

When looking at *parent-child relationships*, F_1 becomes the parent of F_2, equivalently, F_2 is the child of F_1. F_3 is not the child of F_1, but rather the reference plane P_1. Based on this illustration, the general-purpose feature-mapping algorithm for feature relationships can be stated as shown in Figure 5.17.

5.4 Wall thickness

The next important geometrical feature that needs to be derived from the CAD model is wall thickness. Wall thickness is defined as follows:

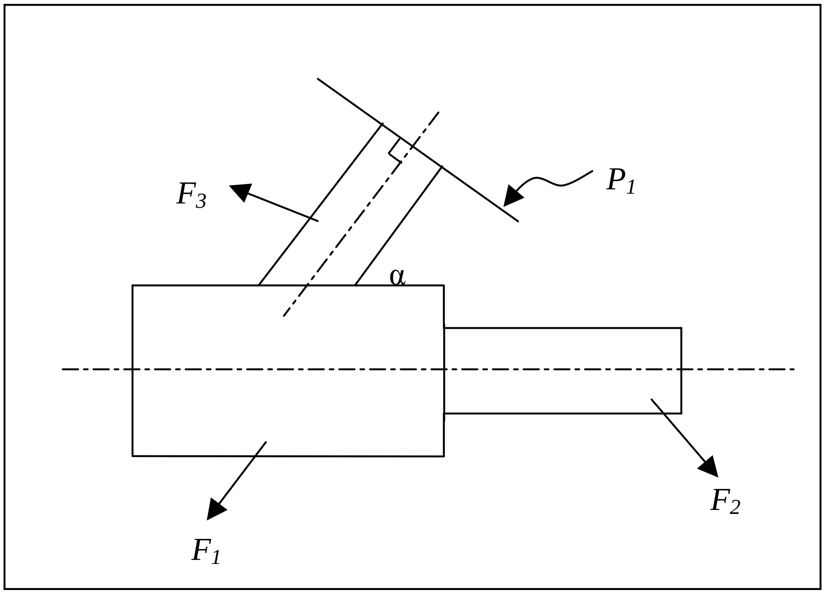


Figure 5.16 A part composed of cylindrical features

START procedure
For each Feature, F_i
    GET ParentFeature, PF_i
    If PF_i is not perpendicular to parent feature's axis Then
        F_i is *NOT COLLINEAR* with F_{i-1}
        Exit For
    End If
    GET ParentSketchType, S_1
    If S_1 is CIRCLE then
        GET SketchCenterPoint, C_i
        If $(C_i)_z = (C_{i-1})_z$ then F_i is *COLLINEAR* with F_{i-1}
        If $(C_i)_z >< (C_{i-1})_z$ then F_i is *PARALLEL* with F_{i-1}
    End If
Next
End

Figure 5.17 Pseudocodes for determining feature relationship

Definition of wall thickness. Suppose p_i is any arbitrary point on surface S_1 of an object O. When a ray with direction normal to S_1 at p_1 is projected into the body of the part, it intersects some other surface, S_2, at point p_2. The distance between p_1 and p_2 is a potential wall thickness.

Figure 5.18 shows two parallel surfaces S_1 and S_2. Suppose a vector $R_1=[1,0,0]$ is projected from p_1 into the body of the part normal to surface S_1. It will intersect a point, p_2 on surface S_2. The distance between p_1 and p_2, d, is a potential candidate, since if a ray $R_2=[-1,0,0]$ is projected from p_2, it will intersect p_1. Now, Figure 5.19 shows two surfaces, S_1 that is a planar surface, and S_2, a non-planar surface. Suppose there is a ray R_1 projected from p_1 on surface S_1. The resulted intersect point on surface S_2 is a point p_2. To check whether d_1 is a potential wall thickness, we project a ray R_2 from p_2. Since the intersect point is not p_1 but p_3, then d_1 is not a potential wall thickness.

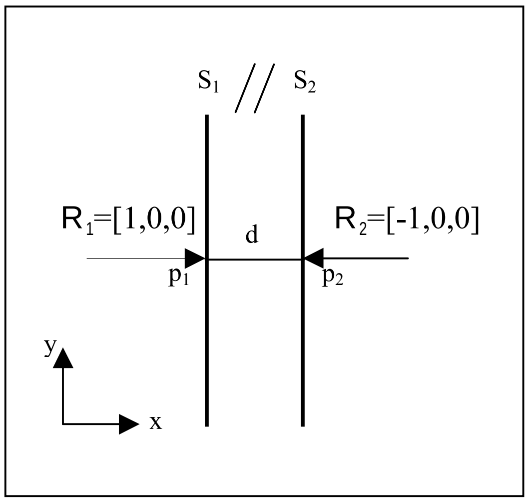


Figure 5.18 Potential wall thickness

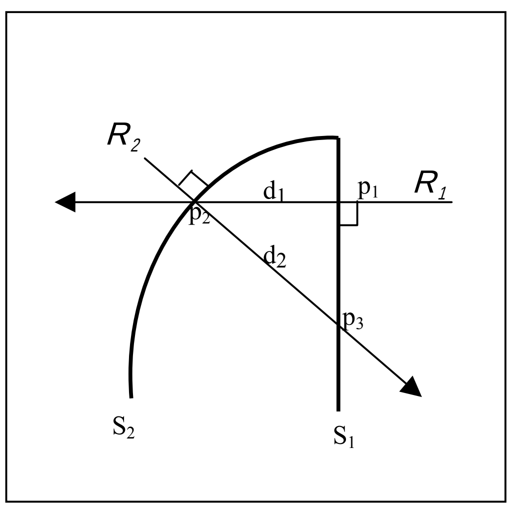


Figure 5.19 Non-potential wall thickness

Based on the illustration above, the algorithm for finding the wall thickness can be developed as follows:

Step 1. Project a ray R_1 from an arbitrary basepoint, p_1 where R_1 is the opposite normal vector to surface S_1 at point p_1.

Step 2. Find the distance d_1 that is between p_1 and the intersection point, p_2 on other surface S_2.

Step 3. Find the normal vector, R_2 to surface S_2 at point p_2.

Step 4. Using the opposite of R_2 to project this ray from point p_2

Step 5. Find the distance d_2 that is between p_2 and the intersection point p_3

Step 6. Either d_1 or d_2 is the candidate wall thickness iff $d_1 \approx d_2$

Step 7. Set the smallest candidate wall thickness as the minimum wall thickness and the largest candidate as the maximum wall thickness where the biggest candidate is equal or smaller than the smallest of the length, width, and height of the part envelope.

The algorithm above is difficult to implement on commercial CAD systems such as SolidWorks. When a ray is projected from a point on a planar surface and the normal vectors have the same direction all over the surface. However, when a ray intersects a point on a non-planar surface, it is difficult to derive the normal vectors of the intersection points, since the surface equation on which the intersect point is located cannot be easily derived. To solve this problem, surface tessellation procedures are used.

5.4.1 Surface tesselation

Parts can contain planar and/or non-planar surfaces. For planar surfaces, the base point can be located anywhere on the surface, since the direction of the normal vector of every point on a planar surface is the same. For non-planar surfaces, the direction of the normal vectors at different points will be different. In a typical CAD system, it is easier to find the normal vector of a planar surface than a non-planar surface. Because of this, this research approximates (tessellates) non-planar surfaces as a collection of triangular facets. Each facet is a planar surface, so the process of generating surface normal vectors is greatly simplified.

5.4.2 Base point selection

There are two approaches for selecting the base point from which a ray is projected. These approaches are as follows:

(1) Vertex projection approach

With this approach, each vertex of each triangle is a base point from which a ray is projected. If the CAD system's API (Application Programming Interface) does not directly provide a function call for obtaining the surface normal vector, it can be readily obtained using the cross product of vectors formed by the two edges of the facet.

(2) Center point projection approach

Instead of projecting a vector from each vertex of a facet, the center point of the facet can be used as the base point for the vector. There are two main advantages of using this approach over the vertex projection approach:

1. It is faster and requires less computation time, because it only uses one point for each facet instead of three points as in the vertex projection approach.
2. It is generic and can be applied to many CAD systems.

The center point and the normal vector of the facet can be obtained as follows:

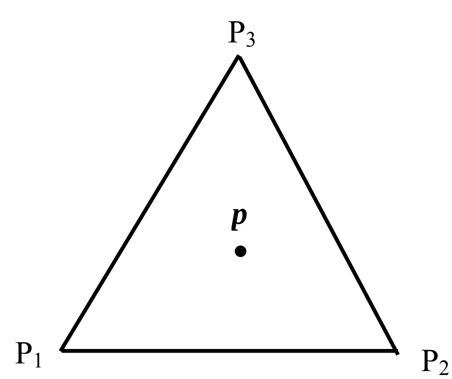


Figure 5.20 Facet $P_1P_2P_3$

Suppose the vertices of the facet are $P_1(x1,y1,z1)$, $P_2(x2,y2,z2)$, and $P_3(x3,y3,z3)$, then the center point, $\boldsymbol{p}(X_p,Y_p,Z_p)$:

$$X_p = (x1 + x2 + x3) /3$$
$$Y_p = (y1 + y2 + y3) / 3$$
$$Z_p = (z1 + z2 + z3) / 3$$

The standard equation of a plane in 3-D space is Ax + By + Cz + D = 0.The normal to the plane is the vector (A,B,C).

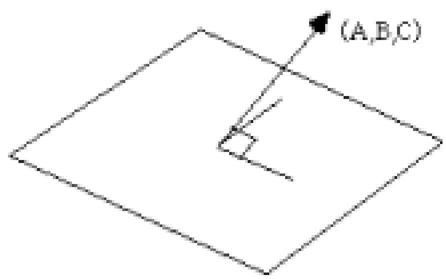


Figure 5.21 A plane and its normal vector

Given three points in space (x1, y1, z1), (x2, y2, z2), (x3, y3, z3) the equation of the plane passing through these points is given by the following determinants.

$$A = \begin{vmatrix} 1 & y1 & z1 \\ 1 & y2 & z2 \\ 1 & y3 & z3 \end{vmatrix} \quad B = \begin{vmatrix} x1 & 1 & z1 \\ x2 & 1 & z2 \\ x3 & 1 & z3 \end{vmatrix} \quad C = \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix} \quad D = - \begin{vmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{vmatrix}$$

Expanding the above gives

A = y1 (z2 - z3) + y2 (z3 - z1) + y3 (z1 - z2)
B = z1 (x2 - x3) + z2 (x3 - x1) + z3 (x1 - x2)
C = x1 (y2 - y3) + x2 (y3 - y1) + x3 (y1 - y2)
- D = x1 (y2 z3 - y3 z2) + x2 (y3 z1 - y1 z3) + x3 (y1 z2 - y2 z1)

5.4.3 Wall thickness selection

In Giachetti's system, minimum wall thickness and maximum wall thickness are the important geometrical inputs that determine the candidate processes. When the algorithm for wall thickness is applied to the whole part, the following situations may happen:

1. Uniform wall thickness, minimum wall thickness=maximum wall thickness
2. Non-uniform wall thickness, minimum wall thickness< maximum wall thickness

Situation 1 happens in a part that is characterized by constant wall thickness. Situation 2 is the most frequently encountered. It occurs when different sections of the part have different thicknesses.

5.5 Undercut detection

5.5.1 Fundamentals

It is not possible to simply query a CAD system to determine whether or not any undercut exists. Generally speaking, an undercut results when the surface normal vector for any given surface points toward the parting surface rather than away from it. Figures 5.22 and 5.23 and give an illustration of how an undercut is detected.

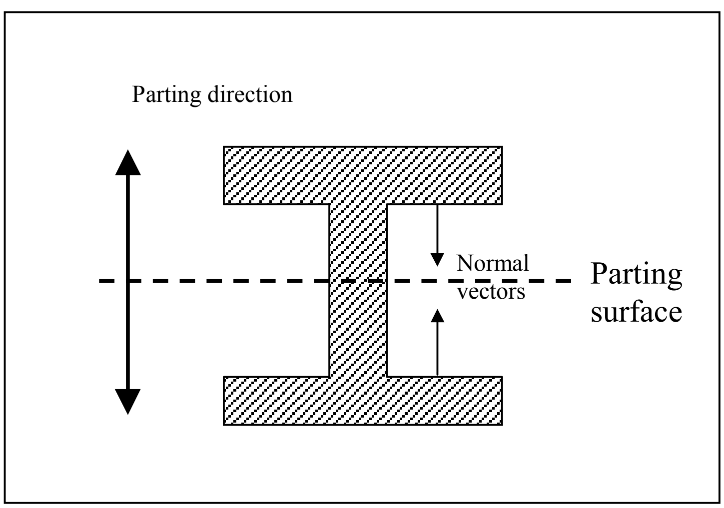


Figure 5.22 Undercuts are detected

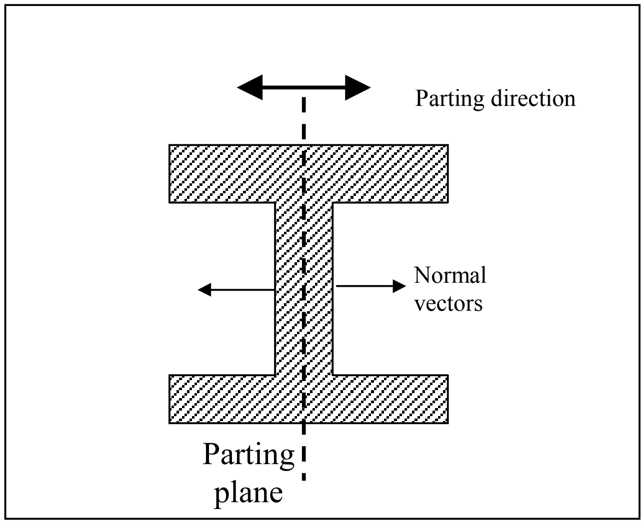


Figure 5.23 No undercuts are detected

The following algorithm uses this property of undercuts to detect undercuts in a CAD model. The undercut detection algorithm is summarized as follows:

Step 1. Specify the location of the parting surface, X or Y or Z = C(onstant). It can be parallel to X,Y, or Z axis at a specified position.

Step 2. Determine the number of faces of the part, *FaceNum*

Step 3. Determine the number of edges of each face, *EdgeNum*

Step 4. For each edge on a face, find the coordinate of each vertex, E_{ij} (x,y,z), where i=1,2, …,*EdgeNum* and j = 1,2, …,FaceNum

Step 5. For each face, find its normal vector, $V_{i,j,}$

Step 6. After obtaining the coordinates of each vertex on each face and the normal vectors of each face, an undercut can be detected according to the following rules:

Case 1: Parting surface is parallel to the X-Z plane:
If { V_{ij} *(y)* < 0 AND E_{ij} *(y)* > Y= C} OR { V_{ij} *(y)* > 0 AND E_{ij} *(y)* < Y= C} then an undercut exists

<u>Case 2: Parting surface is parallel to Y-Z plane:</u>
If { V_{ij} *(x)* < 0 AND E_{ij} *(x)* > X= C} OR { V_j > 0 AND E_{ij} *(x)* < X= C} then an undercut exists

<u>Case 3: Parting surface is parallel to X-Y plane:</u>
If { V_{ij} *(z)* < 0 AND E_{ij} *(z)*> Z= C} OR { V_k > 0 AND E_{ij} *(z)* < Z= C} then an undercut exists

The above rules only work for the case where all of a part's faces are planar. For a part with non-planar faces, the rules need to be slightly modified. Here is the algorithm for detecting an undercut for a part with non-planar faces.

Step 1. Specify the location of the parting surface, X or Y or Z = C(onstant). It can be parallel to X,Y, and Z axis at certain position.

Step 2. Determine the number of faces of the part, *FaceNum*

Step 3. Tesselate (triangulate) each face and determine the number of triangle that make up each face, *TessNum*

Step 4. For each triangle on each face, get its vertex coordinate,E_{klm} (x,y,z), where k= 1,2,3 refers to the vertex number of each triangle, l=1,2,…, TessNum and m= 1,2,…, *FaceNum*

Step 5. For each triangle on each face, get its normal vector, V_{km} (i,j,k) where k=1 to 3 and m=1,2,…, *FaceNum*

Now, an undercut is detected if any of the following rules are satisfied.

<u>Case 1: Parting surface is parallel to the X-Z plane</u>
If {V_{klm} (y) < 0 AND E_{klm} (y) > Y=C} OR {V_{klm} (y) > 0 AND E_{klm} (y) < Y=C} then an undercut exists

<u>Case 2: Parting surface is parallel to the Y-Z plane</u>
If {V_{klm} (x) < 0 AND E_{klm} (x) > X=C} OR {V_{klm} (x) > 0 AND E_{klm} (x) < X=C} then an undercut exists

<u>Case 3: Parting surface is parallel to the X-Y plane</u>
If {V_{klm} (z) < 0 AND E_{klm} (z) > Z=C} OR {V_{km} (z) > 0 AND E_{klm} (z) < Z=C} then an undercut exists

It should be noted here that the algorithm above can also be used for a part with all planar faces. In this case, a planar face is tessellated into two triangles with a common normal vector direction.

5.5.2 Planar versus Non-Planar Parting Surfaces

The algorithms from 5.3.1 require that the location of the parting surface be identified, and assume that the parting surface is planar. In some situations, a non-planar parting surface can eliminate the presence of undercuts. Figure 5.24 shows a cross-section of a part and alternative parting surface locations.

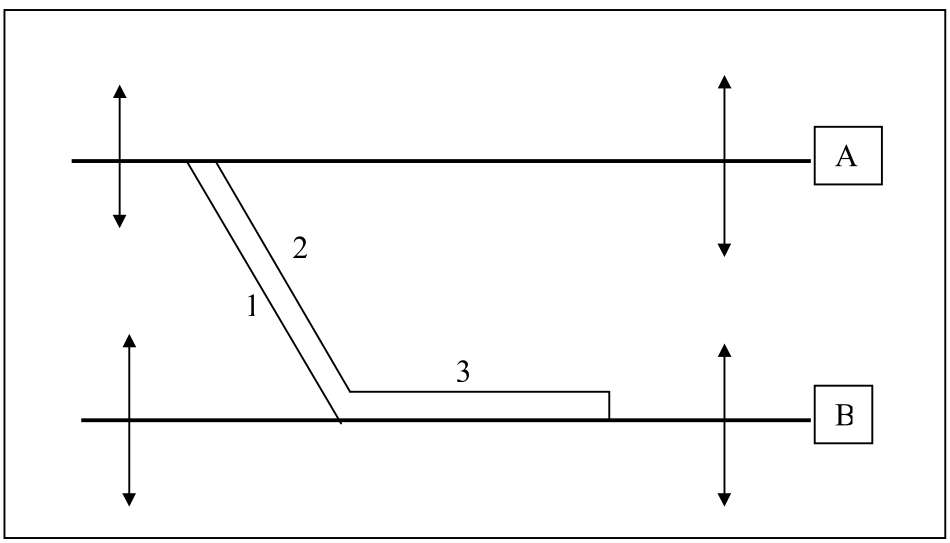


Figure 5.24 Planar parting surface

If only one planar parting surface, either A or B is considered, then undercuts are present. When parting surface A is used, surfaces 2 and 3 are the undercuts, as their normal vectors point towards A. Surface 1 is an undercut if parting surface B is used. Figure 5.25 shows how a non-planar parting surface avoids the undercuts for this part. It is of great interest to determine whether or not undercuts in a part can be prevented using a non-planar parting surface.

