1-1-1990

A feature based approach to the integration of design, manufacturing and process planning

Richard Mark Schulte
Iowa State University

A feature based approach to the integration of design, manufacturing and process planning

by

Richard Mark Schulte

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major: Mechanical Engineering

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa
1990

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This thesis presents a structure for an integrated CAD/CAM system in wireframe models that can be simultaneously used for product design and manufacture of prismatic components. A feature driven, process based design methodology, which derives its basis from traditional geometry based design, feature based design, deterministic and expert heuristic manufacturing knowledge is proposed. This methodology provides an integrated modeling environment for the automation of downstream manufacturing applications such as process planning and numerical code generation. Representation of features in the design database is not limited to the geometry alone but includes material, tooling, and machine specifications and is based on a process to geometry mapping. This incremental, process driven design methodology has been implemented as a prototype system in the geometric modeler, AutoCAD. The research shows that informational completeness of a CAD database depends on the application; and for manufacturing tasks, the vocabulary of the process describing the means for making the part should be mapped directly to the feature based geometry. This effort answers key issues in integrating CAD and CAM in the most widely used wireframe modeling systems.

1 INTRODUCTION

The current generation of computer aided design and manufacture systems (CAD/CAM) are lacking a data base which is suitable for communicating complete design information between each application module. However, certain 'niche' applications of the current geometric modeler databases have been successful. These include automatic mesh generation for finite element analysis and transfer of geometry information to numerical control programming systems. In these applications, information which is passed downstream is a simple collection of vertices, lines, and arcs in either a 2 or 3 dimensional format. The optimization of CAD/CAM data structure integration is still necessary.

It has been suggested (Cutkosky et al. 1988, Dixon 1988 and Hummel and Brooks 1986) that a richer set of information can be passed to downstream applications by developing a more comprehensive database in the geometric modeler (CAD system). The methodology to accomplish this task is the subject of considerable research (Dixon et al. 1987, Miner 1985).

A complete database of the product to be manufactured should include information concerning material, tolerances, surface coatings, weight, size, production methods, inspection criteria, and so on. The generally agreed upon scheme for accomplishing this task is to represent the geometry (line, arcs, circles) as discrete combinations of entities known as features. Features are typically pockets, holes, or protrusions and have associated with them geometrical information as well as locational, tolerance information, and material information. This provides the basis for developing a complete database.

Correct representation of the feature attribute information in a database combined with a subset of artificial intelligence (AI) known as expert system techniques, produces a database which is a truly powerful and useful tool. The expert system software adds to the standard database, which consists of geometric entities, information relevant to different applications. This new type of database can be used to support a number of downstream applications which currently rely exclusively on human intervention and expertise to accomplish. Examples of the applications of the database include process planning, cost analysis, and manufacturability analysis. Using process planning as an example, this new type of database includes a list of tooling needed to machine the part, the cutting speeds and feeds for the tools, a description of the material, a list of the processes required to machine the part as well as the geometric information describing the part.

To fully appreciate the use of features in the database, consider the traditional partition which exists between design and manufacturing knowledge. This partition is the result of two problems. Lack of communication between design and manufacturing is the first problem and can be elicited by environmental or psychological factors. It is not uncommon for the design or research and development center to be located a considerable distance from the manufacturing facilities. This forces product designs to be well developed before the design and manufacturing people interact. The discourse between the manufacturing and design staff typically results in a number of changes which need to be made to facilitate the manufacture of the design. These problems could be avoided if manufacturing and design interacted at an earlier point in time, however short lead times on new product design and day to day problems each faction must deal with usually preclude this interaction.

The second problem of the manufacturing/design partition is the scarcity of one individual who is competent in the design of the product as well as the selection of the optimal manufacturing plan. This is the result of how industry typically trains neophyte engineers. An engineer fresh from college, who has chosen to be a designer, is taught by designers. The same is true for the manufacturing engineer; he is taught by the manufacturing engineer. Thus, both disciplines are taught from the beginning to be experts in one facet of the product life cycle; all other aspects, while not ignored completely, are not emphasized.

The result of this partition is frequently a product, that may perform its task competently, but is more difficult to manufacture than need be. When this happens, the redesign of some portion of the product is required resulting in added cost to the consumer. To avoid the problem of an unmanufacturable product, persons with manufacturing knowledge should be involved at the earliest possible time in the design cycle. This, unfortunately, completes the loop, and again presents the problem of the manufacturing/design partition.

The use of a feature based database and the expert system technique can simulate the required interaction between design and manufacturing personnel to some extent. This will have the advantages of reporting design problems when they occur (real time error detection) and also providing a set of engineering features which will reduce the time required to construct the geometric representation of the product.

The objective of this research is to present a taxonomy for the representation of form features within a wireframe modeling system. The database created from the features will then be used to create process plans and numerically controlled machining center programming statements necessary to machine parts from a specified raw material. This is accomplished by interaction with other databases, such as the tooling and material databases. Also, during the execution of this process it is desirable to offer advice concerning manufacturability, to recognize inconsistencies in the design, and to offer solutions to certain design problems. This system will eventually be part of a comprehensive integrated CAD/CAM system, being developed at Iowa State University, which will be used to research product design and manufacturing activities.

1.1 Research Goals

The goal of this research is to create an integrated system which simultaneously allows design, manufacture and process planning activities to be performed. The focus of this work is on prismatic parts which may be machined from stock on a three axis machining center. The result of this work will manifest itself in four ways:

1. A geometrical representation of the product.
2. The necessary sequence of machining operations which must be

performed to produce the product.

3. A listing of the cutting tools required to support the machining operations.

4. A logically sequenced list of computer numerical control machine tool statements (an N.C. program) which will execute the required machining operations.

1.2 Thesis Organization

The remainder of this work is divided into five sections. The next section, denoted section two, is a review of the literature available on the subject of CAD/CAM integration. This section explains feature based design, feature representation and feature extraction techniques and examines a number of process planning systems developed by the academic community. The third section explains the author's development of an integrated design/process planning system. Following this, an example of the implementation and verification of the system is given. Section five discusses the results of this work and makes comparison to current CAM/CAM systems. The last section, section six, draws conclusions and offers suggestions for future research in this area.

2 REVIEW OF LITERATURE

Much has been written about the integration of computer aided design and manufacturing. Traditionally the modules of a CAD/CAM system have been separated, or at best, restricted to passing simplistic representations of product designs between modules. Newer CAD/CAM systems are working towards true integration of design and manufacturing modules. However, this is still a goal at this time. The subsections of the literature review describe the work which is being performed in the research community towards the creation of a new generation of design and manufacturing software. This new generation of software will provide the designer with real time advice concerning manufacturability, a database which will support downstream applications and a set of geometric features with which the design may be rendered.

2.1 Computer Aided Design and Manufacturing

The widespread use and low cost of the modern digital computer has made accessible to all sizes of industries the tools known as computer aided design and manufacture. Typical, widespread uses for CAD/CAM technology in the early 1990s are:

1. Mass property calculations
2. Interference checking
3. Geometry definition
4. Finite element analysis
5. N.C. program generation

The above applications are performed at a level which requires considerable human interaction. To further automate these processes a fundamental change in the data structure currently being used to represent the product is required (Bond and Chang 1988). Applications which are not available at the commercial level due to the deficiencies in the CAD/CAM database include process planning, automatic Group Technology (GT) classification, manufacturability evaluation, assembly analysis, economic justification, etc.

2.1.1 Geometric Modeling Systems

A plethora of schemes are used to represent the product design in current CAD/CAM modeling systems. There is no database standard among vendors of CAD/CAM modeling software, however there are three broad methodologies used for the graphical representation of the design:

1. Wire frame models
2. Surface models

3. Solid models

Wire frame models use a three dimensional coordinate system to define the end points of lines and arcs. Models of this type are usually the most difficult of the three methodologies to visualize mentally due to limited hidden line removal capabilities. The main advantage of the wireframe model is the modest cost of software and the ability to operate on low cost computers.

A surface model represents objects in terms of points, vertices and faces between the edges of the object. A surface modeler can readily display the sculpted surfaces of an automobile fender or an airplane fuselage. Hidden line removal is also available with this class of geometric modelers. However, mental visualization is still poor.

The solid model is the most powerful three dimensional modeling tool as well as the most expensive and demanding on computer hardware. A solid model can provide information about volume and mass properties of the design, a distinct advantage over the other two modeling methodologies. Various solid modeling representation schemes have been proposed. Two have emerged as the most useful: constructive solid geometry (CSG) and boundary representation (B-REP).