In general, there are four possible combinations between parting surfaces and undercuts:

1. Planar parting surface without undercuts
2. Planar parting surface with undercuts
3. Non-planar parting surface without undercuts
4. Non-planar parting surface with undercuts

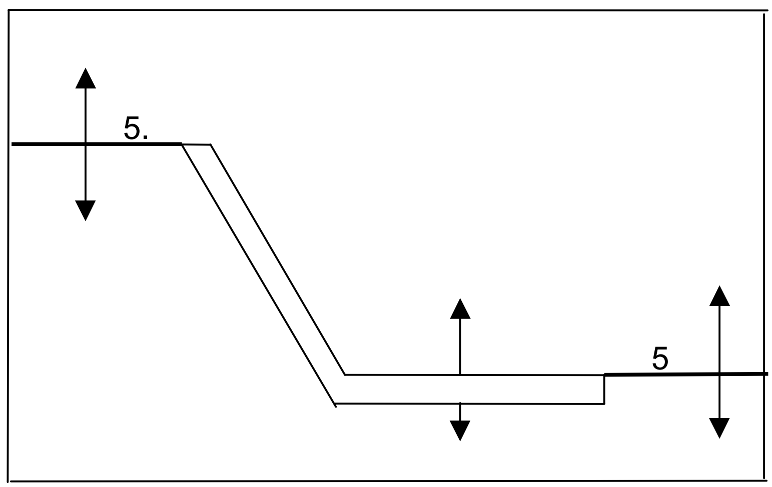


Figure 5.25 A part with no undercut and non-planar parting surface

Figure 5.24 is an example of Case #2 and Figure 5.25 is for Case #3. Non-planar parting surfaces are generally used in mold design, since fabricating molds with non-planar surface is typically less costly than fabricating molds with slides for undercuts. It is of great interest in this research to find the most desirable compromise between parting surface complexity and undercuts.

5.5.3 Determining the number of undercuts

In the previous section, the procedures for detecting the presence of undercuts have been discussed. However, users are required to define first the parting surface. This research is intended to detect any potential undercuts and to determine what type of parting surface will be used for the mold design. Before going further, it is important to understand the concept of *visibility* in order to detect any potential undercuts on a part.

Definition of visibility: Suppose ***S*** is a surface of an object, ***O***. A point ***p*** on ***S*** is completely visible if we cast a ray from *p* towards an arbitrary point far away, and the ray does not intersect the body. Repeat for all points on the surface.

Figure 5.26 illustrates the concept of visibility. If $\boldsymbol{S^1}$ is an imaginary surface that surrounds the sphere ***S***, then all points on ***S*** are visible from at least one point on $\boldsymbol{S^1}$. The practical implication of this concept on moldability in injection molding or die-casting is that if one projects a ray from a point ***p*** onto any surface ***S*** of an object ***O*** with a ray direction ***d*** that is parallel to the parting direction, then the following situations may occur:

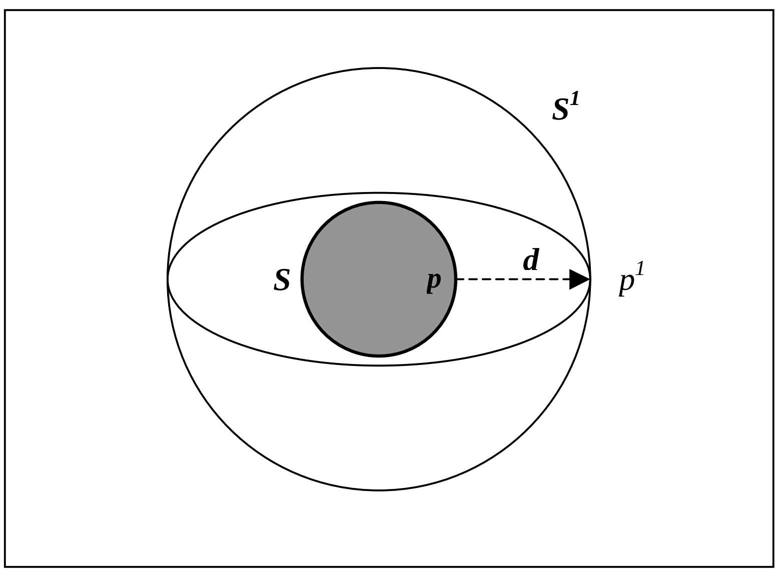


Figure 5.26 Concept of visibility

1. Surface ***S*** is a potential undercut if the ray intersects with the body of object ***O***.
2. Surface ***S*** is not an undercut; hence it is removable from the mold if the ray does not hit any point on the object ***O.***

In this research, it is assumed that the parting direction is linear. So, the parting direction is assumed to be along either the X, Y, or Z-axes only. Figure 5.27 shows a cross section in the X-Z plane with a parting direction parallel to the Z-axis. Since the parting direction is along the Z-axis, we are only interested in surfaces whose normal vectors have a non-zero z-component. We can then project a ray from any arbitrary point on that surface with a direction of either [0,0,1] or [0,0,-1] depending on the z-component of the normal vector. A projection direction of [0,0,1] is used when the z-component of the normal vectors is greater than zero. Otherwise, a projection direction of [0,0, -1] is used for the surface whose z-component of the normal vectors is less than zero.

Assuming that the part above has a constant cross section, then there are 8 surfaces to be considered. It is shown that the normal vectors of surfaces 1,5, and 6 have negative z-components.

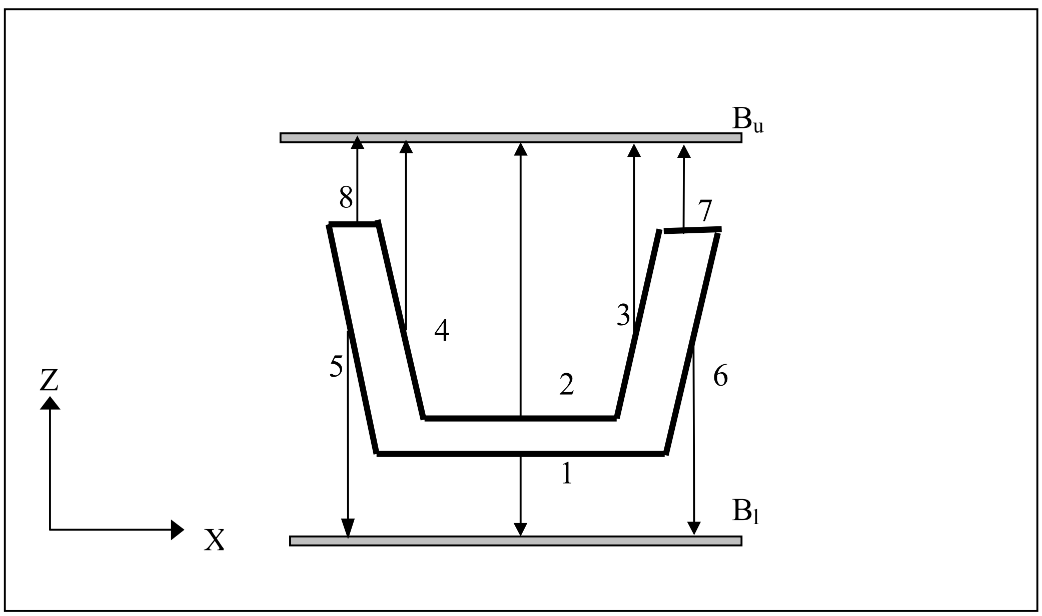


Figure 5.27 Parting surface type determination

To detect whether or not the surface is a potential undercut, then a ray is projected from points on these surfaces in the [0,0, -1] direction. It is visually shown that for any base points on these surfaces, the ray will never intersect another face of the object, and the ray will intersect the lower temporary body, $\boldsymbol{B_l}$**.** Similarly, if we project a ray from any base points on surface 2,3,4,7, and 8 with a direction of [0,0,1], the ray will intersect only the upper temporary body, $\boldsymbol{B_u}$.

Now, if we modify the part by adding a projection feature as shown in the Figure 5.28, undercuts are detected. Surface ***a*** of the projected feature is an undercut, since the ray of [0,0, -1] starting at any points on that surface intersect surface ***b*** of the part. Likewise, surface ***b*** is also an undercut, since its normal vector intersect surface ***a***.

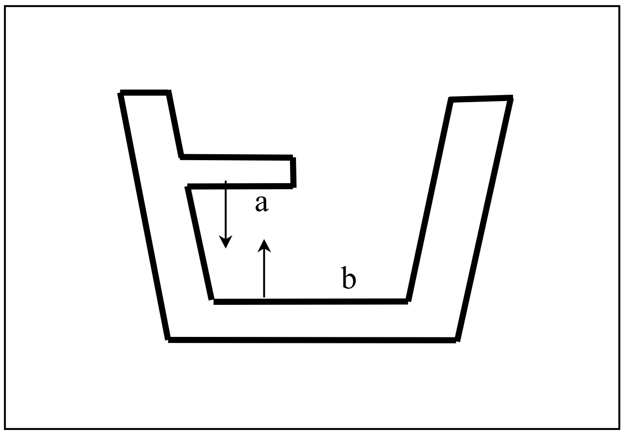


Figure 5.28 A part with an undercut

After detecting the undercuts, the next tasks are (1) to determine if the undercut is an internal or external undercut, (2) to determine the parting surface involved is planar or non-planar. Making use the concept of visibility, the algorithm for determining the type of undercuts is developed as follows:

Assuming the parting direction is along the Z-axis:

Step 0. Set $X_{left} = X_{right}=0$

(where at the end of all iterations, X_{left} is equal to the far left of the bounding box and X_{right} is equal to the far right of the bounding box)

Step 1. Get the bounding box envelope: $\{\boldsymbol{X_L, Y_L, Z_L, X_U, Y_U, Z_U}\}$

Step 2. For each parting direction, create two temporary bodies as follows:

Bounding boxes' X length is $(X_U\text{-}X_L) + 2C$

Bounding boxes' Y length is $(Y_U\text{-}Y_L) + 2C$

Bounding boxes' Z length is t

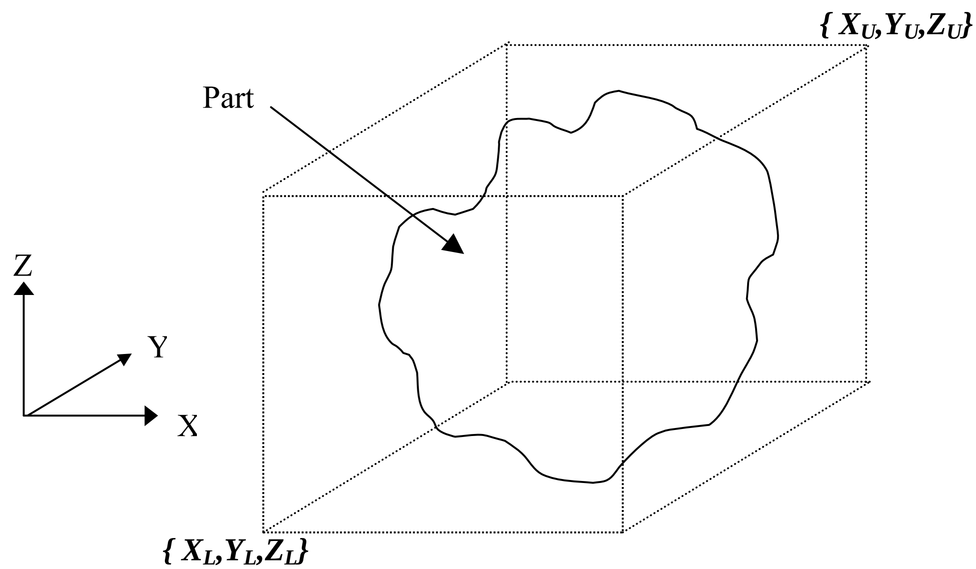


Figure 5.29 Part box envelope

The temporary body is centered around the part's bounding box as shown in Figure 5.30. Likewise, it is positioned above/below the bounding box C units as shown in Figure 5.31.

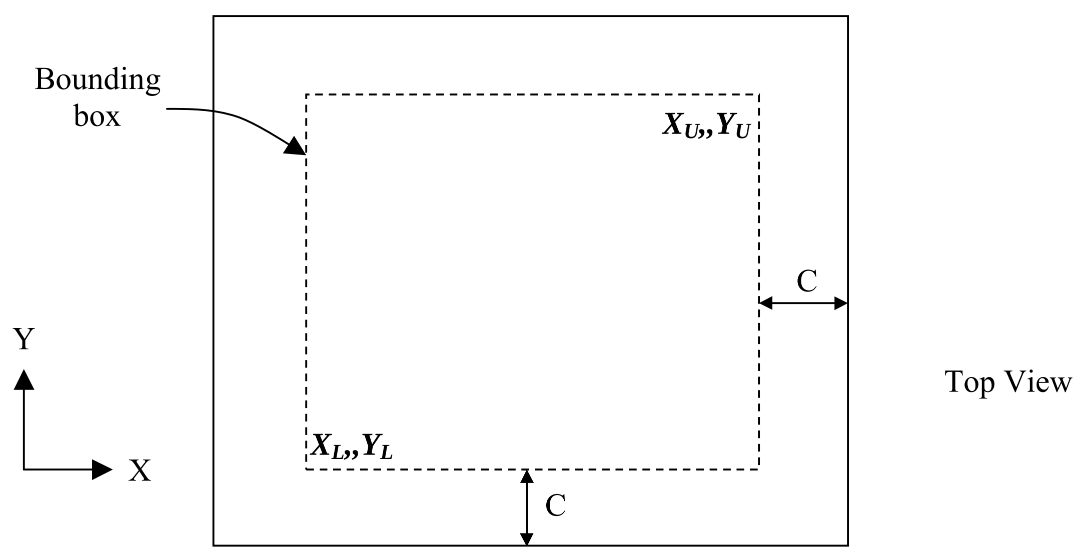


Figure 5.30 The top view of the temporary body

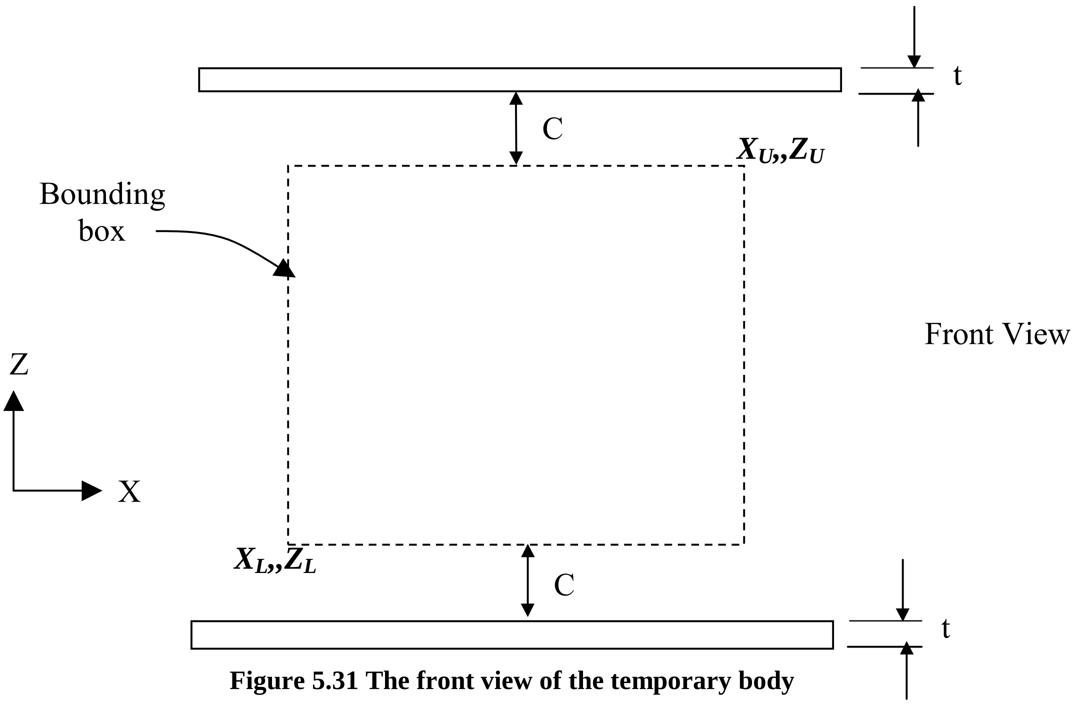


Figure 5.31 The front view of the temporary body

Step 3. Tesselate each face, F_i

Step 4. For each triangle j on face F_i, Get vertex k

Step 5. For each vertex, get the z-component of its normal vector, $\boldsymbol{Nz_{ijk}}$

Step 6. Get x-ordinate of each vertex, $\boldsymbol{Px_{ijk}}$

Step 7. Get z-ordinate of each vertex, $\boldsymbol{Pz_{ijk}}$

Step 8. If $\boldsymbol{Nz_{ijk}}$ is positive then the ray direction, d=[0,0,1]

Step 9. If $\boldsymbol{Nz_{ijk}}$ is negative then the ray direction, d=[0,0,-1]

Step 10. If $\boldsymbol{Nz_{ijk}}$ is zero then GOTO next vertex

Step 11. Project a ray from the vertex with the ray direction d

Step 12. If the ray intersects the part then F_i is an undercut

Step 13. If the parent feature of F_i is *positive features* then F_i is an *external undercut*

Step 14. If the parent feature of F_i is *negative blind-typed features* then F_i is an *internal undercut*

Endif

Step 15. If the ray hits the temporary body then F_i is *visible*

Step 16. NEXT vertex

Step 17. NEXT triangle

Step 18. NEXT Face

Step 19. End procedure

Since the types of parting surface is used in estimating the tooling cost of injection molded and die-cast parts, the algorithm for determining the types of parting surface is presented in the cost analysis section.

Figure 5.32 shows a cross-section of a box with three potential external undercuts (the through hole and two ribs), and one internal undercut. The next task is to determine a parting surface location that minimizes the number of side cores, side cavities, and form pins. If we use the planar parting surface as shown in Figure 5.32, then one side core is needed for external undercut #1, and two side cavities are needed for external undercuts #2 and #3, and a form pin is needed for the internal undercut #4.

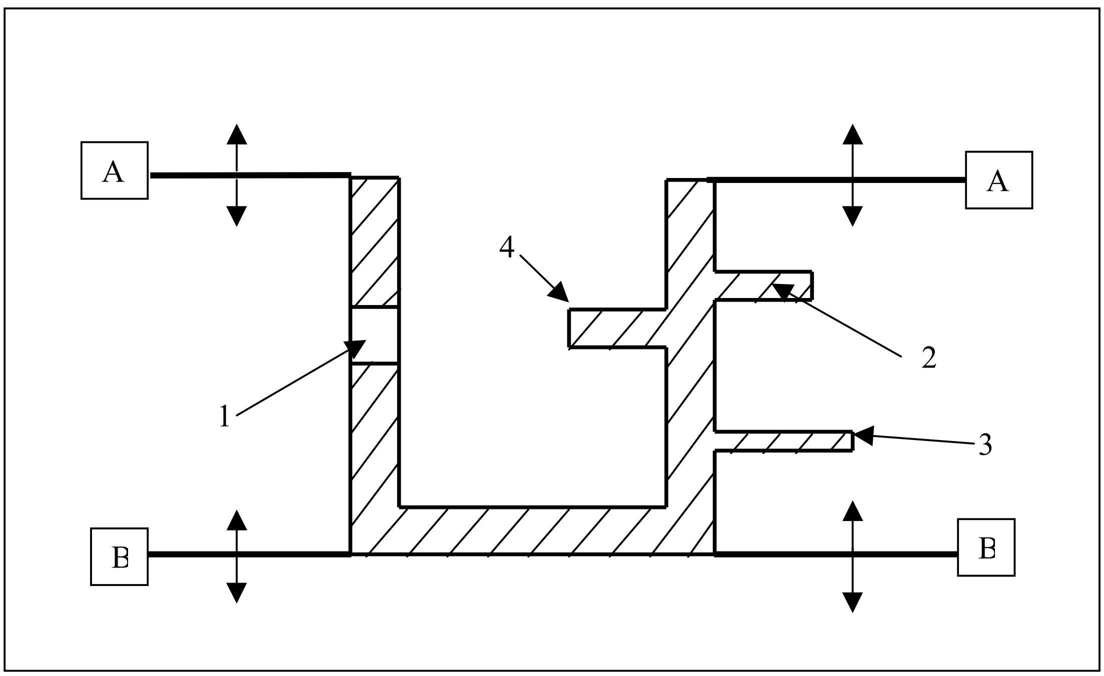


Figure 5.32 Planar parting surface with potential undercuts

When we use non-planar parting surfaces as shown in Figure 5.33, the number of undercuts can be reduced. Parting surface A is parallel to the undercut surface *a* and parting surface B is parallel to the undercut surface *b*. If parting surface A is used, feature #2 is no longer an undercut, which means the need for side cores can be reduced. Features #1 and #3 are still external undercuts, and feature #4 is still an internal undercut. When parting surface B is used, feature #3 is not an undercut; features #1 and #2 are external undercuts, and feature #4 is an internal undercut. It becomes evident that by placing a parting surface parallel and adjacent to the surface causing an external undercut, at least one external undercut can be eliminated. The presence of negative external undercuts and internal undercuts cannot be avoided through the use of non-planar parting surfaces. In this case, side cores are still needed to produce negative external undercuts such as holes, and form pins are still required to create an undercut on an inner surface.

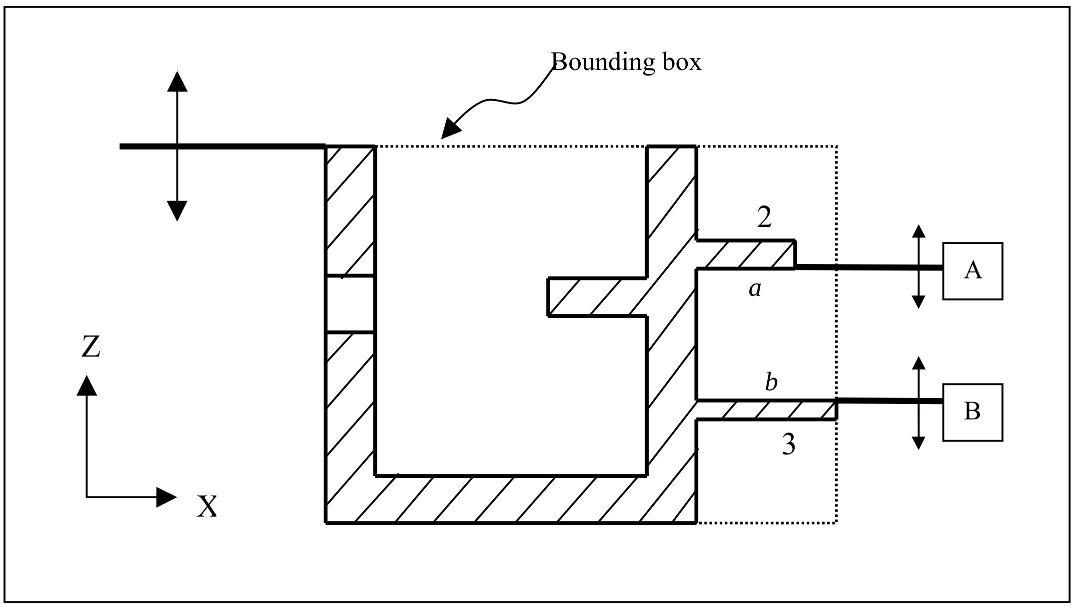


Figure 5.33 Non-planar parting surface in reducing numbers of undercuts

Problem definition: *Given a part with potential undercuts, find the parting surface such that minimizes the number of side cores, side cavities and form pins.*

Two alternative solutions can be considered as follows:

1. Use planar parting surfaces, and assume that all potential undercuts will be produced by side cores, side cavities, and form pins.

2. Use non-planar parting surfaces so that the numbers of potential undercuts is reduced.

5.6 Cost Analysis

After a list of candidate processes for a given part is obtained, it is also of great interest for designers to have a cost estimate for each process. The cost estimate at the early stage of design will help designers make decisions as follows:

- Which process is the most economical?
- Is a detailed design is worth generating or not?
- Should the part be purchased or made in house?

Procedures for calculating cost estimates for specific manufacturing processes are already available in textbooks and journal articles. At present, however, cost calculations are primarily done in a separate manual activity after the design is created. The proposed feature-based high-level process planning system can be used to automate portions of the cost estimation process In order to do this, the primary feature representation must be mapped to a secondary representation that supports cost estimation. It should be noted that the secondary representation is dependent on the application. For instance, the secondary representations for cost analysis may be different from that for manufacturability evaluation. The mapping from the primary representation to the secondary representation can be done only if the system can extract or retrieve the features and their attributes from the CAD file. This is possible if, from the beginning, the part is created using a design by feature approach.

5.6.1 Secondary Representation for Cost Analysis

Secondary representation is revisited here to highlight its relationship with cost estimation. As discussed previously, the secondary representation is application dependent. The secondary representation for injection molding tooling cost estimation may be different from that for estimating stamping tooling cost. Before doing feature mapping, the parameters included in the secondary representation for a specific application need to be defined. For example, Table 5.4 shows the secondary representation model for tooling cost of injection molding and die-casting.

There are four cost factors, C_b, C_s, C_t, and C_{dm} that need to be determined in order to estimate the tooling cost. Of these four cost factors, only C_t is geometric independent. The other three factors are geometric dependent meaning that the associated application-based features for each cost factor can be derived from a CAD model. The application-based features for the C_b factor are the number of internal and external undercuts, type of parting surface, part size, and type of mold.

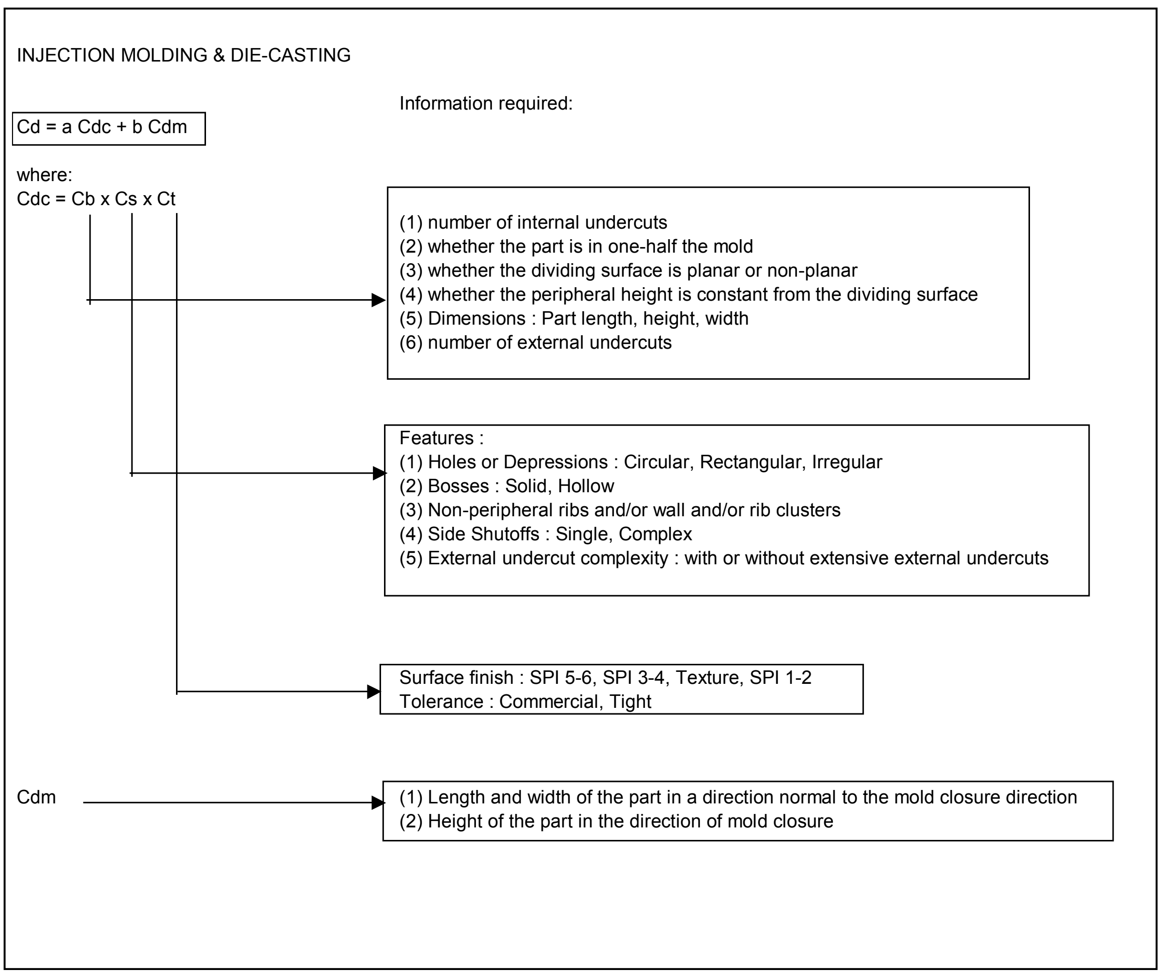


Table 5.4 Secondary Representations for Injection Molding & Die Casting Tooling Cost [Dixon 1995]

The application-based features for the C_s factor are holes or depressions, bosses, ribs, side shutoffs, and the complexity of external undercut. Features such as holes and bosses can be recognized using the algorithms for feature shape recognition presented in section 5.3.1. In this research, it is assumed that ribs and side shutoffs are predefined features. So, once it is picked from the feature library, its attributes can be directly obtained. Dixon and Poli [Dixon 1995] define that external undercuts other than unidirectional holes or depressions are considered extensive. The application-based features for the C_{dm} factor are determined based on the bounding box of the part and the parting direction. The following are algorithms for deriving application-based features for tooling cost estimation for injection molded and die-cast parts.

```
Begin procedure
Set ExtUndercutNum=0  'number of external undercut
Set IntUndercutNum=0  'number of internal undercut
FOR each Feature, FeatObj
FOR each Face, Fi
        Tesselate Fi
        FOR each vertex, (assuming the parting direction along Z-axis)
                Get the z-component of its normal vector, Nzijk
                IF Nzijk is positive THEN define the ray direction, d=[0,0,1]
                IF Nzijk is negative THEN define the ray direction, d=[0,0,-1]
                IF Nzijk is zero THEN Goto next vertex
                Project a ray from the vertex with the ray direction d
                IF the ray intersects the part THEN Fi is an undercut
                        IF the parent of FeatObj, ParentObj is a positive feature
                            THEN
                                FeatObj is an external undercut
                                ExtUndercutNum=ExtUndercutNum+1
                                GoTo next Feature
                        ELSEIF the parent of FeatObj, ParentObj is a negative blind-
                                typed feature  Then
                                ParentObj is an internal undercut
                                IntUndercutNum=IntUndercutNum+1
                                GoTo next Feature
```

Figure 5.34 Pseudocodes for determining the number of internal and external undercuts

```
                END IF
            END IF
        NEXT vertex
NEXT Face
NEXT Feature
End procedure
```

Figure 5.34 Pseudocodes for determining the number of internal and external undercuts (continued)

```
Begin procedure
FOR each visible face, VFi
        GET surface type,Si
        IF Si is a plane THEN
        'assuming parting direction is along Z-axis
                GET the normal vector of Fi, Ni
                IF Ni=[0,0,1] THEN
                        GET Edges of Fi, Eij  'j is number of edges
                        GET Vertices of Eij, Vijk ' k=1,2
                        GET the X-component of Vijk, PXijk
                        GET the Z-component of Vijk, PZijk
                        IF PXijk is smaller than the far left vertex, XLeftUp THEN
                                XLeftUp=PXijk
                                ZLeftUp=PZijk
                        IF PXijk is greater than the far right vertex, XRightUp THEN
                                XRightUp=PXijk
                                ZRightUp=PZijk
                ELSE IF Ni=[0,0,-1] THEN
                        GET Edges of Fi, Eij  'j is number of edges
                        GET Vertices of Eij, Vijk ' k=1,2
                        GET the X-component of Vijk, PXijk
                        GET the Z-component of Vijk, PZijk
                        IF PXijk is smaller than the far left vertex, XLeftDown THEN
                                XLeftDown=PXijk
                                ZLeftDown=PZijk
                        IF PXijk is greater than the far right vertex, XRightDown
                                THEN
                                XRightDown= PXijk
                                ZRightDown=PZijk
                ELSE
                        Parting Surface is non planar
```

Figure 5.35 Pseudocodes for determining the type of parting surface

```
            END IF
        ELSE IF Si is not a plane THEN
            Parting surface is non planar
        END IF
Next visible face
IF (XLeftUp is equal the far left of the bounding box, XL AND XRightUp is equal
        the far right of the bounding box, XR) THEN
        IF ZLeftUp=ZRightUp THEN Parting Surface is planar
        If ZLeftUp≠ZRightUp THEN Parting Surface is non planar
ELSE IF (XLeftDown is equal the far left of the bounding box, XL AND
        XRightDown is equal the far right of the bounding box, XR) THEN
        IF ZLeftDown=ZRightDown THEN Parting Surface is planar
        IF ZLeftDown≠ZRightDown THEN Parting Surface is non planar
ELSE
        Parting Surface is non planar
END IF
End procedure
```

Figure 5.35 Algorithm for determining the type of parting surface (continued)

```
Begin procedure
IF the number of undercuts, IntUndercutNum= 0 THEN
        FOR each visible face, VFi
            Tesselate VFi
            FOR each vertex, P
                'assuming parting direction along z-axis
                Get the z-component of its normal vector, Nzijk
                IF Nzijk is positive THEN
                    GET the z-component of P, Pz
                    IF Pz > Zmax THEN Zmax = Pz
                    IF Pz <= Zmax THEN GoTo next vertex
                    GET the x-component of P, Px
                    IF Px is smaller than the far left vertex, XLeftUp THEN
                        XLeftUp=Px
                    IF Px is greater than the far right vertex, XRightUp
                        THEN XRightUp=Px
                ELSE IF Nzijk is negative THEN
                    IF Pz > Zmin THEN Zmin = Pz
                    IF Pz <= Zmin THEN GoTo next vertex
```

Figure 5.36 Pseudocodes for determining the type of mold

```
                                GET the x-component of P, Px
                                IF Px is smaller than the far left vertex, XLeftDown
                                        THEN XLeftDown=Px
                                IF Px is greater than the far right vertex, XLeftDown
                                        THEN XLeftDown= Px
                        END IF
                Next vertex
        Next face
IF ZLeftUp=ZRightUp=Zmax or ZLeftDown=ZRightDown=Zmin THEN
        The mold is in one half
ELSE
        The mold is in both halves
END IF
End procedure
```

Figure 5.36 Pseudocodes for determining the type of mold (continued)

5.6.2 Tooling Cost

The tooling cost estimation for injection molding and die casting developed by Dixon and Poli [Dixon 1995] is adopted in this research. Since this research focuses on the early stage of design, where only configuration information is available, it is of great interest to estimate the tooling cost only. Appendix B presents the manual-based Dixon and Poli's procedures for estimating the tooling cost of injection molded and die-cast parts.

This chapter has described techniques that significantly contribute to the availability of feature mapping algorithms to convert a primary part representation into a secondary part representation. The feature mapping process is considered a major task in this research since it makes the integration of design synthesis and design evaluation possible. Since this research is intended to be implemented in a feature-based CAD environment and thus the feature mapping algorithms are developed using both low and high-level primary part representation. By using the high-level representation, the feature mapping process does not require complex algorithms. All information such as feature creation can be derived directly by taking advantage of the design by feature approach implemented in the CAD system. However, using the high-level representation alone is not sufficient to completely convert the primary part representation into a secondary part representation that is required as input to downstream applications. The feature mapping algorithms using the low-level representation such as the topology and geometry of feature need to be developed. Using a hybrid approach using high-level and low-level information of the part reduces the complexity of feature recognition algorithms without sacrificing flexibility in creating a part design.

The application-based features covered in this chapter are the geometrical inputs to the high-level process planning system. As discussed in Chapter 2, the existing high-level process planning systems are not CAD-based. It has been of great interest in this research to automate the current manual-based system by using commercial CAD systems, since a CAD system is a common tool for creating a part design. The application-based features for the tooling cost estimation for near net shape manufacturing processes such injection molding and die-casting is also covered in this research.

For the high-level process planning systems, the application-based features include undercuts, minimum and maximum wall thicknesses, part shape, and part size. For cost estimation, the application-based features include external and internal undercuts, parting surface types, feature object, and complexity of external undercuts. These algorithms have been implemented and tested, and the results of testing are presented in the next chapter.

6. IMPLEMENTATION AND TESTING

6.1 System Implementation

This chapter discusses the implementation and testing of the feature-based high-level process planning system. The implemented system is named FEBDAPP, an acronym for FEature-Based Design And Process Planning. Figure 6.1 shows the architecture of the FEBDAPP system. FEBDAPP consists of three sub-systems: (1) CAD, (2) user interface, and (3) applications. SolidWorks, the commercial CAD system, is used as the main feature-based design environment. The advantage of using SolidWorks is that it includes a complete API (Application Programming Interface) with functions or methods that can be called from either Visual Basic or Visual C++. Furthermore, SolidWorks shares the same solid modeling engine (Parasolid) as Unigraphics and several other CAD systems. Together, these CAD systems account for large user and application bases. It is expected that procedures developed in this research will be extendable to the CAD environments used in many research institutions and industrial establishments.

The API functions are essential for developing the application software. One example of API functions in this implementation involves determining the box envelope of the part. Not only do these API functions determine the information about the part geometry, but they can be also used to assign attributes to the part geometry. For instance, with API functions, we can specify the tolerance, surface finish, material specification, and other design requirements. Of course, not all API functions built in SolidWorks can fulfill the requirements of this research. If there is no API function that can be used to obtain the secondary representation, new functions or methods are created.