As shown in Figure 1, CSG describes the part in terms of solid primitives such as blocks, cylinders, cones, etc. Boolean operations are performed on the solid primitives to create the final part shape. Boolean operations are geometric unions, differences, and intersections of the various primitives. Also, geometrical cross sections can be taken at any plane which intersects the part.

The boundary representation approach to solid modeling, as shown in Figure 2, uses faces, edges, and vertices to create shapes. These entities are then used to create a tree of connectivities.

2.1.2 Computer Aided Manufacturing Systems

Computer aided manufacturing systems cover a wide domain of applications. The most popular CAM system is the N.C. programming workstation. These workstations are used to create programs which instruct computer numerically controlled machine tools (milling machine, lathes, grinders, sheet metal punches, welders, lasers, etc.) to perform certain tasks. The input into the N.C. programming system is either a blueprint of the part

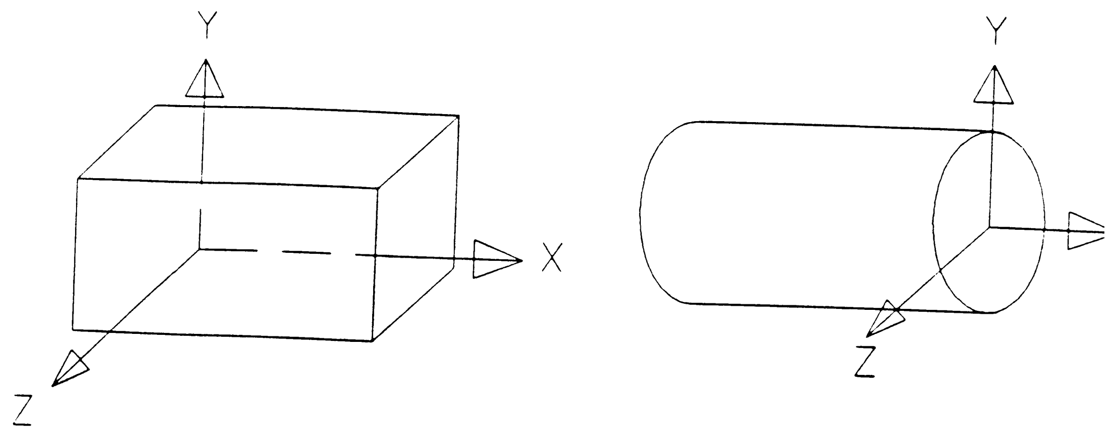

Figure 1: Constructive solid modeler geometry

Figure 2: Boundry representation (B-REP) solid modeler

or a database link to the geometric modeler. In the case of the database link to the geometric modeler, a data transmission standard such as the initial graphics exchange standard (IGES) supplies a list of points, lines, arcs, and circles to the N.C. programming station. A manufacturing engineer is then required to interpret the geometric entities and to select the correct sequence of tools and process parameters to physically transform some initial material into the configuration desired. Once this sequence is identified, a high level programming language such as APT, or COMPACT II is used to write a part program.

Other commercially available uses for CAM systems include:

1. Inspection: Used to program coordinate measuring machines (CMM) in much the same manner as the N.C.
programming workstation.
2. Forgability analysis: Used to analyze the plastic flow of materials. Provides insight to the problem of die filling and material stress conditions.
3. Casting analysis (riser design): Used to optimize the size of the casting riser and to check for the possibility of hot shortness in the

casting material.

4. Assembly simulation: Used to simulate the assembly process and to design assembly lines or cells.

5. Robotics: CAM systems are used in this application to program robotics for a variety of industrial uses.

2.2 Automated Process Planning

Process planning is the function of interpreting a pictorial representation of a component and deriving a sequence of manufacturing processes to transform specific raw materials into the finished part. The manufacturing processes required for process planning cover a large domain space. Thus, the human process planner must acquire many years of experience to become a competent process planner. Often, the process planner specifies processes and the sequences thereof based on previous experience of what has worked in the past. While the process plan specified by the process planner may not be the most cost effective method of manufacture, he/she is usually content with specifying a process plan which possesses a high probability of success on the first production run. Thus, it can be said that the process planner develops the process plan based on heuristic rules (Hummel and Brooks 1986, Vaghul et al. 1985). Heuristic rules are rules of thumb. An automated process planning system, in essence, is the collection of the heuristic rules the human process planner uses and a methodology for the application of these rules to a CAD model for the generation of a process plan.

The emergence of the subset of artificial intelligence known as expert systems has made the construction of automated process planning systems an achievable goal. An expert system ia a collection of a set of rules obtained from a human expert and coded into a computer program by a knowledge engineer. Knowledge engineers write programs using artificial intelligence for a multitude of applications. The goal of an expert system is not to select the best answer, but rather an answer which is 'good enough.' Expert systems for the process planning problem have been developed by Hummel and Brooks (1986), Vaghul, Dixon, Zinsmeister and Simmons (1985), Juri, Saia and De Pennington (1990), Shah, Hsiao and Robinson (1990) and Shah and Miller (1989).

In all of the cases listed above, the general structure of the systems is to first develop a modeling environment which provides the designer with a set of features from which to construct the part. Features are geometric forms such as counterbored holes, pockets or bosses. These features are then analyzed by a knowledge based expert system to develop a process plan for the part simultaneously during the design session.

2.3 Future Direction

The task set before the developers of CAD/CAM software systems is the integration of the two disciplines (design and manufacturing). Expert systems will play a major role in the integration of CAD and CAM.

In the long term, system developers are working towards the stage where a design from the design department will directly drive a material requirements system (MRP) and all of its attendant functions. These functions include purchasing, inspection, process planning, facilities planning, master production scheduling, etc. Much work is still to be done before this type of system is a reality.

3 FEATURE-BASED DESIGN AND REPRESENTATION SCHEMES

3.1 What is Feature Based Design?

Cunningham and Dixon (1988) formally define a feature as:

"...higher order abstract geometric forms or entities that are used in reasoning about the topology and geometry of designed artifacts during various design and manufacturing activities...."

Feature based design systems are systems which present the designer with features rather than the lines, circles and arcs which are prevalent in the majority of current CAD systems. This allows the designer to work at a higher level of abstraction. For instance, if the designer requires a threaded hole, he can simply select a threaded hole from a features menu. The system will have specific knowledge of the threaded hole and as such the designer need not be concerned with details such as minor diameter, pitch diameter and standard diametrical tolerances. The system already possesses this knowledge, thus the designer is freed from tedious details.

Vaghul et al. (1985) state the advantages of using features when designing mechanical products as follows:

1. It makes unnecessary the interpretation of the CAD model after the design is completed to extract features. The "building" of features is already completed. Current CAD/CAM integration methods rely on reading the CAD database and extracting geometric features for use with other applications such as N.C. programming after the design is complete.

2. Designing with features provides domain specific feature primitives to facilitate design. For instance, if an electronic manufacturer needs to repeatedly make molds for multi-pin electrical connectors, the multi-pin connector can be coded into a feature within the CAD system. This frees the designer from drawing, in some cases, twenty six or more circles representing the individual female connectors and the housing for the connector.

3. The feature database can be used by many other downstream applications such as finite element analysis, process planning and CAM systems.

3.2 Feature Types

According to Shah and Rogers (1988), most mechanical products may be defined with three types of features:

Form features: These describe the nominal geometric configuration of the mechanical product. Examples of form features are cubes, cylinders, and pockets.

Precision Features: These represent acceptable deviations in the form feature from the nominal configuration. The categories of these deviations are location, orientation, size, form and surface finish (Ranyak and Fridshal 1988). Each category is divided into classes of ANSI tolerancing standards. For example, the classes of the category 'location' are:

Distance: The length between two entities.

Angle from Base: The angle from one the three primary axis of the part.

Location: True position.

Surface Profile: Used for irregularly shaped parts.

Surface Runout: The departure from the desired axis of rotation and the actual axis of rotation of the part.

Precision features may be used for tolerance stackup analysis and assembly considerations.

Material Features: These fully describe the material from which the product is to be produced. Information in a material feature can include: material type, grade, heat treatment, surface coatings and physical properties.

Other feature types are imaginable, and may be application specific.

3.3 Feature Representation Schemes

The use of features to provide a complete database for part description has been proposed by Cunningham and Dixon (1988). There are two methods by which this may be accomplished: feature recognition (feature extraction) and design with features.

Jared (1983) advocates the feature recognition methodology whereby a model of the part is created using a boundary representation and simple geometric entities. After the design is complete, reasoning about the geometry is performed by the system. The purpose of this reasoning is to assemble the geometric entities into recognizable features, hence the term feature recognition. Jared divides features into two broad categories: depressions and protrusions. Depressions are features on the interior of the part while protrusions generally refer to the outside or exterior.

The methodology of feature recognition proposed by Jared (1983) is first to scan the boundary representation and examine all vertices, edges and loops. Next, edges are classified as concave, convex or smooth. Vertices are classified by default as convex. Only if two or more edges are concave, is the vertex classified as concave. The next step is a taxonomic classification and prioritization of faces. The faces are given a priority based on the number of convex vertices contained, the number of sets of concave edges and the number of concave edges in the face. Following this, the set of faces is examined, along with the data describing vertices and edges, and features are deduced by the system. These features are classified as pockets, slots, grooves, implicit protrusions, etc. The features describing the part are now explicitly described and may, at this point, be utilized by various applications such as process planning or forging analysis.

Henderson (1984) also advocates the use of a feature recognition scheme. The algorithm used by Henderson creates relations between geometrical entities. These relations are represented in the database as adjacent, opposite, convex, concave or perpendicular.

Others such as Kyprianou (1980), Woo (1982), Staley, Henderson and Anderson (1983), and Lee and Fu (1987) have also presented algorithms for the extraction of features.

The idea behind designing with features is to create the database concurrently with the design process, rather than after the fact as with feature recognition schemes. Also, design with features provides a more convenient user interface, manufacturing information and evaluation possibilities, and redesign recommendations (Cunningham and Dixon, 1988).

Vaghul et al. (1985) represent a feature in their database as the location and dimensions of the feature. This information is gleaned easily from the explicit geometric modeler information. The data structure here is used with an injection molding expert system, however the data structure can be expanded to include other information such as tolerancing or cost analysis.

A hierarchical structure for the representation of form features in a database has been introduced by Gindy (1989). The technique classifies the features according to their External Access Directions (EAD). An EAD is a possible direction from which a prismatic feature may be machined. A feature may have zero to six EADs. A feature with zero EADs is a protrusion, one EAD is a pocket, two EADs is a hole and so on. Next, features are classified as open or closed. An open feature is one in which it is not possible to traverse the perimeter of the feature entirely. A closed feature is one whose perimeter forms a loop. Finally, entry and exit planes are represented in the data base. If the entry and exit planes are parallel to each other and perpendicular to the feature's perimeter and depth axis, then the feature is classified as "through"; otherwise the feature is "not through".

The object-oriented programming approach of Hummel and Brooks (1986) also represents features in a hierarchy format also. However, Hummel and Brooks utilize the features to represent the volume of material which must be removed to form the final product from the rough stock. A simple rectangular pocket is represented in the data base as a feature which has as its attributes width, length, depth and corner radius. Each attribute also has associated with it an accompanying tolerance attribute. A pocket is a child of the "depression" class so it automatically inherits a list of surface faces and entrance faces. Hummel and Brooks have specified the root class as "feature".

3.4 Discussion

Each of the feature representation schemes given above has its own particular advantages and disadvantages. These will be discussed in a broad sense below.

A major disadvantage of designing with features is the infinite number of features required to represent all products and the finite number of features which can be presented in the user interface of a geometric modeling system. Research into defining a closed set of features has been conducted by Pratt and Wilson (1985). They have defined a set of features that are used in a number of engineering applications. However, even this extensive study will not envelop all the features required to represent all products.

Feature recognition and extraction attempt to classify the geometry from the geometric modeler and to classify all shapes according to a finite set of heuristic and deterministic rules. For instance, a rectangular pocket with three of its four corner radii the same and the fourth one of another size could be classified as a pocket quite easily with a feature extraction scheme. This same pocket can be just as easily represented in the design with features scheme if the rectangular pocket with one dissimilar corner radius is a choice on the user interface. If it is not an option, the feature must then be designed and coded into the system. Coding a seldom used feature may be laborious and also inefficient if the feature is used infrequently.

Algorithms for feature extraction quickly grow complex for even simple product representations (Shah and Rogers, 1988). Thus, the inverse of the above argument may be stated as: Can a finite algorithm completely classify all product geometry?

To be computationally efficient and accurate, the feature extraction methodology should be performed at the conclusion of a modeling session, or sporadically during the modeling session. This allows inconsistencies in the geometric model to be built upon for a certain period of time before the inconsistency is discovered. By designing with features, the frames representing the features can be continuously updated. Thus, feature-to-feature spatial relationships such as shared vertices, edges, and surfaces are checked at each juncture of the design process and inconsistencies are identified for immediate correction.

3.5 Need for Present Research

The new generation of feature based design systems promises to close the gap in CAD/CAM generation. It represents the next logical step from existing geometric modelers and captures the manufacturers' intent at the design stage. By designing with features, checks can be made on the feature database with each addition or modification to the database. Although these systems provide explicit representation and reasoning for the part to be processed, they have certain limitations.

As stated previously, a major disadvantage of designing with features is the infinite number of features required to represent all products and the finite number of features which can be presented in the user interface of a geometric modeling system. Pratt and Wilson (1985) have defined a set of features that are used in a number of engineering applications. However, even this extensive study will not envelop all features required to represent all products.

Features are attractive for CAD and Computer-Aided Process Planning (CAPP) because they raise the levels of abstraction and provide more complete information to the planner. However, FBDS still need some form of recognition mechanism when used with application shells; that is, application programs need to be written to facilitate easy integration, though the mapping is done at a higher level.

Features are important from a manufacturing standpoint, but an integrated approach towards process-driven design is best. Process planning should not occur after the design is complete, as manufacturability cannot be built into the product after the design is finalized. This idea is also echoed in the work of Cutkosky et al. (1988). Process planning should be attempted as the design evolves, which is termed programming-by-design, and this incremental mode derives its benefits from features and the process driving the features. It is felt the best approach would be to associate every feature in the geometry with the process motion that generates the feature, so that a process plan is readily available as the design progresses and obviates the intermediate step of process planning. Also, it is clear from most of the literature that FBDS alone are being created first and then mapping shells are designed for generating process plans based on the FBDS. Further, all of these systems reported in the literature are either prototype systems or the concept has been proven in solid modeling systems. Features alone do not help to assist manufacturing automation. A mapping of the process to the feature as the design evolves seems to be the best approach and will result in the implementation of the feature based, process driven methodology in a wireframe modeling system. The system elaborated in the next few sections validates this philosophy as a practical, viable, and highly useful modeling technique.

4 DEVELOPMENT OF INTEGRATED CAD/CAM

4.1 Overview of Data Structure Implementation

Data abstractions flexible enough to overcome the basic limitations of wireframe models and to provide for future expansion have been implemented. Features are portrayed as a collection of various objects, consisting of attributes and their relationship between the objects. The information is stored in a hierarchical format such that instances of a feature can acquire and inherit information from its super-classes. The master template is a linked list that provides the skeleton with pointers to the appropriate subobjects that represent information such as tooling, material properties, and machine tool specifications. The master template corresponds to the feature database, and the instances of the feature containing tooling, material, and other information are also linked lists that are used to store properties and attributes of each feature (see Figure 6). An instance of a feature is created when the user progresses through the queries of the master template and each node has a correspondence to its superclass, the feature itself. Definition of the subclasses is attempted by filling the slots with appropriate information. Pointers are used to specify the order of features instanced in the database that would be later used in the process plan generation. Defaults, for elevation and location, are provided for each feature class to accommodate standard representation.

The interpretation of the feature's data is implemented in the process module. The parameters are either available explicitly or need to be evaluated. Evaluated parameters are either simple calculations or those which depend on relationships between features. As features are instanced, the relationships are obtained by an interrogation procedure in the process module that allows for direct mapping of the process. The process module goes through a series of rule checks with a simultaneous extraction of explicit information needed for each check and computes the feasibility of the instanced feature. Upon successful interrogation, the information is used to generate the process plan and N.C. code.

4.2 Geometric Modeler and Data Modules

The development of an integrated CAD/CAM system is the objective of this work. To that end, a prototype design with features system has been created and implemented. The geometric modeler chosen for this task is Autodesk's AutoCAD. To create a complete design with features/process planner, four modules were added to the basic geometric modeler. These modules are:

1. A design module composed of a set of features such as holes, pockets and protrusions for use when creating part geometry.

2. A collection of databases consisting of tooling and material types.

3. A module containing a set of heuristic rules which selects process plans and the parameters thereof.

4. A module for the creation of COMPACT II programming language statements.

These modules produce, simultaneously during the design session, a process plan for the part being designed and a set of programming statements for use with a computer numerically controlled (CNC) machining center. No interaction with the designer is needed to develop the process plan or the COMPACT II program.