The user interface has been developed using Visual Basic with OLE (Object Linking and Embedding) automation. OLE is a powerful and flexible technology for sharing data between applications. Both SolidWorks and Microsoft Access are OLE-based applications, so creating a Visual Basic interface between them is relatively straightforward.

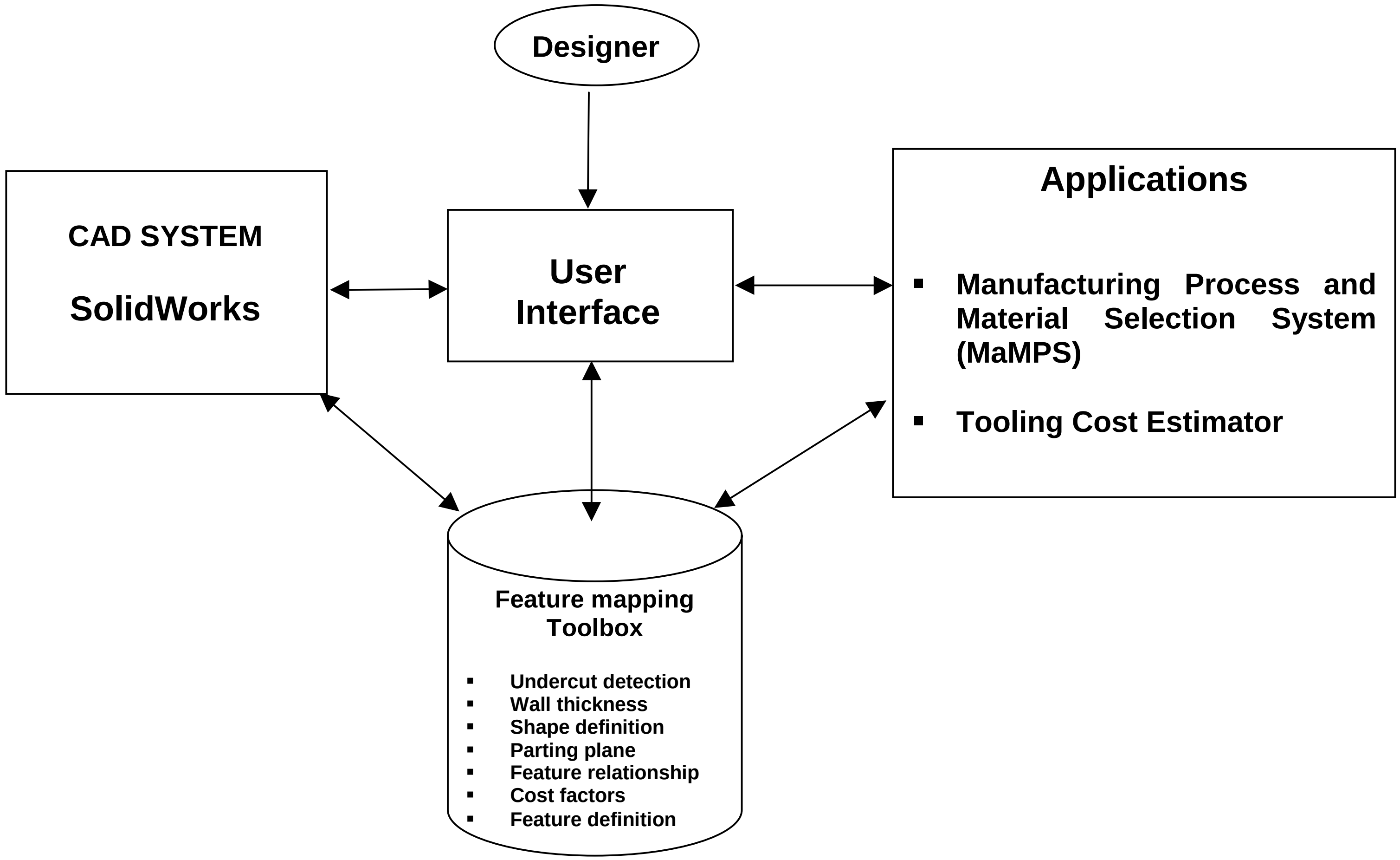


Figure 6.1 The FEBDAPP system's architecture

As discussed in Chapter 4, three levels of objects are used in extracting the application-based features. These levels are PART, FEATURE, and FACE. SolidWorks provides methods or functions to extract information from a CAD model for each object level. At the PART level, information such as the number of features can be obtained. At the FEATURE level, the built-in methods can do things such as finding the parents of each feature, check whether one of the parents is a sketch, determine the number of arcs or lines, and determine how a feature is created. Based on this information, one can obtain feature attributes such as feature object, feature shape, and feature relationship. At

the FACE level, SolidWorks provides us with the basic methods needed to triangulate an object, get the vertices of each triangle, and get the normal vectors at each vertex. As discussed in Chapter 4, the built-in API functions provided in the commercial systems, give only the explicit type of information. Nevertheless, the existing collection of the API functions can be used to implement the feature mapping algorithms developed in Chapter 5 to derive application-based features.

6.2 Part Shape

Defining the part shape automatically from the CAD model is considered one of the most challenging tasks in this research. Taking advantage of design by feature approach, an incremental shape definition technique has been implemented and tested. For example, the part shown in Figure 6.2 has three main features: (1) Base-Extrude, (2) Boss-Extrude, and (3) Cut-Extrude. SolidWorks basically names the features based on how they are created. In this case, all features are created using extrusion. Once the part is completely built, using algorithm presented in Chapter 5 to recognize the feature and the incremental shape definition technique, the part shape can be identified. The algorithm for part shape definition works as follows:

The shape of the part shown in Figure 6.3 evolves during the part creation. Since the feature is created by extruding a closed six-line profile, it is defined as either a prismatic or a constant cross section (CCS). At this stage, the part is defined also as a prismatic or a constant cross section.

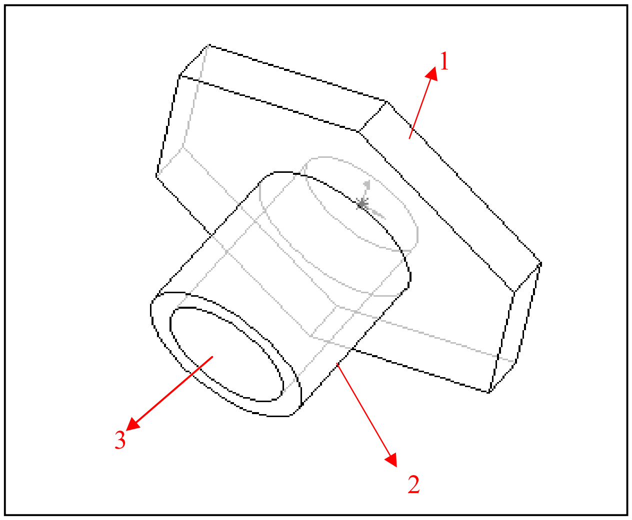


Figure 6.2 A prismatic part

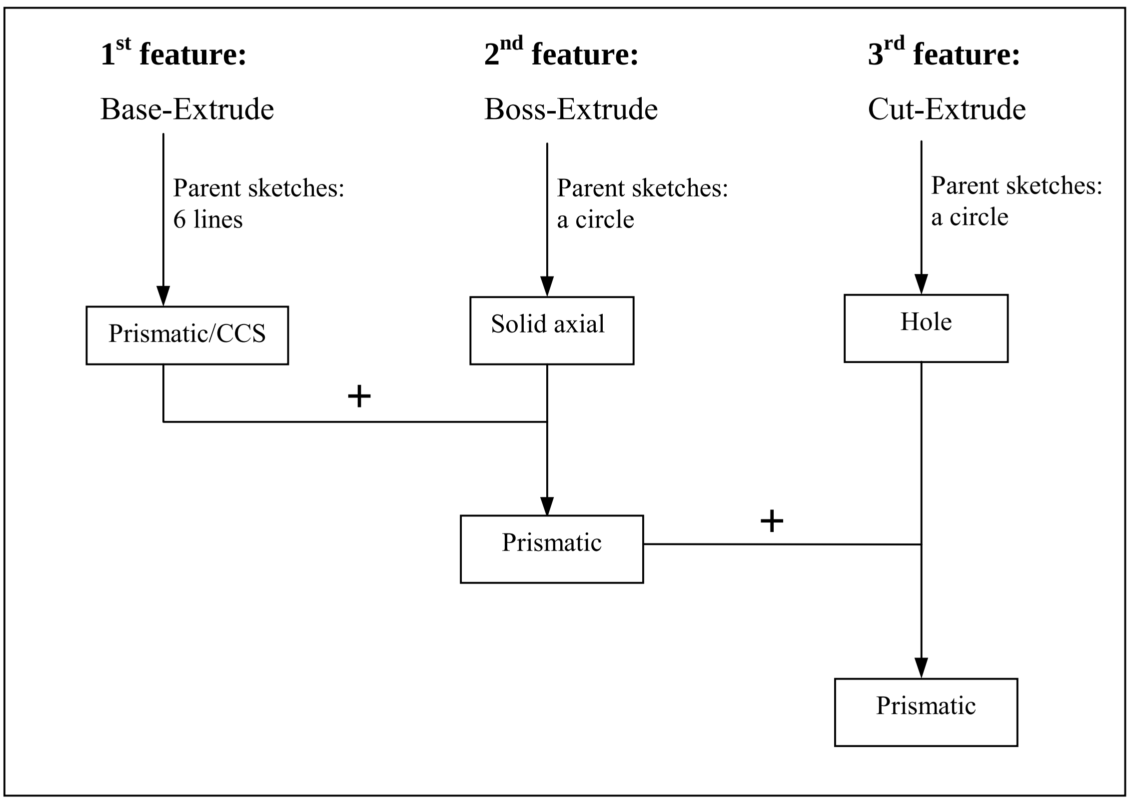


Figure 6.3 Incremental shape definition

The second feature is a cylindrical solid axial feature, since it is built from extruding a circle profile. After adding the 2^{nd} feature to the 1^{st} feature and using the relations developed in the decision tables presented in Chapter 5, the new part shape is identified as prismatic. Since the 3rd feature is a negative feature, its addition to the existing part does not change the part shape. Therefore, the final part shape is defined as prismatic.

Figure 6.4 shows a solid axial part that is built by revolving a sketch around an axis of revolution. In the SolidWorks' feature manager, the part is basically composed of only one main feature that is the base feature, called "Base-Revolve". The search for shape definition of this part is very straightforward, since the base feature is created from revolving a sketch around an axis of revolution and one line of the sketch lies on the axis of revolution. A solid axial part is then defined.

As discussed in Chapter 5, the part in Figure 6.4 can also be built by adding several features. If this is the case, the algorithm checks the relationship between features, if the feature relationships are all collinear, then the part is a solid axial part.

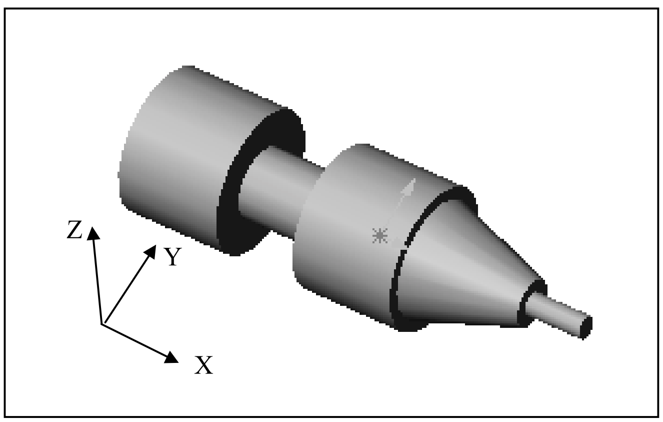
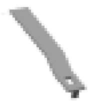
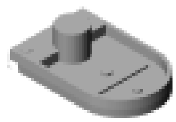
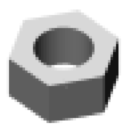
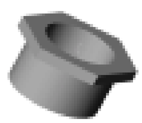
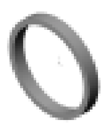
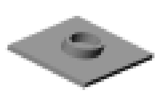
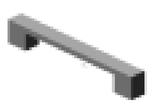
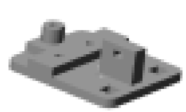


Figure 6.4 A solid axial part

6.3 Wall thickness

Chapter 5 described two procedures for finding wall thickness that differed in how rays are projected. Several parts have been tested to verify the center-point projection and vertices-point projection approaches. The results are tabulated in Table 6.1.

Table 6.1 Comparison of vertices and center shooting technique in wall thickness determination

Part	Actual results		Center Point		Vertex	
	Min (m)	Max (m)	Min	Max	Min	Max
	2.88E-3	2.896 E-3	2.88E-3 (3.08%)	2.896E-3 (3.63%)	2.89E-3 (3.43%)	2.95E-3 (5.6%)
	2.032E-3	3.81E-3	2.032E-3	3.81E-3	2.032E-3	3.81E-3
	2.69E-3	2.69E-3	2.77E-3 (3.86%)	2.77E-3	2.69E-3 (0.86%)	2.72E-3
	1.345E-3	1.63E-3	1.311E-3 (2.53%)	1.626E-3 (0.25%)	1.346E-3 (0.07%)	1.626E-3 (0.25%)
	0.01	0.01	9.65E-2 (3.5%)	1.037E-3 (3.7%)	9.99E-3 (0.01%)	0.01
	0.005	0.01	5.23E-3 (4.6%)	9.99E-3 (0.01%)	0.005	0.01
	0.005	0.01	0.005	0.01	0.005	0.01
	6.35E-3	1.016E-2	6.298E-3 (0.8%)	1.016E-2	6.35E-3	1.016E-2

In general, the vertex-projection technique gives a better result in term of accuracy than the center-point projection technique. When two surfaces are parallel, vertex projection technique gives exact results.

The error in the center-shooting technique is caused due to the tessellation of non-planar surface. Figures 6.5 and 6.6 illustrate this situation. In Figure 6.5, suppose E_1 and E_2 represent the 2-D representation of wall surfaces. Let $\boldsymbol{p}$ be the base point for the ray-projection. Due to the tessellation of the inner surface, here represented with E_1, the base point shifts to $\boldsymbol{p^1}$. Due to this tessellation, the actual thickness increases from t to $t^1 = t+d$.

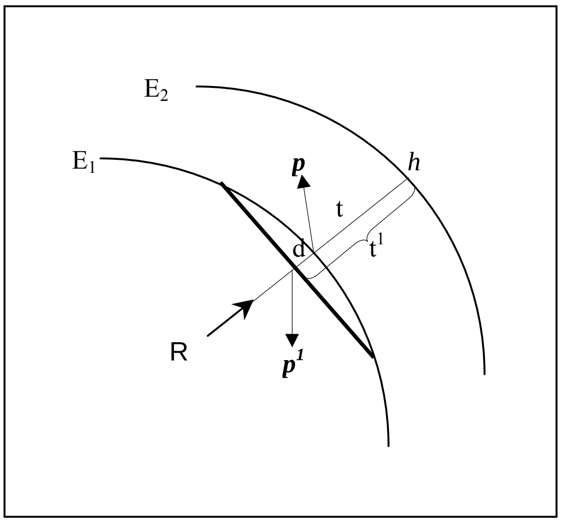


Figure 6.5 Error from the inner surface

Figure 6.6 shows a ray R is projected at p, which is now on the outer surface E_2. After tessellation, the base point $\boldsymbol{p}$ shifts to $\boldsymbol{p^1}$ and the thickness decreased from t to $t^1 = t-d$.

6.4 Parting surface and undercuts

In Chapter 5, an algorithm for determining the parting surface types has been presented. When the algorithm is implemented in SolidWorks, there is a problem in detecting the visibility of each surface.

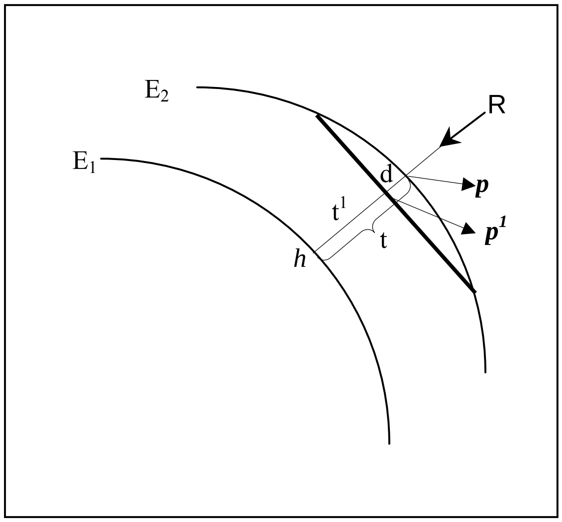


Figure 6.6 Error from the outer surface

Figure 6.7 illustrates the problem. Suppose a ray R is projected at base point, p_1. The direction of ray R is along the edge e_1. Even though surface S_1 is visible, which means that there is no interior body that blocks any rays from outside, instead of intersecting at p_2 on the temporary body, the ray intersects at point p^*, which is on the interior body. The rules say that if a ray intersects an interior body of the part, then the surface is not visible. In order for the ray to avoid intersecting a point on the interior body, then the triangle needs to be modified.

Each basepoint, which is the vertex of the triangle, is shifted a little bit to inside the triangle. To make sure that the new basepoint is still on the surface of triangle, the old basepoint is shifted in the direction of resultant of two edge vectors originating from that basepoint. Figure 6.8 shows illustrates this shift.