4.3 Design Module

The design module developed is the front end of the design with features system which interacts exclusively with the designer. The design module allows the designer to function at a higher level of abstraction than is the case for conventional modeling systems through the use of features. The designer can now design with threaded holes, pockets and slots rather than lines, arcs, and circles. The features offered to the designer in the system described here are of the following feature types.

1. Outside Features: These features are used to describe the perimeter of the part. A part may possess more than one outside feature. The specific features in this class are:

A. Cube

B. Filleted cube

C. Chamfered cube

D. Cylindrical protrusion

2. Inside Features: These features describe the depressions or pockets the part possesses. Multiple inside features are allowed. The specific features in

this class are:

A. Cube

B. Filleted cube

C. Chamfered cube

D. Cylindrical depression

E. Annular groove

F. Slot

3. Hole Features: As the name implies, these features represent holes. The specific hole types available in this system are as follows:

A. Simple hole

B. Counter bored hole

C. Close diametrical tolerance hole

D. Tapped hole

Each of these hole types may be selected from the user interface as a single hole, a circular array or a rectangular array.

4. Raw material feature: This feature describes the configuration of the raw

material from which the part is to be manufactured. Raw material configurations supported by this system are:

A. Round cornered prismatic stock

B. Square cornered prismatic stock

C. Round bar stock

Each of the feature types is implemented in such a way that they minimal interaction with the designer to incorporate them into the design. For example, when selecting a tapped hole, as shown in Figure 3, the designer need only know location, depth and thread type. The knowledge base converts the thread type to the correct major, minor and pitch diameters and displays them via the graphical interface. Another example of this is the selection of the counter bored hole type as shown in Figure 4. The designer need not be cognizant of the head and shank sizes of a socket head cap screw, but merely knows the size of socket head cap screw required for the design. The knowledge base converts the socket head cap screw size to the correct head size, shank size, and head depth and also adds a clearance for each of these attributes. Again, this allows the designer to work at a higher level of abstraction than with a conventional CAD system.

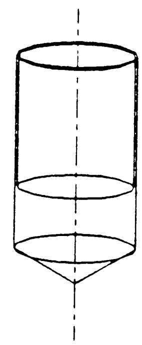

Figure 3: Tapped hole

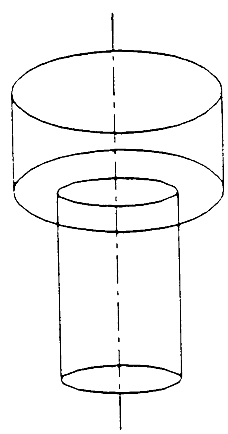

Figure 4: Counterbored hole

As a parallel to CAD systems in which lines, arcs and circles are instanced in any sequence the designer chooses, the system described here allows the designer to select features in any order he chooses. The order in which features are selected has no bearing on the final process plan derived. Also, as with any CAD system, the system described here allows the designer to view the design from any face or projection he wishes. This is shown in Figure 5.

Features are ultimately represented in the drawing data base as a series of files. Each file contains a singular feature and specific information describing its geometric components such as length, height, elevation, etc. To allow other modules of the design with features system to access and use the specific feature information, the information describing each feature is placed in a discrete file. Each file is named with a group coding scheme consisting of five fields. Each of the fields in the code represents a particular aspect for the feature to the process planning system.

A. Field one (single digit): specifies whether the file contains geometry or tooling information. This field can have the value of 1

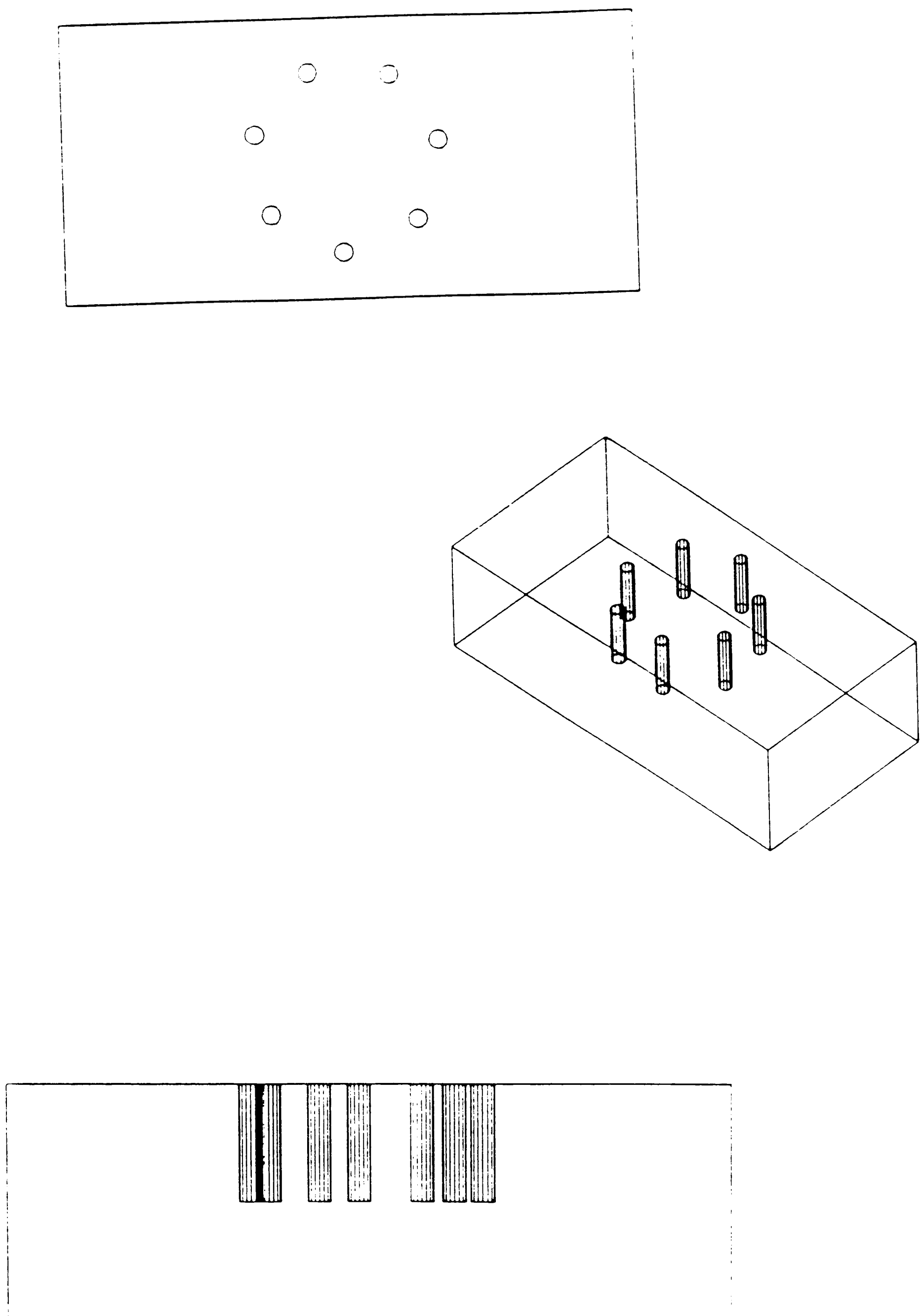

Figure 5: Multiple views of simple part

(geometry) or 2 (tooling).

B. Field two (single digit): specifies raw material, outside feature, inside feature or hole. Raw material is represented as a 0, an outside feature as a 1, an inside feature as a 2 and a hole is represented as a 3.

C. Field three (two digit): specifies the feature type (right circular cylinder, hole, pocket, etc.). The numerical codes this field can assume are:

Cube	10
Filleted Cube	20
Cylinder	30
Chamfered Cube	40
Annular Groove	50
Counterbored Hole	60
Reamed Hole	70
Tapped Hole	80
Drilled Hole	90

D. Field four (single digit): This field is set to zero for the case of a geometry feature. If the file is for tooling, this digit will represent the type of cutting tool. The numerical values this field may assume are as follows:

Center Drill	1
Twist Drill	2
Tap	3
Counterbore	4
End Mill	5
Reamer	6

E. Field five(two digit): a sequential counter which eliminates the chance that a file name would be duplicated.

As an example, the file named 2360423 would signify to the system the following information:

1. This is a tooling file (2).
2. The feature is a hole (3).
3. The hole type is counter bored (60).

4. The file contains process information about a counterbore tool (4).
5. This is the 23rd file created during this design session (23).