The resultant vectors, R_1, R_2, and R_3 can be expressed as:

$$R_1 = \begin{bmatrix} (x_2-x_1)+(x_3-x_1) \\ (y_2-y_1)+(y_3-y_1) \\ (z_2-z_1)+(z_3-z_1) \end{bmatrix} = \begin{bmatrix} (x_2+x_3-2x_1) \\ (y_2+y_3-2y_1) \\ (z_2+z_3-2z_1) \end{bmatrix}$$

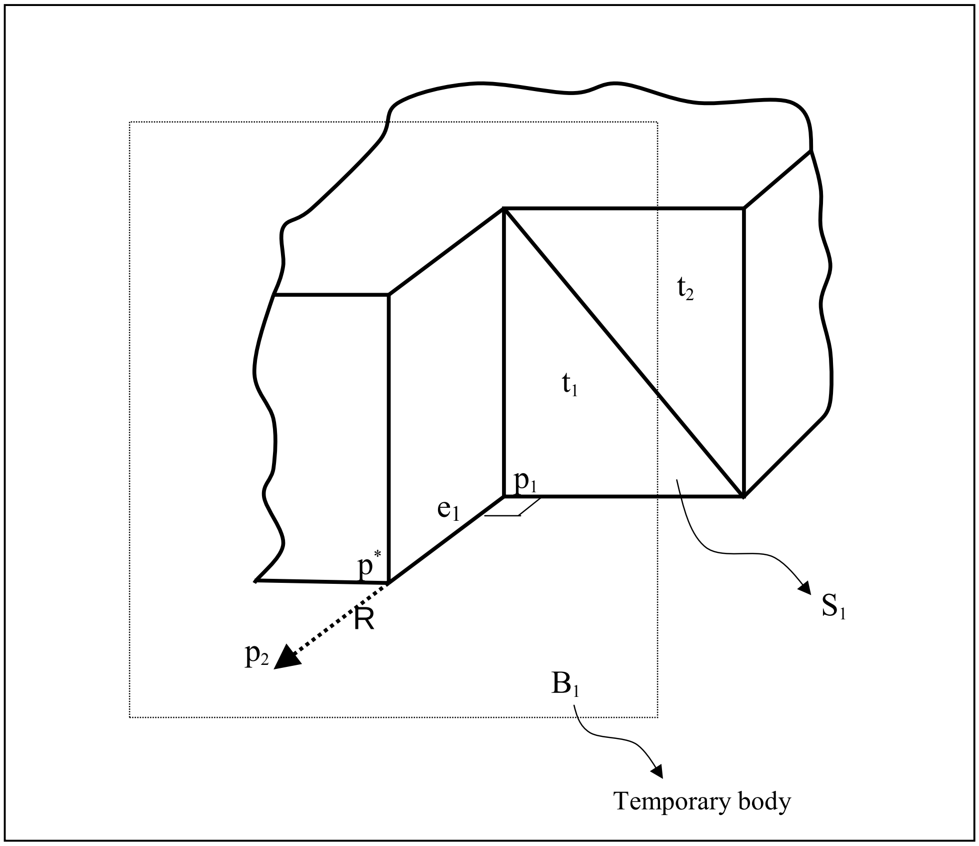


Figure 6.7 The problem in determining the surface visibility

Similarly,

$$R_2 = \begin{bmatrix} (x_1 + x_3 - 2x_2) \\ (y_1 + y_3 - 2y_2) \\ (z_1 + z_3 - 2z_2) \end{bmatrix}$$

and

$$R_3 = \begin{bmatrix} (x_1 + x_2 - 2x_3) \\ (y_1 + y_2 - 2y_3) \\ (z_1 + z_2 - 2z_3) \end{bmatrix}$$

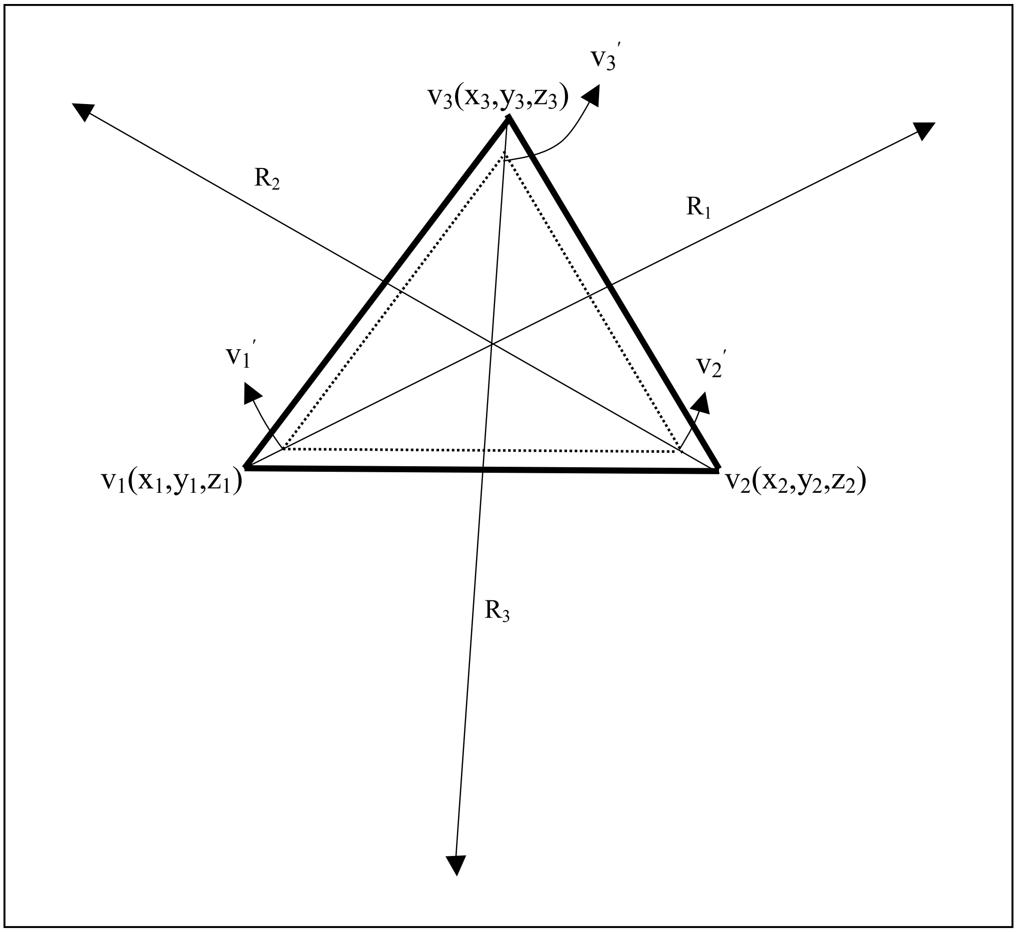


Figure 6.8 Shifted basepoints

The shifted basepoints can be expressed as:

$$v_i' = v_i + \alpha R_i \text{ , } i = 1,2,3$$

As discussed in the previous chapter, there are four possibilities of parting surface and undercuts combination:

1. Planar parting surface with no undercuts
2. Planar parting surface with undercuts
3. Non-planar parting surface with no undercuts
4. Non-planar parting surface with undercuts

6.4.1 Planar parting surface with no undercuts

Figure 6.9 shows an example of a part with planar parting surfaces and without undercuts. First, parting direction along each axis, X,Y, and Z is used to check the surface visibility. When a parting direction along X-axis is used, two faces, f_1 and f_2 are detected as potential undercuts. Parting directions along Y and Z axes do not detect any undercuts. So, parting directions in these two axes are candidate parting directions.

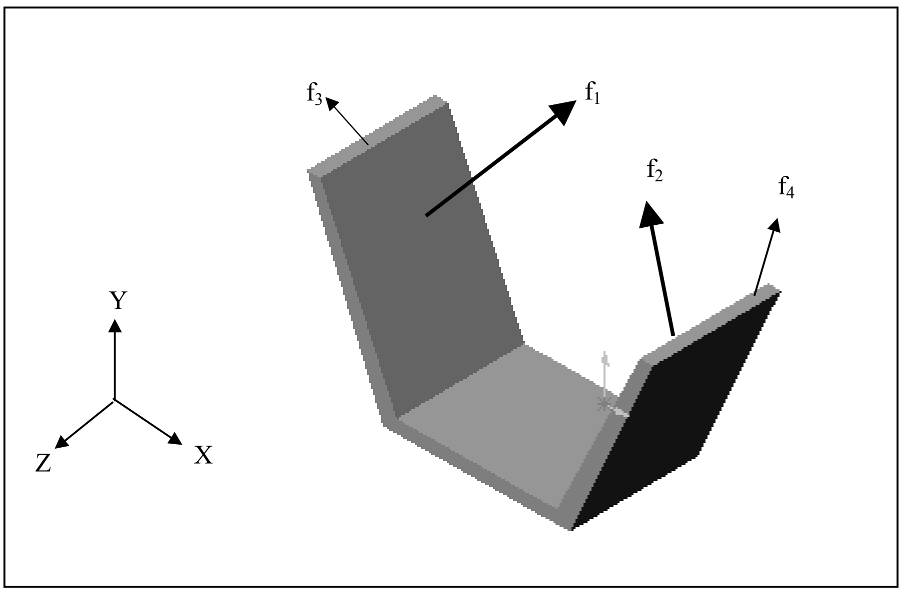


Figure 6.9 A part with planar parting surface and no undercuts

In this example, suppose parting direction along Y-axis is selected. The associated parting surface is planar, since the outer edges on faces f_3 and f_4 have equal Z. Figure 6.10 displays the FEBDAPP's output for parting surface and undercuts. It is shown that the parting surface is planar and no undercuts are detected.

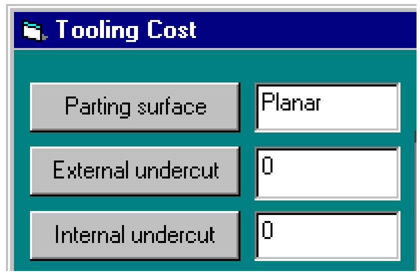


Figure 6.10 FEBDAPP's output for planar parting surfaces and no undercuts

6.4.2 Planar parting surface with undercuts

Assuming that the parting direction for the part shown in Figure 6.11 is along Z-axis, the part can be molded with planar parting surface. This is because the outer edges on the upper surface, S_u and the lower surface, S_l have constant Z-value. The boss feature, F_1 is an internal undercut. An internal undercut can be defined as a feature that prevents the molding from being ejected from the core. The hole feature, F_2 is an external undercut. An external undercut is defined as a feature that prevents the molding from being withdrawn from the cavity.

Figure 6.12 displays the FEBDAPP's output for the tooling cost estimation of the part shown in Figure 6.11. It is determined that the part requires planar parting surface. It has one external undercut and one internal undercut.

6.4.3 Non-planar parting surface with no undercuts

In some cases, non-planar parting surfaces are used to eliminate the presence of undercuts. Figure 6.13 shows a simple part that requires non-planar parting surfaces. External undercut can be avoided if a non-planar parting surface is selected. Assuming that the parting direction is in Y-axis, using planar parting surface parallel to the X-Z plane always creates an external undercut. For this part, the upper parting surface is parallel to X-Z plane at Y=Y-ordinate of edge E_1. The lower surface is located at Y=Y-ordinate of E_2. Figure 6.14 shows the results for parting surface types and undercuts. It is determined that the parting surface is non-planar. No external or internal undercut is detected.

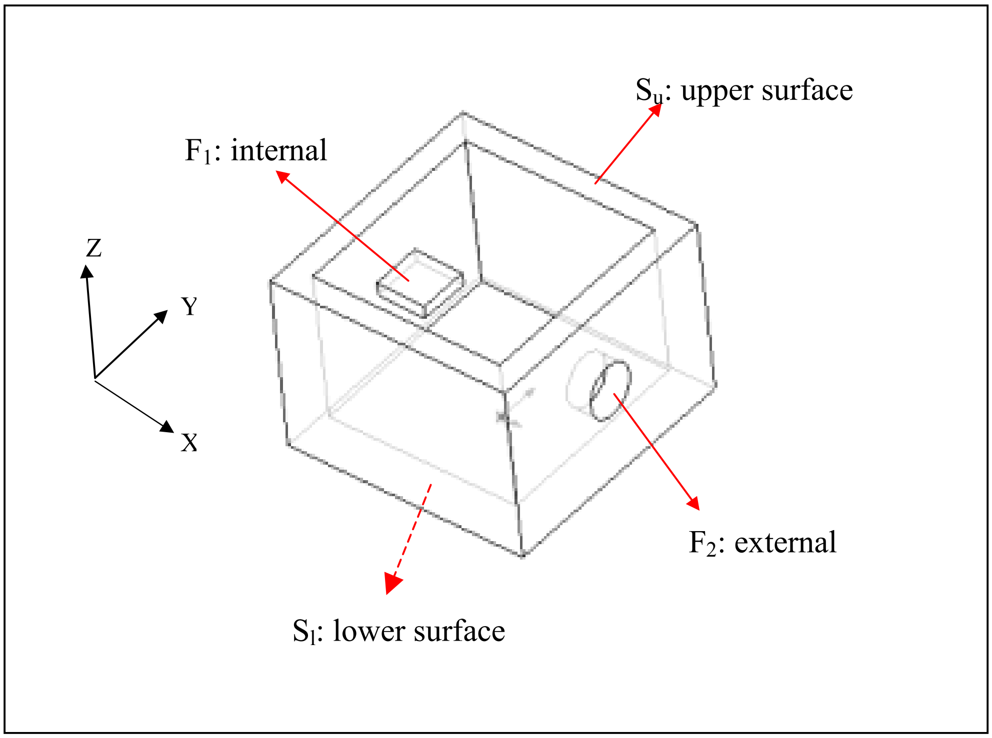


Figure 6.11 A part with planar parting surface with undercuts

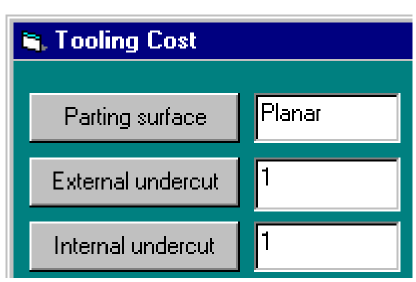


Figure 6.12 Output for planar parting surface with undercuts

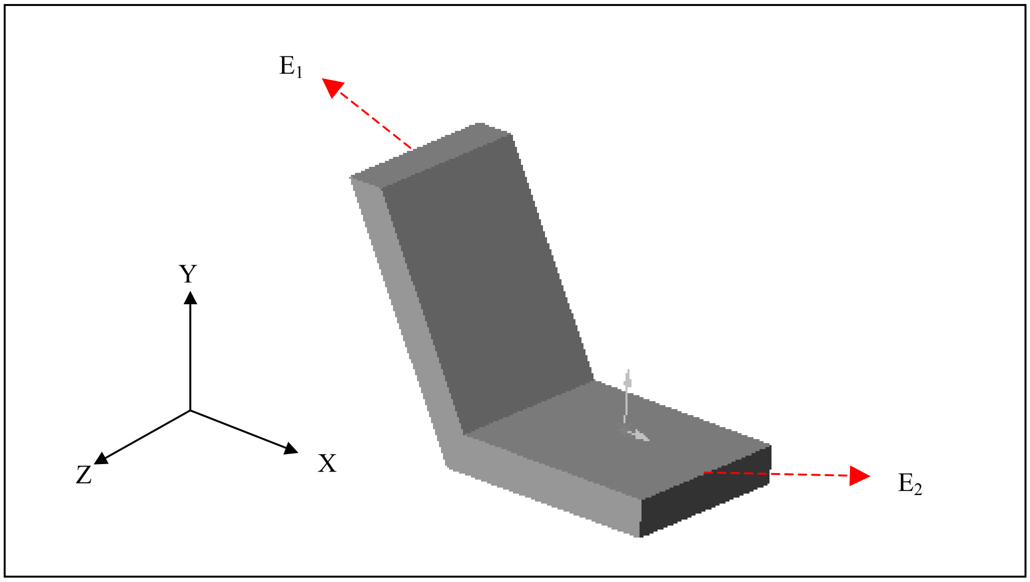


Figure 6.13 A part with non-planar surface without undercuts

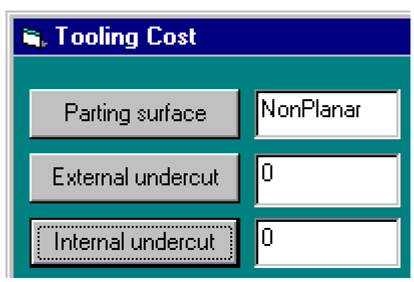


Figure 6.14 Output non-planar parting surface without undercuts

6.4.4 Non-planar parting surface with undercuts

In some situations, non-planar parting surface cannot avoid the presence of undercuts. Figure 6.15 shows a part that requires non-planar parting surface with undercuts. When a planar parting surface is used, two external undercuts are detected. Suppose, the planar parting surface is located at the lower surface, then both F_1 and surface S_1 are external undercuts. If the parting direction is selected along Y-axis, then the procedure finds that a non-planar parting surface is needed and that the boss feature F_1 is an external undercut. Figure 6.16 shows the results for this part.

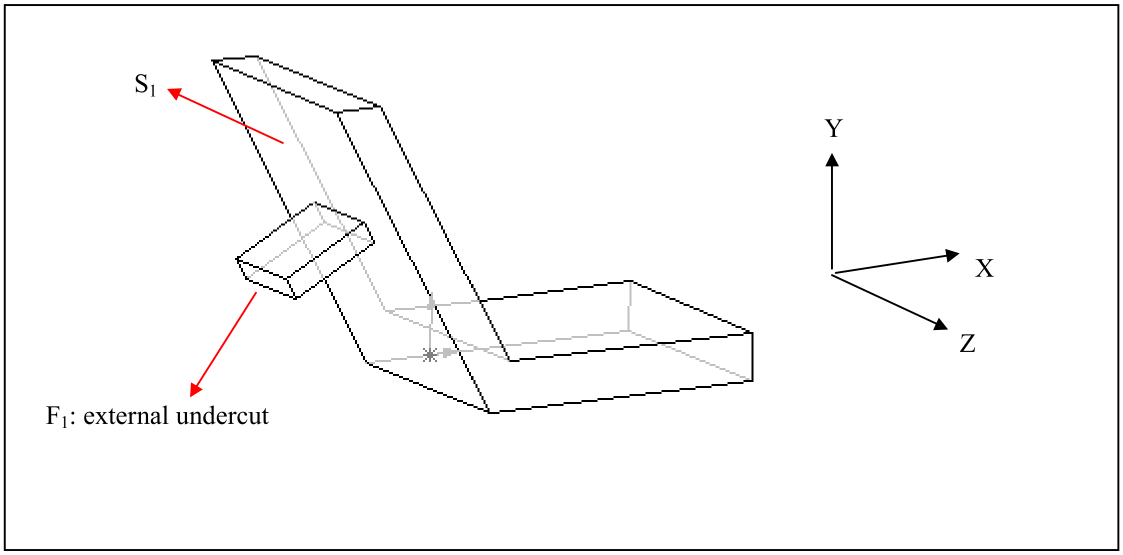


Figure 6.15 A part with non-planar parting plane and with undercuts

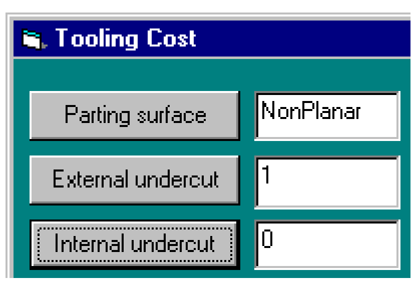


Figure 6.16 Output for non-planar parting plane with undercuts

6.5 Candidate Processes

Once all application-based features are extracted from a CAD model, MaMPS is executed to select the candidate processes. The solid axial part shown in Figure 6.4 is used to demonstrate the capability of FEBDAPP system. The parting direction along either Z-axis or Y-axis results in no undercuts, while the parting directions along X-axis will produce undercuts. Suppose parting directions along Z-axis is chosen.

Figure 6.17 shows all the candidate processes that can be used to produce the part. Two manufacturing processes can be used: machining or injection molding. It should be noted here that the process domains here are determined based on the geometrical, technological and production inputs. MaMPS uses a compatibility index to rank the candidate processes. The highest compatibility index is one and the lowest is zero. A candidate process with the compatibility index of one means that the process can fully satisfy all the design requirements. For producing the solid axial part shown in Figure 6.4, machining process is preferred over injection molding since the compatibility index of machining process is higher than that of injection molding.