4.4 Database Modules

To support the process planning function, databases for tooling, material and, ultimately, machine tools are required. Figure 6 shows the overall structure of the design with features system and the interactions between each of the modules. Information shown below for each database module indicates the information processed by that particular module. The user of the system interacts with the system at the user interface level exclusively.

The other modules are not directly accessible to the user. The machine tool database is absent from Figure 6 for reasons which will be explained later. The structure of each of the databases will now be examined.

The tooling database must contain information about the tool types which are available. This is machine dependent as well as facility dependent. Also, the available tools are in a constant state of flux, therefore the tooling database must be readily updated to reflect both new tool types and the

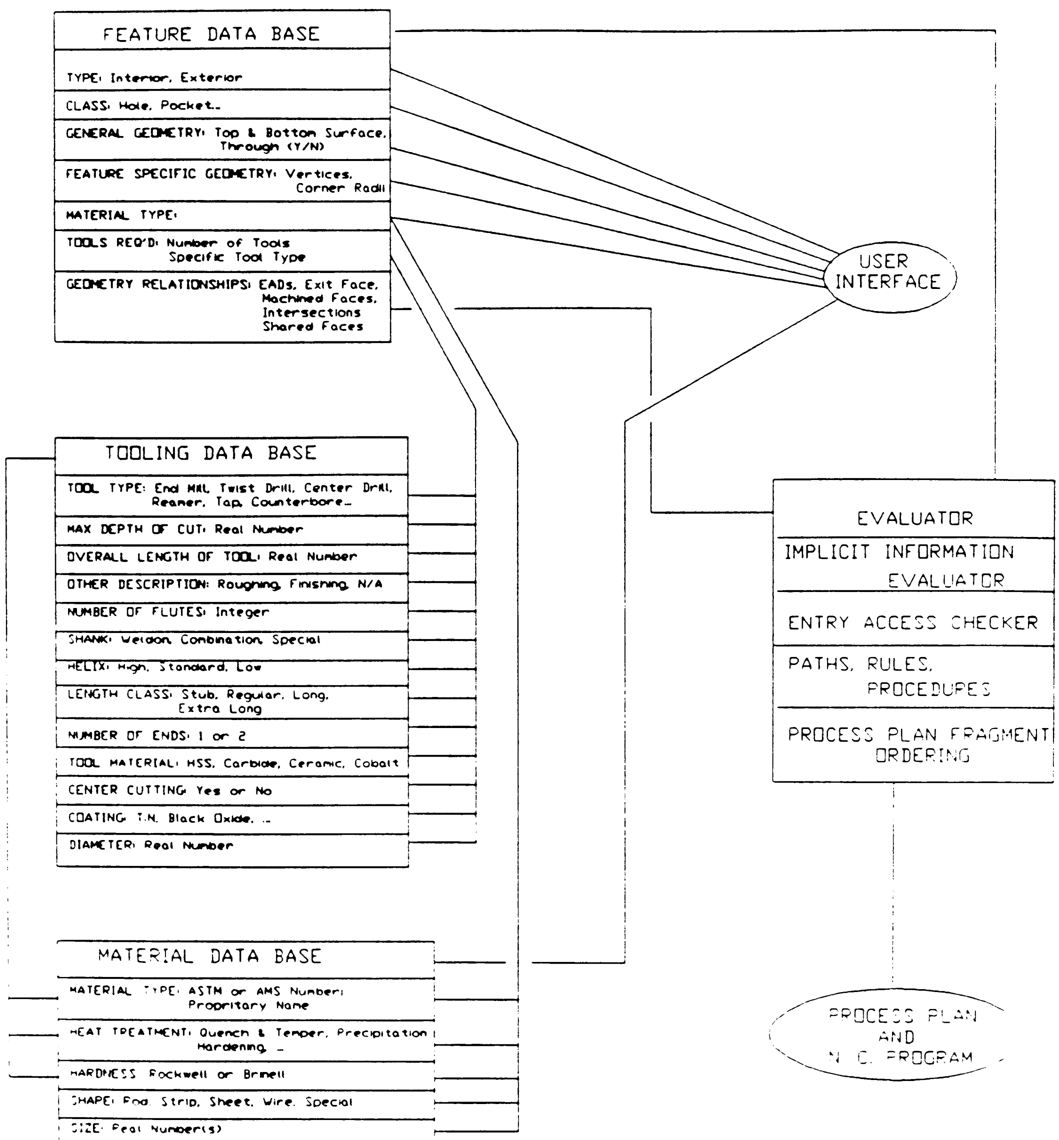

Figure 6: System data structure

absence of worn or scrapped cutting tools. These requirements point to the use of a data structure which allows modification at the user interface level as well as a flexible structure at the system level. This suggests the use of LISP frames for the tooling database. A segment of the tooling database for a 1.00 inch diameter end mill, is shown in Figure 7. This segment implicitly

```
(MILL_TOOLS                    ;machine type
 (END_MILL                     ;tool type
  (DIA (1.0)                   ;tool diameter
     (D_O_C (2.0 4.0 6.0))  ;maximum depth of cut
     (LENGTH (5.0 7.0 9.0)) ;overall tool length
     (TYPE (FINISHING ROUGHING));cutter description
     (NO_FLUTES (2 3 4))    ;number of flutes
     (SHANK (WELDON COMBINATION))  ;shank description
     (HELIX (HIGH STANDARD SLOW))  ;cutter helix
     (L_CLASS (STUB REGULAR LONG)) ;length class
     (NO_END (SINGLE DOUBLE))      ;number of ends
     (T_MAT (HSS COBALT CARBIDE))  ;tool material
     (CENTER_CUT (YES NO))      ;center cutting Y/N 2
     (COATING (TiN BLACK_OXIDE))   ;tool coating type
     )
  (DIA (1.125)                 ;next tool diameter
     ....
```

Figure 7: Lisp cutting tool frame

states all permutations of each tool type are available for use. In practice this would not be true, thus there would be a number DIA attributes with the value of 1.00 inch and the subclasses would specify the available tools. The structure shown in Figure 7 shows all values the subclasses can assume.

A graphical representation of the tooling database taxonomy is shown in Figure 8. Each tool type (drill, reamer, tap, etc.) under each machine tool has attributes similar to the attributes shown above.

The material database must also offer flexibility. New materials are constantly entering the market and the database must allow for the addition of these materials. Again, LISP frames are used for the representation of the material database, thus the structure of the material database is similar to the tooling database. The information contained in the material frames are as follows:

1. Cutting tool information
   - i. Size
   - ii. Cutting speed
   - iii. Chip loading
   - iv. Coolant information
2. Heat treatment data (if applicable)
3. Physical properties such as tensile strength, hardness and size.
4. Other attributes as needed

(Refer to Figure 9 for a graphical representation of the taxonomy of the

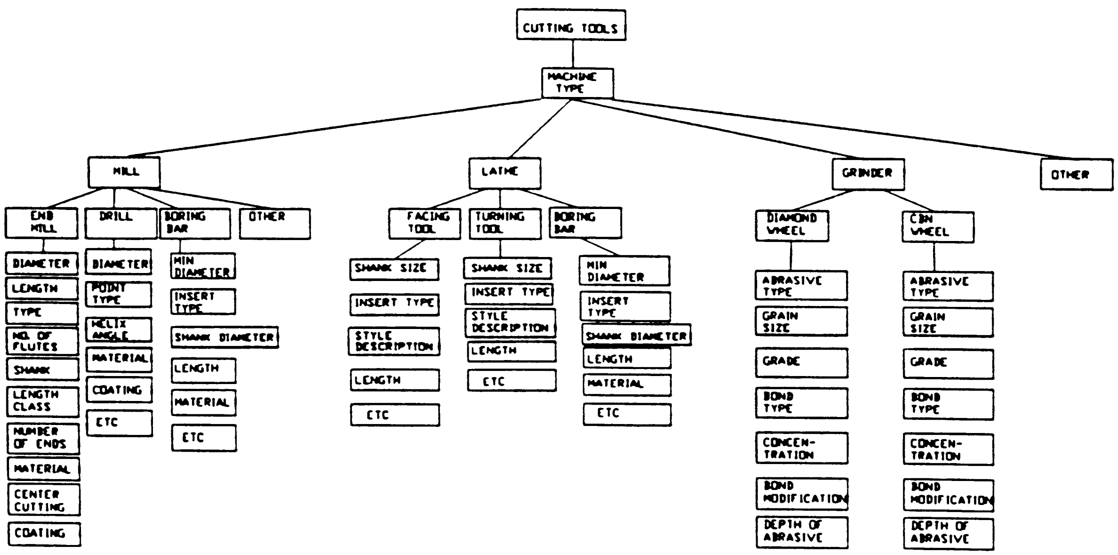

Figure 8: Taxonomy of tooling database

material database.)

Specific values for each of the above cutting condition attributes is dependent upon the tool type used. For example, a 1.00 inch diameter end mill will sustain a heavier chip load than a 1/8" diameter end mill. Thus, cutting speed and chip loading can assume several different values, not only for each tool type, but each tool size within the tool type. Interpolation can be performed between specific tool sizes to produce the correct cutting conditions for the tool size under consideration.

A machine tool database is also required. A machine tool database provides support to the process planner and assists in determining the optimal machine tool for each facet of the process plan. The machine tool database contains information such as spindle speeds, horsepower, supported tooling, table motion limitations, material removal rates, lists of fixtures, and so on. At this time the machine tool database would only function to select the capacity, horsepower, and tool holding requirements. When the system is capable of recognizing parts other than prismatic, more machine tool types will need to be added to the machine tool database. Hummel and Brooks (1986) have stated that the machine tool database will also be necessary for the evaluation of the availability of machine tool resources. From this, the

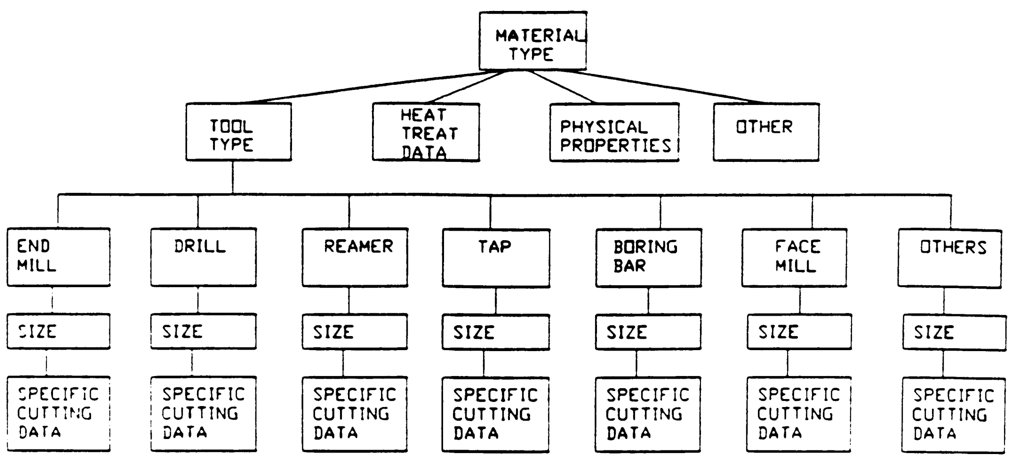

Figure 9: Taxonomy of material database

feasibility of automated scheduling can be inferred. Since this preliminary application of a design with features system only addresses three axis machining centers, a machine tool database is not included in this work.

4.5 Process Module

The purpose of the process module is to select the process or series of processes which will produce the feature instanced by the designer. The process planning function is performed by the system in two parts. In the first part, the feature is represented graphically on the video screen and the processes required to machine the feature are identified. This is done each time a feature is instanced. The second task of the process planner is to analyze the features instanced and the processes required to machine them, order the machining processes into a sensible sequence and finally, create an N.C. program. This is performed when the designer signals he has completed the design session for a particular product.

By designing with features, the process planning function can be implemented in a rather straight forward manner. Consider that, within the domain space addressed, features have associated with them a discrete set of process plan options. The correct process plan option which should be selected is dependent upon the size of the feature and the feature's proximity to other features. For example, a simple hole in a metallic prismatic block could be created by one of the following processes.

1. Center drill; then twist drill.
2. Center drill, twist drill and bore.
3. End mill using circular or helical interpolation.

While other processing options could be envisioned, these three examples will illustrate the concept.

Referring to Figure 10, suppose a query of the feature database shows there is a simple hole which is 0.500 inch in diameter and 1.000 inch deep. Querying the database further, it is known the simple hole does not intersect other features. Based on these two queries of the feature database, the process planner would select option 1, shown above, as the preferred method of creating the hole. Option two was not chosen as it is not practical to bore a .500 inch diameter hole. Option 3 was not chosen because there is no need for circular interpolation as the .500 inch diameter hole can be produced quite easily with standard size twist drills.

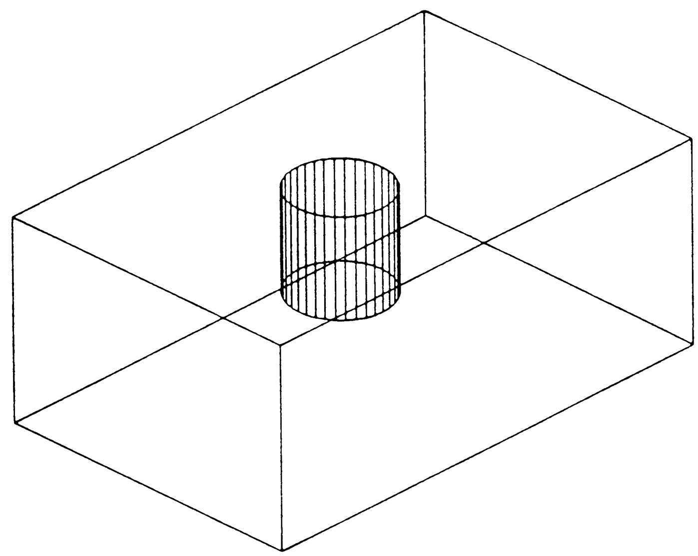

Figure 10: Simple interior feature

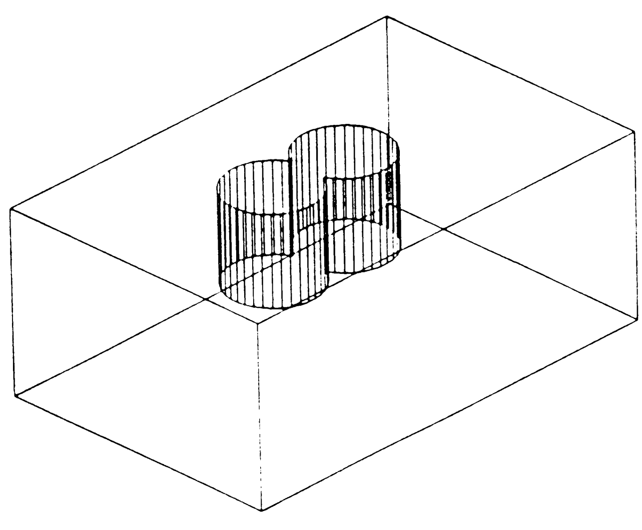

Figure 11: Two intersecting features

The logic explained above is represented in the process planner as heuristics.

As another example, suppose the same size of simple hole is represented in the database, but this hole intersects another feature. This situation is shown in Figure 11. Since trying to drill a hole which intersects another feature will cause the drill to wander or possibly suffer catastrophic failure, option 1 is not a valid process. The process planner knows a simple hole opening into another feature is easily machined using a center cutting end mill and helical interpolation cutting motions, thus option 3 is chosen. The other option shown above, option 2, is needed in the case of very large holes or when larger holes must be produced to non-standard sizes.

Other features such as pockets, protrusions, or grooves also have a discrete set of processing options available to create the feature geometry. Thus, while the design is being created in a CAD system, the process planner can associate process plan fragments to the features represented in the feature database. As new features are added to the database, conditions such as intersections with previous features are checked. If a condition exists which causes a previous process plan fragment to be invalid, the old feature's process plan can be altered.

After all the features have been represented, the database contains a series of process plan fragments associated with each feature. The process plan fragments do not have, at this juncture, an order in which they are to be implemented. This is the second task which the process planner must perform. Activating the second part of the process planner is performed via a selection on the user interface. A finalized, ordered process plan can be generated at any point, any number of times, in the design cycle. This allows partial process plans to be available at any point in the design cycle.