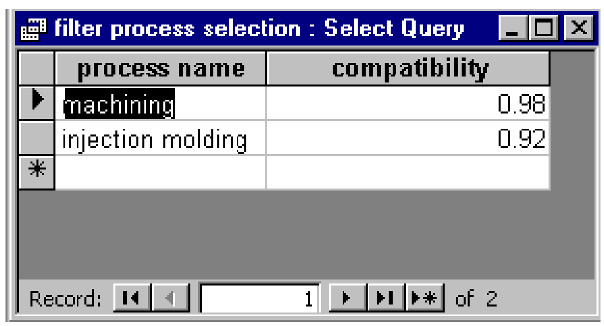


Figure 6.17 Candidate process for the solid axial part in Figure 6.4

The next part used to verify the FEBDAPP system is a screw as shown in Figure 6.18. Y-axis is selected as the parting direction since it gives zero undercuts, while X and Z axes always produce undercuts due to the presence of a through hole. Since the part is composed of a hexagonal base feature and a cylindrical add-on feature, then according to the production rules presented in Chapter 5, the part is identified as a prismatic part. The wall thicknesses of this part are the hollow cylinder's wall thickness and the hexagon's thickness.

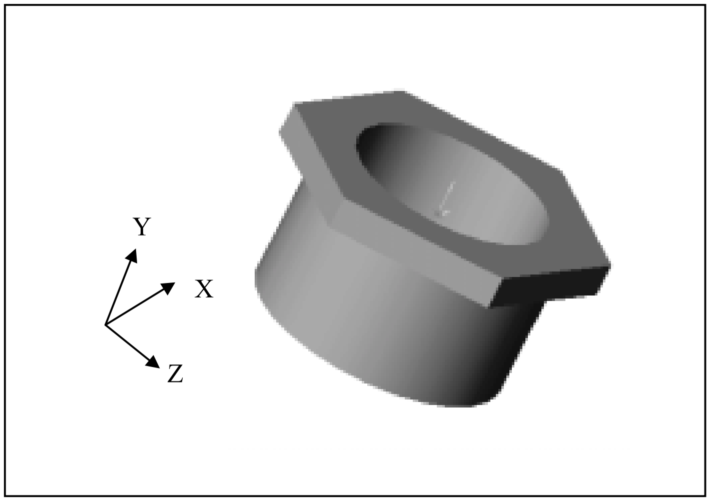
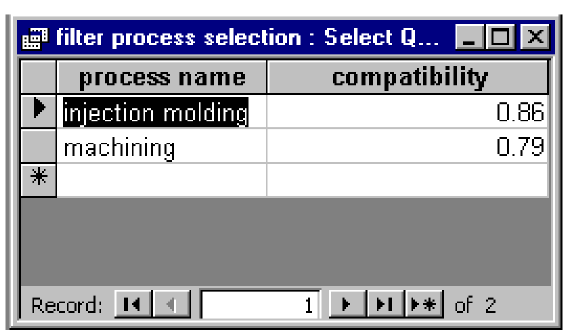


Figure 6.18 A screw

The candidate processes for the screw are listed in Figure 6.19. The injection molding and machining processes are selected as candidate processes. Injection molding is preferred over machining since its compatibility index is higher than that of machining. It should be noted here that the MaMPS'output shown in Figure 6.19 considers *geometrical features* such as wall thickness, undercuts, size, shape, weight; *technological features* such as surface finish and tolerance; and *production features* such as production rate, production volume, and time to market to determine the candidate processes and their compatibility indices.


filter process selection : Select Q...

process name	compatibility
injection molding	0.86
machining	0.79

Record: 1 of 2

Figure 6.19 Candidate processes for the screw

6.6 Relative Tooling Cost

Since the high-level process planning system is concerned with a design at the configuration stage, it is very important to be able to estimate the tooling costs. A new design's tooling cost will be compared with the tooling cost of a reference part. Dixon and Poli [Dixon 1995] use a flat washer with the outside diameter OD=72 mm, the inside diameter ID=60 mm and thickness t = 1 mm as the reference part. This part is chosen to demonstrate the FEBDAPP. The relative tooling cost for the reference part is obviously one. Figure 6.21 shows the output of the FEBDAPP's tooling cost sub-system. The approximate tooling cost for this reference part (in the 1991-92 time frame) is about \$7,000, which includes about \$1,000 in die material cost. As expected, the C_d factor for this part is determined to be 1, which means the tooling cost is the same as the reference part, \$7,000. When other parts are investigated, the tooling cost of those parts is obtained by multiplying their corresponding C_d factors by the tooling cost of the reference part.

One of the goals in this research has been to automate the estimation of cost. The manual-based cost estimation procedure for the reference part developed by Dixon and Poli [Dixon 1995] is presented here. The various factors and coefficients used in the cost estimation are presented in Appendix A.

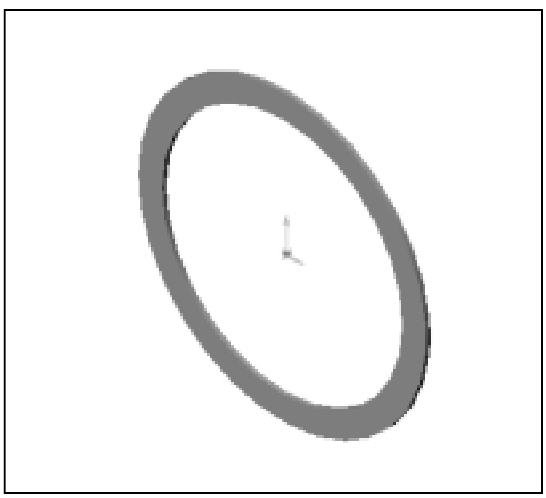


Figure 6.20 A reference part

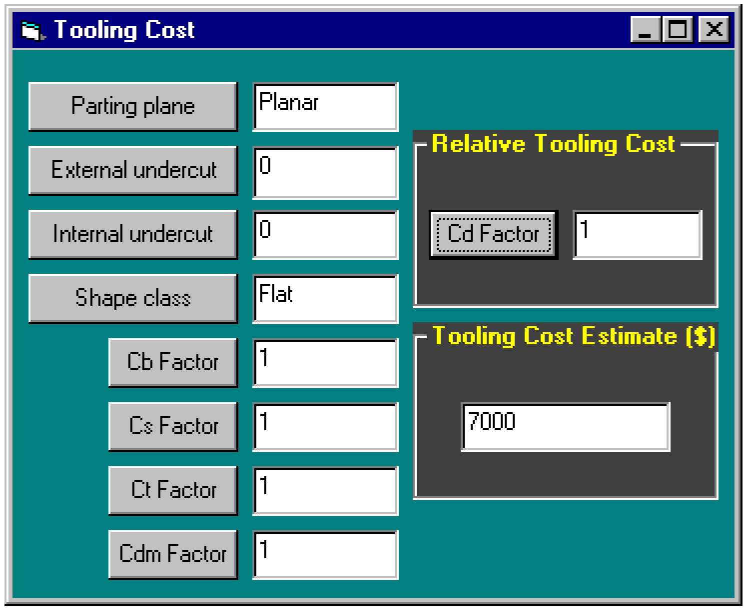


Figure 6.21Tooling cost estimate for the reference part

(a) Relative Die Construction Cost

Basic Shape: L=72; B= 72 ; H=1 → Box/Flat: Flat

Basic Complexity: 1st digit=0; 2nd digit= 0 → C_b =1.00

Subsidiary Comp.: 3rd digit=0; 4th digit=0 → C_s = 1.00

T_a/R_a: 5th digit=1; 6th digit=1 → C_t = 1.00

Total Relative Die Construction Cost, $C_{dc} = C_bC_sC_t = 1.00$

(b) Relative Die Material Cost

L_m=72; B_m=72; H_m=1 ;Die closure parallel to H

L_m/H_m=72 ; C=0.14

$M_{ws} = [0.006CH_m^4]^{1/3}$ =0.0944

$M_{wf} = 0.04L_m^{4/3} = 11.98$

$M_a = (2M_{ws} + L_m)(2M_{ws}+B_m)=(72.1888)(72.1888)=5211.22$

$M_t = (H_m+2M_{wf}) = 24.96$

Thus, $C_{dm}=1.00$

$C_d = 0.8\ C_{dc} + 0.2\ C_{dc} = 1.00$

The next part selected for verification of the FEBDAPP system is a solid axial part shown in Figure 6.4. Figure 6.22 displays the tooling cost estimate for the solid axial part. It is determined that the total relative tooling cost, C_d = 1.79, or the tooling cost estimate is $12,530. The following is the manual procedure for obtaining the tooling cost.

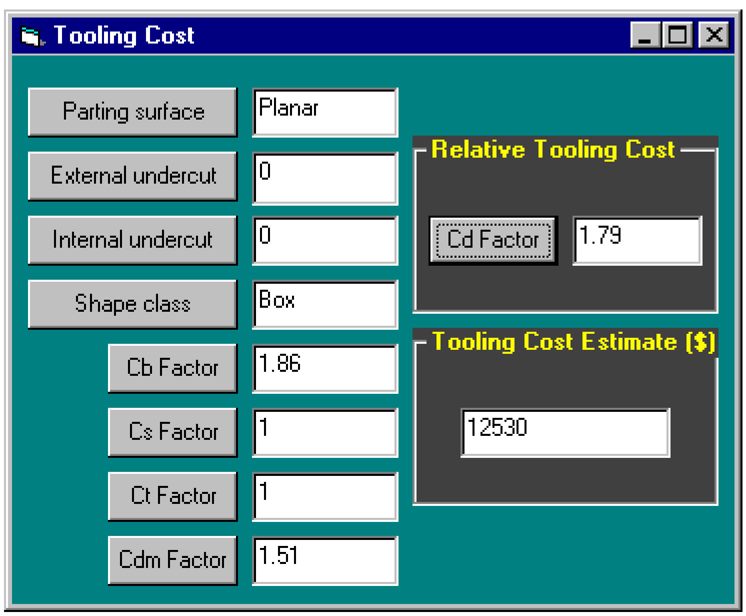


Figure 6.22 Tooling cost for the solid axial part

Relative Die Construction Cost:

Basic Shape: L=195; B= 60 ; H=60 $\rightarrow$Box/Flat: Box

Basic Complexity: 1^{st} digit=1; 2^{nd} digit= 0 $\rightarrow C_b = 1.86$

Subsidiary Comp.: 3^{rd} digit=0; 4^{th} digit=0 $\rightarrow C_s = 1$

T_a/R_a: 5^{th} digit=1; 6^{th} digit=0 $\rightarrow C_t = 1$

Total Relative Die Construction Cost, $C_{dc} = C_bC_sC_t = 1.86$

(b) Relative Die Material Cost

L_m=195; B_m=60; H_m=60 ;Die closure parallel to H

L_m/H_m=3.25 ; C=0.136

$M_{ws} = [0.006CH_m^4]^{1/3}$ =21.95

$M_{wf} = 0.04L_m^{4/3}$ =45.23

$M_a = (2M_{ws} + L_m)(2M_{ws}+B_m)$=(238.9)(103.9)=24,821.7

$M_t = (H_m+2M_{wf})$ =150.46

Thus, C_{dm}=1.51

$C_d = 0.8\ C_{dc} + 0.2\ C_{dc}$ =1.79

6.7 High-level description of the FEBDAPP system

The high-level description of the FEBDAPP system and its various stages can be summarized as follows:

1. First a part is created in SolidWorks. User can run FEBDAPP by clicking on the menu items on SolidWorks menu. Figure 6.23 shows a motor mounting plate created in SolidWorks.

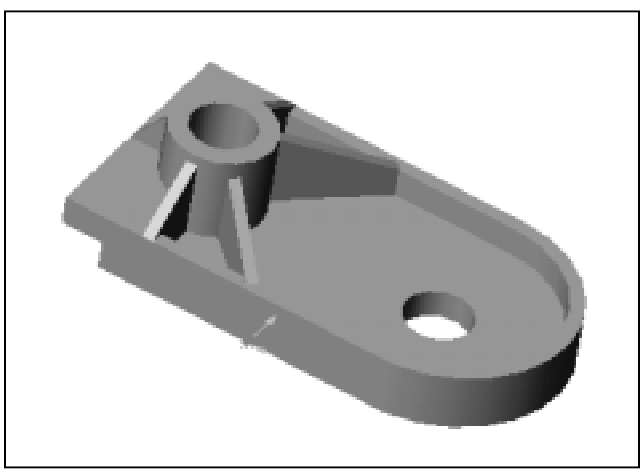


Figure 6.23 A motor mounting plate

2. The main window of FEBDAPP will pop up with the following menu items:

   a. Input to Process Selection
   b. Run Cost Estimator
   c. Candidate Process
   d. Exit FEBDAPP

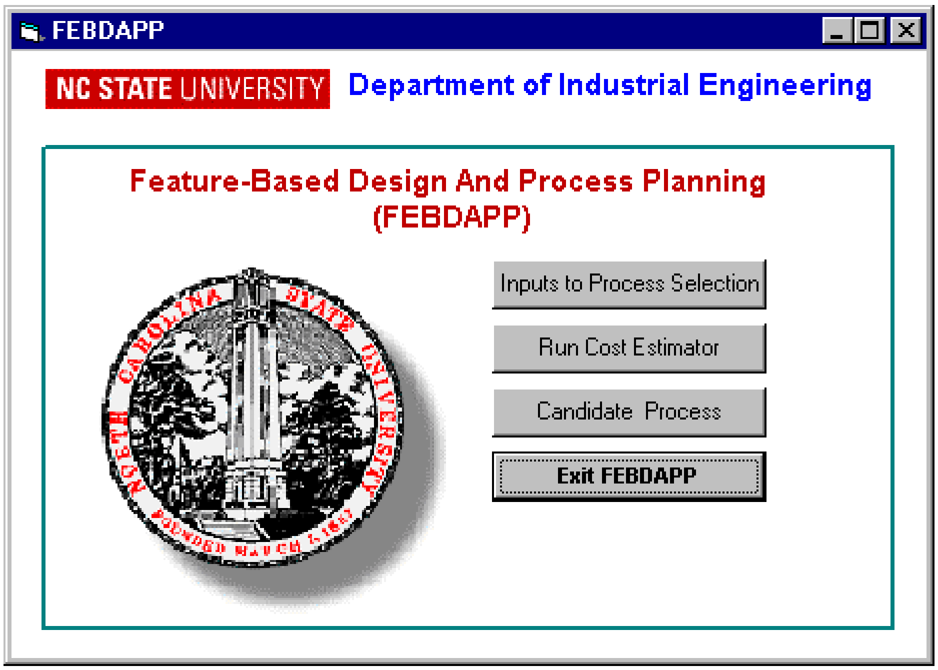


Figure 6.24 FEBDAPP's main window

3. Selecting "Input to Process Selection" on the main menu will show up the Application-based features box as shown in Figure 6.25. User can select on of the following commands: Undercuts, Wall Thickness, Box Envelope, and Shape. The values obtained constitute the geometrical inputs needed for MaMPS.

4. Once all application-based features are obtained, user can select "Transfer to MaMPS" option to transfer the information to the Microsoft Access-based MaMPS.

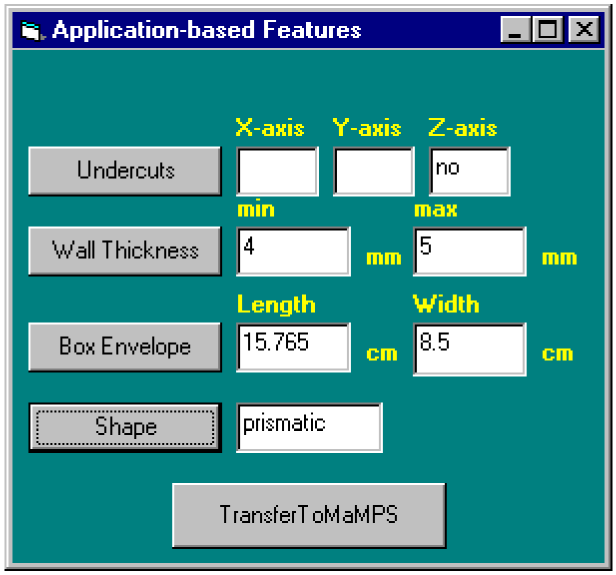


Figure 6.25 FEBDAPP's application-based feature window

5. The user is then presented with the "Process Selection Form" of MaMPS as shown in Figure 6.26.

6. The user specifies all the parameters in "mfg proc compatibility" and then selects "Select Processes" option to obtain the candidate processes.

7. The MaMPS system selects all the processes and tabulates the results in "process selection table" as shown in Figures 6.27.

8. The user selects from the candidate process those to be further analyzed. Figure 6.28 shows the result of the query of "filter process selection".

9. The user selects "Run Cost Estimator" on the FEBDAPP's main window to obtain the tooling cost estimate. Figure 6.29 shows the tooling cost estimation window.