The second task of the process planner operates using both rules and a taxonomy to determine the priority in which the feature's process plan fragments are to be treated. The taxonomy creates a general sequence of processing which begins at the top/exterior and works toward the inside/bottom of the part. The rulebase resolves ambiguities in the ordering of the process plan fragments. An example of this is when more than one feature exists at the same elevation and equidistant from the center of the part. After the process plan fragments are ordered, selection of cutting tools is made by the rulebase and the numerical control programming statements are assembled. An example of a rule from the rulebase is shown below in Figure 12. This rule determines the depth and corner radius of an interior pocket and the amount of stock left by the roughing end mill to be removed. Based on these three facts, a suitable finish end mill is selected.

```
(DEFUN GET_TOOL_5 (MAXDIA MINDOC)
   (SETQ MAXDIA                ;get the maximum tool
                               ;diameter.
    (* 2.0 (FGET 'ID 'F_GEOM 'CORNER_RAD))
   (SETQ MINDOC                ;get the minimum depth
    (- (FGET 'ID 'GEN_GEOM 'TOP);of cut.
       (FGET 'ID 'GEN_GEOM 'BOTTOM)
    )
   )
   (WHILE (< EMDIA MAXDIA)     ;select tool diameter
   (SETQ EMDIA
      (FGET 'MILL_TOOLS 'END_MILL 'DIA))
   (WHILE (=\ nil EMLGH)
   (SETQ EMLGH
      (FGET 'MILL_TOOLS 'END_MILL 'DIA 'DOC))
   )))
```

Figure 12: Sample rule for cutting tool selection

The rule in Figure 12 selects the largest end mill which is still smaller than the required corner radius specified in the feature database. Thus, the corner radius can be circular interpolated to the correct size by using the machine tool's cutter diameter compensation. The rule also selects the shortest available end mill from the tooling database, thereby maximizing tool rigidity.

4.5.1 Rules Used to Generate Process Parameters

The selection of the process parameters such as cutting speed, feed, and depth of cut are derived from information contained in Machinability Data Center's (MDC) Machining Data Handbook. This two volume set is used widely in industry and is in fact the book used as a reference by the author to create this system. The MDC handbook contains the previously mentioned process parameters as well as many others for sixty one different classes of materials which range from free machining carbon steels to nonmetallic composites and plastics.

The methodology of coding the process parameter information into the process planning system is quite simple and follows the scheme of representation in the Machinability Data Handbook. Process parameters for each class of tool are represented in a material cutting parameter database according to the size class of the tool. For instance, cutting parameters for end mills are represented in the database as shown in Figure 13:

```
MATERIAL TYPE:  6061-T6 ALUMINUM ALLOY, WROUGHT
     DEPTH OF CUT - BURIED FEED:  CUTTER DIAMETER / 4
     SPEED, FEET PER MINUTE - BURIED FEED:  450
     END MILL DIAMETER RANGE:  1/8 to 3/8
     FEED, INCH PER TOOTH - BURIED FEED:  .004
```

END MILL DIAMETER RANGE: 3/8 to 1/2
FEED, INCH PER TOOTH - BURIED FEED: .006
END MILL DIAMETER RANGE: 1/2 to 3/4
FEED, INCH PER TOOTH - BURIED FEED: .008
END MILL DIAMETER RANGE: 3/4 to 2"
FEED, INCH PER TOOTH - BURIED FEED: .012
DEPTH OF CUT - PERIPHERAL FEED: CUTTER DIAMETER / 4
SPEED, FEET PER MINUTE - PERIPHERAL FEED: 600
END MILL DIAMETER RANGE: 1/8 to 3/8
FEED, INCH PER TOOTH - PERIPHERAL FEED: .003
END MILL DIAMETER RANGE: 3/8 to 1/2
FEED, INCH PER TOOTH - PERIPHERAL FEED: .006
END MILL DIAMETER RANGE: 1/2 to 3/4
FEED, INCH PER TOOTH - PERIPHERAL FEED: .008
END MILL DIAMETER RANGE: 3/4 to 2"
FEED, INCH PER TOOTH - PERIPHERAL FEED: .010

Figure 13: Sample process parameters

In Figure 13, buried feed refers to the end mill cutting material through 180 degrees of revolution while the peripheral cutting situation is generally a cut of less than 90 degrees of rotation. It should be noted that the cutter must be run more slowly in the buried feed mode of material removal when cutting in a peripheral mode. Also, a method such as air blast or coolant flood must be used to remove chips from the buried feed cutting path.

Since the cutting parameter database only lists certain size classes of cutting tools, linear interpolation within the size class is used to select specific cutting speed, feed and depth of cut. These cutting parameters are not optimum values, but rather they will provide a reasonable starting point for trial part processing. Optimum cutting parameters vary with machine tool, fixturing and cutting tool vendor and must be determined at the machine tool.

Other rules which were input into the process planning system for determining process parameters are largely heuristic in nature. The system has access to approximately 75 IF-THEN heuristic rules. A few of these rules, translated into natural language are shown in Figure 14.

4.6 COMPACT II Code Generation Module

The programming of numerically controlled machine tools is simplified with the use of high level programming languages created specifically for this purpose. These high level programming languages allow one sentence of program code to represent, in some cases, a page or more of machine readable G and M codes. A number of high level programming languages were considered for use with the design with features system. Figure 15 describes the languages considered.

1. IF the hole is more than 3 drill dia deep THEN peck drill
2. IF the milling operation requires a L/D greater than 3 THEN use a carbide end mill
3. IF cutting hardened ASTM 4340, THEN use Titanium Nitride(TIN) coated carbide cutting tools
4. IF using a face mill larger than 2 inches in dia THEN maximum depth of cut is .300 inches
5. IF cutting Aluminum THEN Do not use coated tools
6. IF machining an inside profile with an end mill, THEN use an end mill with a diameter that is smaller than two times the corner radius
7. IF the end mill is smaller than 3/4 inch diameter THEN Maximum radial depth of cut is one-half the cutter diameter
8. IF cutting Aluminum THEN use a high helix end mill
9. IF using a stub length end mill THEN can take 1/2 cutter diameter radial cut and two times cutter diameter axial
10. IF not making a finish cut THEN it is acceptable to use a roughing end mill
11. IF drilling a small hole THEN use center drill to a diameter slightly larger than the hole size
12. IF counterboring hardened material THEN dwell at the end of a counterbore cycle to achieve a better surface finish
13. IF milling an outside profile THEN unless specified otherwise break sharp corners 0.005 maximum
14. IF milling Aluminum THEN use a two flute end mill
15. IF plunge milling THEN use a center cutting end mill
16. IF rough milling THEN leave a small amount of stock for the finishing cut
17. IF tapping hard material THEN use a slow helix tap
18. IF tapping titanium, THEN relieve the heel of the tap by grinding
19. IF tapping large L/D ratios THEN use a slow tap speed
20. IF hole is to be tapped THEN tap drill the hole first
21. IF doing a regular machining operation THEN use coolant
22. IF hole is to be reamed THEN leave .002 to .005 for the reamer to cut
23. IF the correct size counterbore tool is not available THEN use a center cutting end mill and a circular interpolation cutting motion
24. IF tapping a blind hole THEN use a bottoming tap
25. IF minimum thread depth is specified, THEN the tap drill depth is four times the lead of the tap plus the thread depth.

Figure 14: Heuristics used in process planner

APT (Automatic Programmed Tools): A three dimensional language used to direct the cutting tool in a programmer defined path. APT is well suited to the machining of complex surfaces requiring simultaneous three axis (or more) contouring. APT typically requires a large computer.

ADAPT (Air Material Command Developed APT): This language is an extension of APT written to run on small to medium size computers.

AUTOSPOT (Automatic System for Positioning Tools): This is a single pass N.C. processor with three or more axis positioning but only offers limited continuous path capability.

SPLIT (Sunstrand Processing Language Internally Translated): Another multi-axis positioning language with limited contouring capability. SPLIT is oriented towards Sunstrand's own N.C. tools.

COMPACT II: A high level language developed by Manufacturing Data Systems Incorporated (MDSI). COMPACT II has universal application to a wide variety of machine tools and is offered as a software system by at least two software manufacturers currently.

Figure 15: Common high level machine tool languages

The COMPACT II programming language was chosen for this project for the following reasons:

1. The programming language can be used on a personal computer.

2. Instant translation of a standard COMPACT II program to machine readable code is achieved with readily available post-processors.

3. A plot routine internal to the COMPACT II software allows instant visual verification of the generated tool path.

4. COMPACT II is serviced nationwide by two companies, Automation Intelligence and Applicon.

5. COMPACT II is available at a reasonable cost.

A COMPACT II program consists of 4 parts. The first part is the information header. A typical header might be:

```
MACHIN, MILL9001
IDENT, ANC 101 TEST PART
SETUP, 0XB, 0YB, 0ZB
```

This header signifies, on the first line, the machine tool postprocessor to use. In this case the postprocessor for mill 9001 is selected. The next line is for informational purposes only and describes the piece part to be machined.