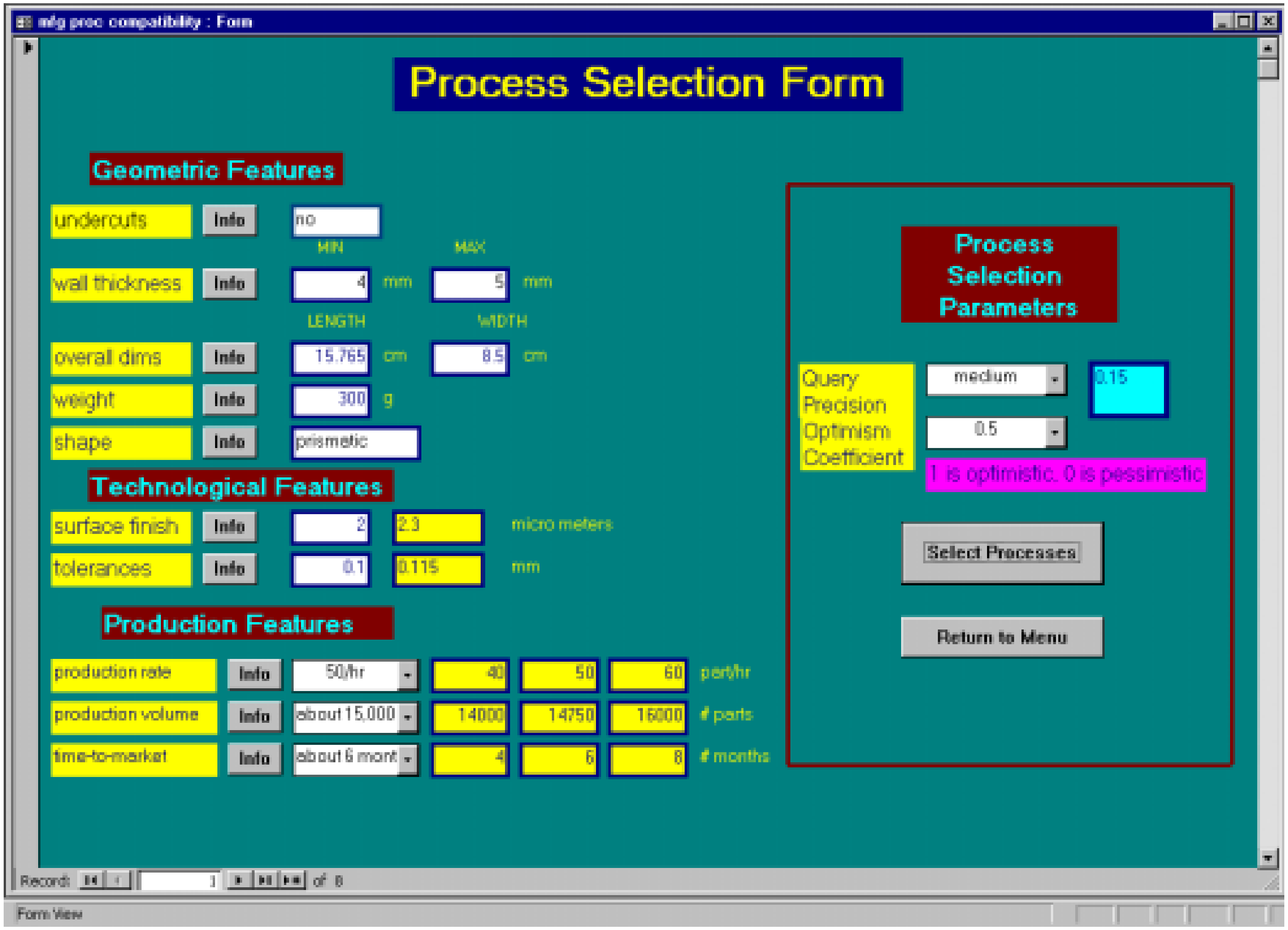


Figure 6.26 MaMPS' process selection form

process selection table : Table

process name	poss wt	poss wdth	poss lgth	rate poss	vol poss	sur poss	tol poss	compatibility
closed die forgir	0	0.11538461538	1	0	1	0	1	0
compression mı	0.2	0	0	1	0	1	1	0
die casting	1	1	1	1	1	1	0.66666666667	0.94372205744
injection moldin	1	1	1	0.66666666667	1	1	0	0
machining	1	1	1	1	0.88125	1	1	0.98210295681
powder metallur	1	0	0	0	1	1	0.33333333333	0
sand casting	0	0	0	1	1	0	0	0

Figure 6.27 MaMPS' process selection table

filter process selection : Select Query

process name	compatibility
machining	0.98
die casting	0.94

Record: 1 of 2

Figure 6.28 MaMPS' filter process selection query

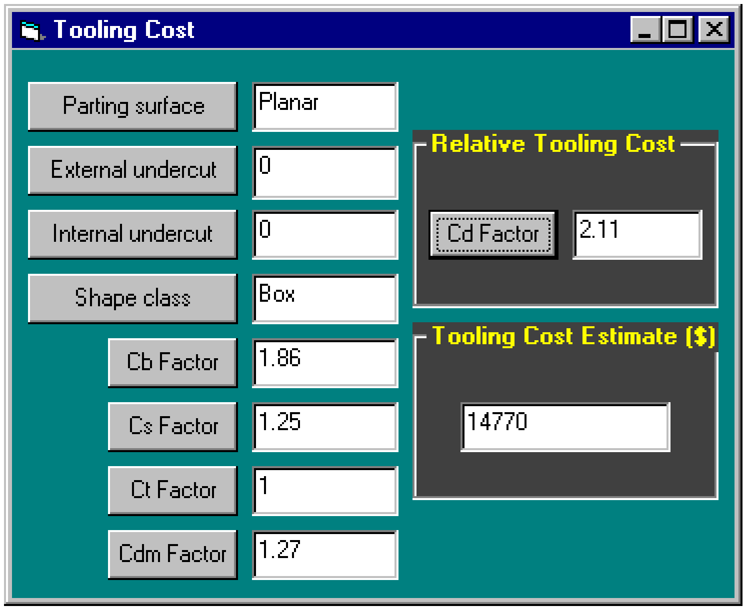


Figure 6.29 FEBDAPP's tooling cost window

Figure 6.29 shows the application-based features for tooling cost estimation of injection molding and die-casting. The relative tooling cost of 2.11 means that the actual tooling cost for the motor mounting part is 2.11 times the tooling cost of the reference part or $14,770.

This chapter presents the implementation of the FEBDAPP system. All application-based features have been successfully derived from SolidWorks and have been passed to MaMPS in order to obtain the candidate processes. Selecting the proper manufacturing processes at the beginning stages of design is very important in order to avoid the unnecessary increasing product development costs. The feature-based approach in this research can be used by designers to obtain information as to which features should be eliminated in order to reduce the tooling cost. FEBDAPP can serve as an effective concurrent engineering tool in which designer can obtain candidate processes and tooling cost estimates for a part being designed during the design process.

7. Summary and Conclusion

7.1 Summary

In this research, a CAD-based high-level process planning system, FEBDAPP, has been developed and implemented. This research is considered the first effort to implement feature technology in integrating an existing CAD system with the advanced manufacturing processes and materials selection systems. It has enabled a simultaneous engineering approach to designs, where designers can obtain on-line manufacturing advisory at the design phase.

The FEBDAPP system is a hybrid system incorporating design by feature and feature recognition approaches. By incorporating advantages from both approaches, the system provides designers with more flexibility in creating a part than the "pure" design by feature approach and requires less complex feature recognition algorithms than the common feature recognition approaches. The hybrid system consists of three main subsystems as follows: (1) part creation, (2) feature mapping, and (3) system integration.

In creating a part, designers can use predefined features to build a CAD model. If this is the case, a "pure" design by approach has been employed. But, in practice, unless the part is composed only of frequently used features, the designers prefer creating a part from a sketch. In the hybrid system adopted in this research, once a part is created, all feature attributes can be directly obtained, since all these attributes are attached during the creation of predefined features. However, if there is at least one feature that is not a predefined feature and is created from a sketch, then feature recognition approach is used to extract the feature information. In this hybrid system, a part can be represented at both macroscopic and microscopic levels. At the macroscopic level of the design by feature approach, a part is described in terms of form features that has semantic meaning, i.e. through hole, blind step etc. At the microscopic level, due to the sketch-based modeling, a part is represented in terms of vertices, edges, and faces.

After the part has been created, the next challenging task is to derive all the information needed for the downstream applications from the CAD model. Since features are application dependent, even for a part that is purely built from design by feature,

feature recognition is required. The process of obtaining part information from the CAD model is called feature mapping. The feature mapping process developed in this research is quite different from that in most other feature recognition research. Most of the existing research deal with machined part, thus feature mapping is defined as how to convert the design features in the CAD model into a set of machining features. A machining feature corresponds to the material that needs to be chipped away from the initial stock of raw material.

This research puts emphasis on parts that are created from the near net shape manufacturing processes. In this type of manufacturing processes, there is no considerable material to be removed from the stock. Feature mapping in this research becomes the tasks of converting the CAD model into a set of *application-based features* that are used for decision making in downstream applications. Feature mapping algorithm is developed based on the boundary representation of the part. Rule-based recognition approach has been developed and implemented. Application-based features such as undercuts, parting plane types, wall thickness, and part shape are the most important features covered in this research.

The CAD-based interface developed as part of this research integrates the CAD system and the existing high-level process planning system. The feature-based SolidWorks CAD system is used as the working environment for part creation. The Manufacturing Process and Material Selection System (MaMPS) [Giachetti 1998] is adopted in this research for process selection. This system is considered to be the most advanced high-level process planning system incorporating the fuzzy-based decision making process and the relational database management system.

The output of MaMPS is a set of candidate processes. An important part of this dissertation has been to develop procedures to generate the additional information on the selected processes. The cost estimation module is developed and implemented for injection molding and die-casting. Dixon and Poli’s procedures[Dixon 1995], which are performed manually, are adopted. This manual-based approach is automated by developing recognition and mapping algorithms to automatically provide all the information needed for the downstream applications.

A Visual Basic-based user interface has been developed and implemented for this research. System integration becomes easier to implement because of the built-in Application Programming Interface (API) functions in SolidWorks that can be called from either Visual Basic or Visual C++. These API functions can be used to directly obtain some basic feature attributes like vertices, lines, faces, and number of features. However, some implicit features such as undercuts detection and parting surface determination cannot be derived directly from the existing API functions. New methods are required for that purpose and were developed as part of this research. It is important to note here that API functions are CAD system dependent. Different CAD developers have their specific API functions. However, since the feature recognition and feature mapping algorithms developed here are based on boundary representation, the methods are generic and can be easily extended to the other CAD system.

7.2 Research Contributions

Some of the major contributions from this research can be summarized as follows:

1. This research lays a solid foundation for an integrated design synthesis and evaluation process. This integration relies on feature mapping that converts the part primary representation into application-based features that are used as inputs for downstream applications.
2. This research provides an effective concurrent engineering tool that helps designers make important decisions on whether or not the designs is worth generating. Design process and applications such as cost estimation, manufacturability and process planning can be done concurrently and in a real time manner.
3. This research developed feature mapping procedures that are generic and can be used in many applications.
4. This research developed and implemented an efficient CAD-based interface software capable of:
    a. Integrating CAD systems and applications.
    b. Reducing decision making process time and computation time for candidate process selection and tooling cost estimation respectively.

5. This research developed a hybrid system combining the design by feature and feature recognition approaches that offers more flexibility to designers in creating a part and requires less complex algorithm to extract feature information than any pure design by feature approach or feature recognition.

7.3 Recommendations for Further Research

This research takes major steps toward the complete integration of CAD systems and manufacturing applications by developing a hybrid system consisting of design by feature and feature recognition approaches. However, this new approach can be extended to other applications. The following is a list of some further research that can be based on the tools and algorithm developed as part of this research:

1. Enhancing the capability of the system in handling other process domains. Other near net shape manufacturing processes such as powder metallurgy, sand casting, and forging are examples.
2. Developing additional applications and moving closer toward a complete feature-based manufacturing advisory system for conceptual design. The feature mapping approach used in this research can be extended for other applications, such as manufacturability analysis.
3. Developing a neutral feature-based system that is CAD system independent. The Application Programming Interface (API) functions used in this research are SolidWorks functions. When other CAD systems are used as the working design environment, the interface commands may need to be changed, since different CAD developers have their own API functions. In order to avoid this problem, instead of using API, a "neutral" conversion system is required so that the feature-based systems become independent of a particular CAD system.
4. Developing a generic part classification and identification system that covers a wide range of part configurations. The incremental shape definition technique developed in this research works well for defining a part shape, when a part is created in a feature-based working environment. However, the shape classification used in this research is

adopted from the existing high-level process planning system. A more generic part classification should be developed in order to cover a wide range of part shapes.

REFERENCES

[Alting 1989] Alting, L. and Zhang, H. (1989), “Computer Aided Process Planning: The State of the Art Survey”, *International Journal of Production Research*, Vol. 27, No. 4, pp. 535-585.

[Boothroyd 1994] Boothroyd, G., Dewhurst, P., and Knight, W. (1994), *Product Design for Manufacture and Assembly*, Marcel Dekker, Inc., New York.

[Boothroyd 1992] Boothroyd, G., Dewhurst, P., and Knight, W., “Selection of Materials and Processes for Component Parts”, *Proceedings of the 1992 NSF Design and Mfg. Systems Conference*, 1992, pp. 255-263.

[Bralla 1986] Bralla, J.G. (1986), *Handbook of Product Design for Manufacturing: A Practical Guide to Low-Cost Production*, McGraw-Hill, New York, 1986.

[Bralla 1999] Bralla, J.G., *Design for Manufacturability Handbook*, McGraw-Hill, New York, 1999.

[Burchard 1987] Burchard, R.L., “Feature-Based Geometric Constraints Applied to Constructive Solid Geometry”, *M.S. Thesis*, Purdue University, 1987.

[CAM-I 1981] CAM-I, “CAM-I’s Illustrated Glossary of Workpiece Form Feature”, Report No. R-80-ppp-02, CAM-1 Inc., Arlington, Texas, 1981.

[Chan 1998] Chan, K., King, C., and Wright, P., “COMPASS: Computer Oriented Materials, Processes and Apparatus Selection Systems”, *Journal of Manufacturing Systems*, Vol. 17, No. 4, 1988, pp. 275-286.

[Chan 1994] Chan, C.H. (1994),”Artificial Neural-Network-Based Feature Recognition and Grammar-Based Feature Extraction to Integrate Design and Manufacturing”, *PhD Thesis*, The University of Iowa, 1994.

[Chen 1997] Chen, Y. -M. and Wei, C. -L., “Computer-Aided Feature-Based Design for Net Shape Manufacturing”, *Computer Integrated Manufacturing Systems*, Vol. 10, No. 2, 1997,pp. 147-164.

[Crawford 1993] Crawford, R., “Integrating 3D Modelling and Process Planning by Features: a Case Study”, *International Journal of Computer Integrated Manufacturing”*, Vol. 6, Nos. 1&2, 1993, pp. 113-118.

[Cunningham 1988] Cunningham, J. and Dixon, J.R., “Designing with Features: The Origin of Features”, *ASME Computers in Engineering Conference*, San Fransisco, July 31-August 4, ASME Press, 1988, pp. 237-243.

[Da Silva 1991] Da Silva, R.E., Wood, K.L., and Beaman, J.J., “Interacting and interfeature relationships in engineering design for manufacture”, *International Journal of Systems Automation, Research and Applications*, Vol. 1, 1991, pp. 263-286.

[Dargie 1982] Dargie, P.P., Parmeshwar, K., and Wilson, W.R.D. (1982), “MAPS-1: Computer-Aided Design System for Preliminary Material and Manufacturing Process Selection”, *ASME Transaction*, Vol. 104, pp. 126-136.

[Dewhurst 1988] Dewhurst, P. and Boothroyd, G.,”Early Cost Estimating in Product Design”, *Journal of Manufacturing Systems*, Vol. 7, No. 3, 1988, pp. 183-191.

[Dixon 1995] Dixon, J.R. and Poli, C., *Engineering Design and Design for Manufacturing a Structured Approach: Text & Reference for Mechanical Engineers*, Field Stone Publishers, Conway, MA, 1995.

[Dong 1991] Dong, X. and Wozny, M.J., “Instantiation of User Defined Features on A Geometric Model”, *Product Modeling for Computer-Aided Design and Manufacturing*, edited by Turner, J., Pegna, J. and Wozny, M., Elsevier Science Publishers B.V., 1991.

[Esawi 1998] Esawi, A.M.K. and Ashby, M.F., “Computer-Based Selection of Manufacturing Processes: Methods, Software and Case Studies”, *Journal of Engineering Manufacture*, Vol. 212, Np. B8, 1998, pp. 595-610.

[Farris 1992] Farris, J., “Selection of Processing Sequences and Materials During Early Product Design”, *Ph.D Thesis*, University of Rhode Island, 1992.

[Feng 1996] Feng, C.X., Kusiak, A., and Huang, C.C., “Cost Evaluation in Design With Form Features”, *Computer-Aided Design,* Vol. 28, No. 11., 1996, pp. 879-885.

[Gadh 1989] Gadh, R. and Prinz, F.B., "Recognition of Geometric Forms using the Differential Depth Filter", *Computer-Aided Design*, Vol. 24, No.11, November 1989, pp. 583-598.

[Giachetti 1998] Giachetti, R.E., "A Decision Support System for Material and Manufacturing Process Selection", *Journal of Manufacturing Systems*, Vol. 9, 1998, pp. 265-276.

[Gindy 1989] Gindy, N.N.Z., "A Hierarchical Structure for Form Features", *International Journal of Production Research*, Vol. 27, No. 12, 1989, pp. 2089-2103.

[Goel 1996] Goel, V. and Chen, J., "Application of Expert Network for Material Selection in Engineering Design", *Computers in Industry*, Vol. 30, 1996, pp. 87-101.

[Gorti 1998] Gorti, S.R., Gupta, A., Kim, G.J., Sriram, R.D. and Wong, A., "An Object-Oriented Representation for Product and Design Processes", *Computer-Aided Design*, Vol. 30, No. 7, 1998, pp. 489-501.

[Han 1997] Han, J.H. and Requicha, A.A.G., "Integration of Feature-Based Design and Feature Recognition, *Computer-Aided Design"*, Vol. 29, No. 5, 1997, pp. 393-403.

[Hoffmann 1998] Hoffmann, C.M. and Arinyo, R.J-,"On User-Defined Features", *Computer-Aided Design*, Vol. 30, No. 5, 1998, pp. 321-332.

[Hummel 1989] Hummel, K.E., "Coupling Rule-based and Object-Oriented Programming for the Classification of Machined Features", *Computers in Engineering*, Vol. 1, 1989, pp 409-418.

[Joshi 1988] Joshi, S. and Chang, T.C.,"Graph-based Heuristics for Recognition of Machined Features from a 3D Solid Model," *Computer-Aided Design*, Vol.20,No.2, March, 1989, pp 58-66.

[Kang 93] Kang, T. and Nnaji, B.O. (1993). "Feature Representation and Classification for Automatic Process Planning Systems", *Journal of Manufacturing Systems*, Vol. 12, No.2, pp. 133-145.

[Kim 1994] Kim, C., "A Representation Formalism for Feature-Based Design as a Formal Concurrent Engineering Approach", *Ph.D Thesis*, North Carolina State University, Raleigh, NC, 1994.

[Laako 1993] Laakko, T. and Mantyla, M., “Feature Modeling by Incremental Feature Recognition”, *Computer-Aided Design*, Vol. 25, No. 8, 1993, pp. 479-492.

[Latif 1993] Latif, M.N., Boyd, R.D., and Hannam, R.G., “Integrating CAD and Manufacturing Intelligence Through Features and Objects”, *International Journal of Computer Integrated Manufacturing*, Vol. 6, No. 1 & 2, 1993, pp. 87-93.

[L.-Wright 1989] Lawlor-Wright, T. and Hannam, R.G., “A Feature-Based Design for Manufacture: CADCAM Package”, *Computer-Aided Engineering Journal*, December 1989, Vol. 6, No. 6, 1989, pp.215-220.

[Lee 1992] Lee, C.-H., “A Knowledge-based Systems Approach for Manufacturing Process Selection in Design”, *Ph.D Thesis*, The Ohio State University, 1992.

[Lee 1998] Lee, J.Y. and Kim, K., “A Feature-based Approach to Extracting Machining Features”, *Computer-Aided Design”*, Vol. 30, No. 13, 1998, pp. 1019-1035.

[Lenau 1993] Lenau, T. and Mu, L., “Features in Integrated Modeling of Products and Their Production”, *International Journal of Computer Integrated Manufacturing*, Vol. 6, No. 1 & 2, 1993, pp. 65-73.

[Liao 1996] Liao, T.W., “A Fuzzy Multicriteria Decision Making in Material Selection”, *Journal of Manufacturing Systems*, Vol. 15, No. 1, 1996, pp. 1-12.

[Lovatt 1998] Lovatt, A., “Process Selection in Engineering Design”, *Ph.D Thesis*, Cambridge University, UK, 1998.

[Luby 1986] Luby, S.C., Dixon, J.R. and Simmons, M.K., “Creating and Using A Features Data Base”, *Computers in Mechanical Engineering*, Vol. 5, No. 3, 1986, pp. 25-33.

[Nachman 2000] Nachman, C., “Attributes: API Programming Power Unleashed”, *Solid Solutions*, Vol. 4, No. 1, 2000, pp.12-16.

[Oh 1995] Oh, J.S., “A Feature-Based Design Formalism For Concurrent Design For Assembly”, *Ph.D Thesis*. North Carolina State University, Raleigh, NC, 1995.

[Patel 1988] Patel, R.M. and McLeod, A.J., “The Implementation of a Mechanical Engineering Design Interface Using Engineering Features”, *Computer-Aided Engineering Journal*, Vol. 5, No. 6, 1988, pp. 241-146.

[Parkan 1998] Parkan, C. and Wu, M., “Process Selection with Multiple Objective and Subjective Attributes”, *Production Planning & Control*, Vol. 9, No. 2, 1998, pp. 189-200.

[Parks 1989] Parks, R.D. and Chase, T.R., “Representing Mechanical Parts Using Feature Specifications and Positional Constraints: A Constraint with PDES”, *Engineering Database Management: Leadership Key for the 90’s*, edited by Rasdorf, W.J. and Filton, R.E., ASME, 1989.

[Pinilla 1989] Pinilla, J.M., Finger, S., and Prinz, F.B., “Shape Feature Description and Recognition Using an Augmented Topology Graph Grammar,” *NSF Engineering Design Research Conference*, University of Massachusetts, June 11-14, 1989, pp.285-300.

[Pratt 1993] Pratt, M.J.,“Applications of Feature Recognition in the Product Life-Cycle”, *International Journal of Computer Integrated Manufacturing”*, Vol.6, Nos. 1&2, 1993, pp. 13-19.

[Pugh 1991] Pugh, S., *Total Design: Integrated Methods for Successful Product Engineering*, Addison Wesley, 1991.

[Ravi 1990] Ravi, B. and Srinivasan, M.N., “Computer-Aided Design”, *Computer-Aided Design*, Vol. 22, No. 1, 1990, pp. 11-19.

[Raviwongse 2000] Raviwongse, R., Allada, V., and Sandidge, T. “Plastic Manufacturing Process Selection Methodology Using Self-Organizing Map (SOM)/Fuzzy Analysis”, *International Journal of Advanced Manufacturing Technology*, Vol. 16, 2000, pp. 155-161.

[Regli 1995] Regli, W.C., “Geometric Algorithms for Recognition of Features from Solid Models”, *Ph.D Thesis*, The University of Maryland, College Park, 1995.

[Requicha 1989] Requicha, A.A.G. and Vandenbrande, J.H., “Form Features for Mechanical Design and Manufacturing,” *Computers in Engineering*, Vol.1, 1989, pp 47-52.

[Roller 1989] Roller, D., “Design by Features: An Approach to High Level Shape Manipulations”, *Computers in Industry*, Vol. 12, 1989, pp. 185-191.

[Rossen 1992] Rosen, D.W.,”A Feature-Based Representation to Support the Design Process and the Manufacturability Evaluation of Mechanical Components”, *PhD Thesis*, University of Massachusetts, 1992.

[Rumbaugh 1991] Rumbaugh, J. et al. *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[Sakurai 1988] Sakurai, H. and Gossard, D.C., “Shape Feature Recognition from 3D Solid Models”, *Computers in Engineering*, Vol.1, 1988, pp.515-519.

[Salomons 1993] Salomons, O.W., Van Houten, F.J.A.M., and Kals, H.J.J., “Review of Research in Feature-Based Design”, *Journal of Manufacturing Systems*, Vol. 12, No.2, pp. 113-132.

[Shah 1991] Shah, J.J., Rogers, M.T., Sreevalsan, P.C., Hsiao, D.W., Mathew, A., Bhatnagar, A., Liou, B.B., and Miller, D.W., “The A.S.U. Features Testbed: An Overview,” *Computers in Engineering*, Vol. 1, 1989, pp 233-241.

[Shah 1991] Shah, J.J.,”Assessment of Features Technology”, *Computer-Aided Design*, Vol. 23, No. 5, 1991, pp. 331-341.

[Shah 1992] Shah, J.J., “Features in Design and Manufacturing”, *Intelligent Design and Manufacturing*, edited by Kusiak, A., John Wiley & Sons, Inc., 1992, pp.39-72.

[Shah 1996] Shah, J.J., Jeon, D.K., Urban, S.D., Bliznakov, P. and Rogers, M., “Database Infrastructure for Supporting Engineering Design Histories”, *Computer-Aided Design*, Vol. 28, No. 5, pp. 347-360.

[Shah 1995] Shah, J.J. and Mäntylä, M., *Parametric and Feature-Based CAD/CAM Concept, Techniques, and Applications*, John Wiley & Sons, Inc., New York, 1995.

[Shah 1988] Shah, J.J. and Mary, T.R., “Functional Requirement and Conceptual Design of the Feature-Based Modelling System”, *Computer-Aided Engineering Journal*, February, 1988, pp. 9-15.

[Shea 1989] Shea, C. and Dewhurst, P., “Computer-Aided Materials and Process Selection”, *Proceedings 4th International Conference on Product Design for Manufacture and Assembly*, Newport, R.I., 1989.

[Smith 1999] Smith, C.S., "The Manufacturing Advisory Service:Web Based Process and Material Selection, *Ph.D Thesis*, University of California, Berkeley, 1999.

[Smith 1996] Smith, C.S., and Wright, P.K.,"CyberCut: A World Wide Web Based Design to Fabrication Tool", *Journal of Manufacturing Systems*, Vol. 15, No. 6, 1996, pp. 432-442.

[Swift 1997] Swift, K.G. and Booker, J.D., *Process Selection from Design to Manufacture*, Arnold, London, 1997.

[Tsang 1989] Tsang, J.P. and Brissaud, D., "A Feature-Based Approach to Process Planning", *Computers in Engineering*, Vol.1, 1989, pp 419-430.

[Ullman 1992] Ullman, D.G., *The Mechanical Design Process*, McGraw-Hill, New York, 1992.

[Ulrich 1995] Ulrich, K.T. and Eppinger, S.D., *Product Design and Development*, McGraw-Hill, Inc., New York, 1995.

[Wade 1989] Wade, J. and Colton, J.S., "A Framework for Feature-Based Representation of the Design Process", *Engineering Database Management: Leadership Key for the 90's*, edited by Rasdorf, W.J. and Fulton, R.E., ASME, 1989.

[Wierda 1990] Wierda, L.S., "Design-Oriented Cost Information: The Need and the Possibilities", *Journal of Engineering Design*, Vol. 1, No. 2, pp. 147-167, 1990.

[Wierda 1991] Wierda, L.S., "Linking Design, Process Planning and Cost Information by Feature-Based Modeling", *Journal of Engineering Design*, Vol. 2, No. 1, 1991, pp. 3-19.

[Wilson 1988] Wilson, P.R. and Pratt, M.J., "A Taxonomy of Features For Solid Modeling", *Geometric Modeling for CAD Applications*, edited by M.J. Wozny et al., Elsevier Science Publishers B.V., North-Holland, 1988, pp. 125-137.

[Woo 1982] Woo, T.C. (1982), "Feature Extraction by Volume Decomposition", Proceedings of Conference of CAD/CAM in Mechanical Engineering, Cambridge, MA, pp.39-45

[Xiang 2000] Xiang, W., Chuen, C.W., Wong, C.M., and Yam, L.H., "Feature Representation and Database Schema of an Object-Oriented Feature-Based Product Model for Hydraulic Manifold Blocks", *International Journal of Advanced Manufacturing Technology*, Vol. 16, 2000, pp. 182-188.

[Yu 1993] Yu, J. -C., Krizan, S. and Ishii, K., “Computer-Aided Design for Manufacturing Process Selection”, *Journal of Intelligent Manufacturing*, Vol. 4, 1993, pp. 199-208.

[Zamanian 1991] Zamanian, M.K., Fenves, S.J., Thewalt, C.R. and Finger, S.,”A Feature-based Approach to Structural Design,” *Engineering with Computers*, 7, 1991, pp 1-9.

APPENDIX A

RELATIVE TOOLING COSTS ESTIMATE FOR INJECTION MOLDING AND DIE CASTING [Dixon 1995]

The cost of an injection molded part consists of three sub-costs: (1) tooling (or mold) cost, K_d/N, (2) processing cost (or equipment operating cost), K_e, and (3) part material cost, K_m.

Total cost per part = $K_d/N + K_e + K_m$ (1)

where K_d is the total cooling cost for a part and N is the number of parts to be produced with the mold

At the configuration design stage, we can do little to estimate processing costs (K_e) or part material costs (K_m). However, we can make a reasonably accurate estimate of tooling costs (K_d).

Relative tooling cost, C_d. The tooling cost of injection molded or die-cast parts can determines as a ratio of expected tooling costs to the tooling costs for a reference part.
The relative tooling cost:

C_d = Cost of Tooling for Designed Part / Cost of Tooling for Reference Part

That is:

$$C_d = (K_{dm} + K_{dc}) / (K_{dmo} + K_{dco}) \quad (2)$$

where K_{dmo} and K_{dco} refer to die material cost and die construction cost for the reference part. The reference part is a flat washer with OD=72 mm, ID=60 mm and t=1mm. The approximate tooling cost this reference part is about \$7,000 (1991-1992 time frame) including about \$1,000 in die material costs.

Equation (2) can be written as

$$C_d = K_{dm}/(K_{dmo}+K_{dco}) + K_{dc}/(K_{dmo}+K_{dco})$$

$$= A(K_{dm}/K_{dmo}) + B(K_{dc}+K_{dco}) \quad (3)$$

where

A $= K_{dmo}/(K_{dmo}+K_{dco})$

B $= K_{dco}/(K_{dmo}+K_{dco})$

Based on data collected from mold makers, a reasonable value for A is between 0.15-0.20 and a reasonable value for B is between 0.80-0.85. For this research, we will take A and B to be 0.2 and 0.8, respectively. Therefore, Equation (3) becomes:

$$\mathbf{C_d = 0.8\, C_{dc} + 0.2\, C_{dm}} \quad (4)$$

where C_d : the total die cost of a part relative to the die cost of the reference part

C_{dc} : the die construction cost for the part relative to the die construction cost of the reference part

C_{dm} : the die material cost for the part relative to the die material cost of the reference part

A-1. Relative tooling construction cost, C_{dc}

The relative tooling construction cost, C_{dc} is expressed as follows:

$$C_{dc} = C_b\, C_s\, C_t$$

where

C_b : the approximate relative tooling cost due to size and basic complexity

= *f(internal undercut, external undercut, parting plane, part size)*

$\boldsymbol{C_s}$: a multiplier accounting for other complexity factors called subsidiary factors
= *f(features aligned with the mold closure direction, cavity detail, complexity & number of external undercuts)*

$\boldsymbol{C_t}$: a multiplier accounting for tolerance and surface finish
= *f(surface finish, tolerance)*

$\boldsymbol{C_b}$ is determined based on the number of internal and external undercuts, the type of parting planes, and the part size. The feature mapping algorithms for these application-based features have been presented in this chapter. $\boldsymbol{C_s}$ is determined based on the occurrence of typical features such as holes, bosses, ribs, extensive external undercuts, and side shutoffs that are aligned to the mold closure direction. $\boldsymbol{C_t}$ involves non-CAD based features, such as surface finish and is general tolerance. Therefore, feature recognition algorithms need to be developed to recognize features to determine the $\boldsymbol{C_s}$ factor.

All features associated with $\boldsymbol{C_s}$ can be created and stored as predefined features. So, when the design by feature approach is used to create the part, these features can be easily extracted.

A-2 Determining C_b Factor

Tables A-1 and A-2 show the classification system for determining C_b. This classification system requires that the part to be evaluated be classified as either *flat* or *box shaped*. In order to define either flat or box shaped, the basic envelope for the part needs to be determined first. The lengths of the sides of the basic envelope are denoted by L, B, and H where $L \geq B \geq H$. A part is considered flat if L/H is greater than 4; otherwise it is considered box shaped.

Table A-1 Classification System for Basic Tool Complexity, C_b for **Flat Parts**

BASIC COMPLEXITY (C_b)				L≤250mm				250mm<L≤480mm				L>480 mm		
				Number of External Undercuts				Number of External Undercuts				Number of External Undercuts		
				Zero	One	Two	≥2	Zero	One	Two	≥2	Zero	One	≥1
				0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Parts without internal undercuts	Parts whose peripheral height from a planar dividing surface is constant	Part in one half	**0**	1.00	1.23	1.38	1.52	1.42	1.65	1.79	1.94	1.83	2.07	2.33
		Part not in one half	**1**	1.14	1.37	1.52	1.66	1.61	1.84	1.99	2.13	2.09	2.32	2.58
	Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface		**2**	1.28	1.51	1.66	1.80	1.81	2.04	2.19	2.33	2.34	2.58	2.84
Parts with internal undercuts	On one face	Parts whose ONLY dividing surface is planar, or parts whose peripheral height from a planar dividing surface is constant	**3**	2.33	2.57	2.71	2.86	2.75	2.98	3.13	3.27	3.17	3.40	3.66
		Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface	**4**	2.98	3.21	3.36	3.50	3.52	3.75	3.89	4.04	4.04	4.28	4.54
	On more that one face	Parts whose ONLY dividing surface is planar, or parts whose peripheral height from a planar dividing surface is constant	**5**	4.20	4.43	4.58	4.72	4.62	4.85	4.99	5.14	5.03	5.27	5.53
		Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface	**6**	5.37	5.60	5.74	5.89	5.90	6.13	6.28	6.42	6.43	6.67	6.93

Second Digit

First Digit

Table A-2 Classification System for Basic Tool Complexity, C_b for **Box Shaped Parts**

BASIC COMPLEXITY (C_b)					L≤250mm				250mm<L≤480mm				L>480 mm		
					Number of External Undercuts				Number of External Undercuts				Number of External Undercuts		
					Zero	One	Two	≥2	Zero	One	Two	≥2	Zero	One	≥1
					0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Parts without internal undercuts	Parts whose peripheral height from a planar dividing surface is constant		Part in one half	**0**	1.64	1.87	2.02	2.16	2.89	3.12	3.27	3.41	4.28	4.51	4.77
			Part not in one half	**1**	1.86	2.09	2.24	2.38	2.99	3.22	3.37	3.51	4.42	4.66	4.92
	Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface			**2**	1.92	2.15	2.29	2.44	3.38	3.61	3.76	3.90	5.01	5.24	5.50
Parts with internal undercuts	On one face	Parts whose ONLY dividing surface is planar, or parts whose peripheral height from a planar dividing surface is constant		**3**	3.19	3.43	3.57	3.72	4.44	4.68	4.82	4.97	5.83	6.07	6.33
		Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface		**4**	3.73	3.97	4.11	4.26	5.20	5.43	5.58	5.72	6.82	7.06	7.32
	On more that one face	Parts whose ONLY dividing surface is planar, or parts whose peripheral height from a planar dividing surface is constant		**5**	5.37	5.61	5.75	5.89	6.62	6.86	7.00	7.14	8.01	8.24	8.51
		Parts whose peripheral height from a planar dividing surface is not constant, or parts with a non-planar dividing surface		**6**	6.28	6.52	6.66	6.81	7.74	7.98	8.12	8.27	9.37	9.60	9.86

Second Digit

First Digit

A-3 Determining C_s Factor

C_s is mainly influenced by two things: (1) cavity detail, and (2) complexity of external undercuts. Cavity detail is influenced by features like ribs, bosses, holes, etc. aligned with the parting direction. Table A-3 shows the method for rating the cavity detail as low, moderate, high, or very high. External undercuts other than unidirectional holes or depressions are considered extensive since the creation of such tooling is more costly.

Table A-3 Determination of Cavity Detail

		Number of Features (n)	Penalty per Features	Penalty
Holes or Depressions	Circular		2n	
	Rectangular		4n	
	Irregular		7n	
Bosses	Solid		n	
	Hollow		3n	
Non-peripheral ribs and/or walls and/or rib clusters			3n	
Side Shutoffs	Simple		2.5n	
	Complex		4.5n	
Lettering			n	
			Total Penalty	

For SMALL parts (L≤250mm):
Total Penalty ≤10 →Low cavity detail
10 < Total Penalty ≤20 →Moderated cavity detail
20 < Total Penalty ≤40 →High cavity detail
Total Penalty >40 →Very high cavity detail

For MEDIUM parts (250<L≤480mm):

Total Penalty ≤15 →Low cavity detail
15 < Total Penalty ≤30 →Moderated cavity detail
30 < Total Penalty ≤60 →High cavity detail
Total Penalty >60 →Very high cavity detail

For LARGE parts (L>480mm):

Total Penalty ≤20 →Low cavity detail
20 < Total Penalty ≤40 →Moderated cavity detail
40 < Total Penalty ≤80 →High cavity detail
Total Penalty >80 →Very high cavity detail

The subsdisiary complexity rating, C_s is determined by using the following table.

Table A-4 Subsidiary complexity rating, C_s

SUBSDISIARY COMPLEXITY			External Undercut Complexity	
			Without extensive external undercuts	With extensive external undercuts
			0	**1**
Cavity Detail	Low	**0**	1.00	1.25
	Moderate	**1**	1.25	1.45
	High	**2**	1.60	1.75
	Very High	**3**	2.05	2.15

Fourth Digit

Third Digit

A-4 Determining C_t Factor

C_t is influenced by surface finish and tolerance and determined from Table A-5.

Table A-5 Tolerance and Surface Finish Rating, C_s

			TOLERANCES	
			Commercial	Tight
		Sixth Digit →	**0**	**1**
SURFACE FINISH	SPI 5-6	**0**	-	-
	SPI 3-4	**1**	1.00	1.05
	TEXTURE	**2**	1.05	1.10
	SPI 1-2	**3**	1.10	1.15
		↑ **Fifth Digit**		

A-5 Die material cost

The die material cost, C_{dm} is determined based on the following factors:

M_{ws} : the thickness of the mold side walls

M_{wf} : the thickness of the core plate

L_m ***and*** B_m : the length and width of the part in a direction normal to the mold opening direction

H_m : the height of the part in the direction of mold closure

M_t : the required thickness of the mold base

where

$M_{ws} = [0.006C\ H_m^4]^{1/3}$

where $C = -0.0865 + 0.1689(L_m/H_m) - 0.0426(L_m/H_m)^2 + 0.0036(L_m/H_m)^3$

$M_{wf} = 0.04\ L_m^{4/3}$

$M_a = (2M_{ws}+L_m)(2M_{ws}+B_m)$

$M_t = (H_m+2M_{wf})$

The relative mold material cost, C_{dm} can be determined as follows:

At $M_t=275$: $C_{dm} = 1.5582+0.0203\ M_a-0.00001\ M_a^2$

At $M_t=225$: $C_{dm} = 1.1989+0.0201\ M_a-0.00001\ M_a^2$

At $M_t=275$: $C_{dm} = 1.1548+0.0176\ M_a-0.000009\ M_a^2$

At $M_t=275$: $C_{dm} = 1.0483+0.0153\ M_a-0.0000073\ M_a^2$

At $M_t=275$: $C_{dm} = 0.8054+0.015\ M_a-0.0000092\ M_a^2$

At $M_t=275$: $C_{dm} = 0.7599+0.0123\ M_a-0.0000068\ M_a^2$