The last line, SETUP, defines the zero position of the programmable machine tool. Other information is typically placed on the SETUP line, however it is outside the scope of this report and will not be detailed here.

The next part of a COMPACT II program is the geometry definition section. This section is made up of sentences which define single geometrical elements which are then combined together to form the description of a part boundary. Suppose the designer using the design with features system selects the feature CUBE. The system would query the designer about the size of the CUBE and might deduce the following vertices shown in Figure 16.

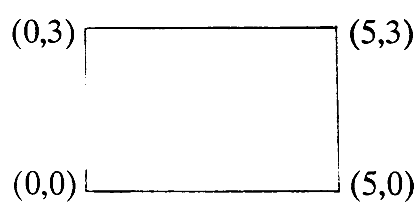

Figure 16: Vertices of a Cube: COMPACT II application

The vertices shown in Figure 16 would be used by the system to create four COMPACT II programming language sentences which define the four lines which make up the cube. The sentences are as follows:

```
DLN1, PT(0XB,0YB),PT(5XB,0YB)
DLN2, PT(5XB,0YB),PT(5XB,3YB)
DLN3, PT(5XB,3YB),PT(0XB,3YB)
DLN4, PT(0XB,3YB),PT(0XB,0YB)
```

These four line definition sentences would then be built upon to create a part boundary definition of the cube. This part boundary would manifest itself as follows:

```
DPB1,S(LN4),LN1;LN2;LN3;LN4,F(LN1),NOMORE
```

This part boundary statement's English translation is:
"Starting at the intersection of line 4 and line 1, connect line 1 to line 2, then line 2 to line 3, then line 3 to line 4. Finish the part boundary at the intersection of line 4 and line 1."

The third section of the COMPACT II program is the tooling section. This section defines the tools to be used to manufacture a particular product, the process parameters, and the path which the tool must follow. These sentences are created by the process planner using the heuristic logic contained within the system. Using the example of the CUBE shown above, a simple form of the possible tooling statement to machine the outside of the cube would be as follows:

ATCHG,TOOL1,TLCMP1,400RPM,.010IPR,.75TD,180TPA

CUT,PB1/CL,-1ZB,CON

RET,COF,STOP

These statements, in English, translate as follows.

The first statement:

"Using tool 1 and the tool length compensation register 1, start the spindle at 400 revolutions per minute and set the feed rate register to .010 inches per revolution of the spindle. Tool 1 is a .75 diameter tool with a 180 degree tool point angle (an end mill)."

The second statement:

"Cut part boundary 1 with the cutter on the left side of the boundary at a depth of 1 inch in the negative Z direction. Also, turn the coolant on."

The last statement:

"Retract the tool to the home position, turn the coolant off and stop".

The fourth and last section of a COMPACT II program is simply the one word sentence:

END

which signifies to the COMPACT II command processor that no further statements need to be evaluated.

All features contained within the system are evaluated in this manner. The geometry statements which need to be created are fixed, for a given feature, and do not depend on material or interactions with other features. The tooling statements are variable in nature, being ultimately controlled by the process planner's heuristic and deterministic rules.

4.7 N. C. Code Generation Module

As stated previously, the COMPACT II program must be postprocessed into a machine readable form. This process is similar in nature to compiling a FORTRAN, PASCAL or BASIC language program into assembly language. There are many companies who offer commercial software packages for the compilation or postprocessing of a COMPACT II program into machine readable language, most notably Automation Intelligence. The creation of a postprocessor which would reside within the design with features system was not undertaken for this project for the reasons stated above and the fact it was seen as "reinventing the wheel".

4.8 Summary

The design with features methodology has been combined with an effective process planning system. The design module consists seven different features which are commonly found on parts manufactured by three axis machining centers. The process planning module of the system makes extensive use of heuristic rules for the creation of process plans. The process plans are created by considering the workpiece material, the individual features size, and the tooling available in the tooling database. The output from the system is a COMPACT II machine tool program which must then be processed into a machine readable program.

5 EXAMPLES AND VERIFICATION

This section will explain the necessary hardware and operation system files configurations to operate the design with features system on Intel 80286 computers. Following this, two examples of the use of the system will be given. The first is an interactive session with the system which details the manner in which process plans and a geometric representation of the CAM-I ANC 101 test part are created. The second example, an MBB test piece, proves some of the finer aspects of the software and provides explanation of the validity rules involved in the creation of this modeling environment.

5.1 Software and Hardware Setup

The design with features system was developed on an Intel 80286 based

CPU: Intel 80286, 12 MHz
Video Display: Enhanced graphics, 14 inch
monitor
Hard drive: 32 megabyte
Memory: 1 megabyte (more is desirable)
Auxiliary input device: Mouse

Figure 17: Minimum computer system requirements

computer. The minimum specifications required for the use of this software system are shown in Figure 17.

The software setup necessary for executing the design with features system is placed in two files. The autoexec.bat file, which is executed each time the computer is powered up, needs to have the commands shown in Figure 18 added to it.

```
SET ACADCFG=\ACAD10\WORK
SET ACAD=\ACAD10
SET ACADFREERAM=24
SET LISPHEAP=43000
SET LISPSTACK=2000
```

Figure 18: Commands included in AUTOEXEC.BAT File

The first statement in Figure 18 gives AutoCAD the directory location of the AutoCAD configuration file and all the files used by the design with features system. This file controls certain default settings used in the creation of a new drawing. The second statement gives the location of the actual AutoCAD system files. The last three statements control the location and amounts of memory AutoCAD allocates for use by AutoLISP programs. The entire design with features system is written in AutoLISP. The LISPHEAP variable defines the amount of memory, in bytes, for the actual AutoLISP files. LISPSTACK defines the amount of memory, again in bytes, for the storage of AutoLISP variables. The total amount of memory set aside for LISPSTACK and LISPHEAP must not exceed 45,000 bytes.

The other file which is used to setup the design with features system is the ACAD.LSP file shown in Figure 19.

```
    (vmon)
    (prompt "\nLoading feature-based design & N.C.
         programming tools.....")
    (setq toolnm 1  ;counter for tool numbers
     geomnm 1  ;counter for geometry entities
         featnm 1  ;counter for feature entities
         filenm 10 ;counter for coding filenames
    routine 1  ;verification flag for first routine
              ;execution
);setq end
```

Figure 19: ACAD.LSP file

The ACAD.LSP file is executed automatically upon startup of the AutoCAD system. This file functions in much the same way as a DOS autoexec.bat file. The first command, '(vmon)' sets the virtual paging of memory to the "on" position. Virtual paging of memory allows a number of large AutoLISP programs to be swapped in and out of LISP memory. The end result of swapping memory is the ability to run larger AutoLISP routines than would be possible without swapping. This causes some reduction of the overall throughput of the Autocad system, however the design with features system would not run without virtual memory paging. The next command which begins with "(prompt..." tells the user of the system that the design with features system is loading upon startup of the AutoCAD system. The remaining commands are simply internal flags used for various counting purposes during the design session. The ACAD.LSP file must reside in the directory called out in the SET ACADCFG command shown in Figure 18.

Overall, the design with features system consists of approximately 150 AutoLISP and related files. These files require .5 megabytes of hard disk space. It is recommended that a minimum of 1 megabyte of disk space be available in addition to this for drawing creation and N.C. program generation.

5.2 CAM-I Test Part

At this point, it is instructive to show an example of the design with features methodology and the process planning system. This example uses the CAM-I ANC 101 test part (Hummel and Brooks, 1986) as shown in Figure 20. The ANC 101 test part was selected because it is a simple part which the system described here is fully capable of processing. This example will be explained in a step by step manner and could be used as a tutorial for individuals wanting to learn how to use the system.

The first operation to be performed in the geometric modeler environment is the selection of the raw material geometric form. This is done by pointing to an icon on the user interface labeled RAW MATR'L. Selections for raw material include: round corner square stock, prismatic stock, and bar (round) stock. Special stock configurations can be added as needed. For this example, prismatic stock is selected. Selection of the required material type and heat treatment is also made at this time.

Next, a simple exterior rectangular cube is selected to represent the outside of the part. This option can be found on the system pull down menu under the heading of O.D. FEATURES. The designer must specify the height, length, width, corner chamfers, and elevation of the feature. The feature is then placed in the drawing coordinate system. Also, a process fragment for a simple exterior cube is created by the system and placed in the feature database. The process plan fragment consists of the selection of an end mill to machine the outside perimeter of the rectangular cube.

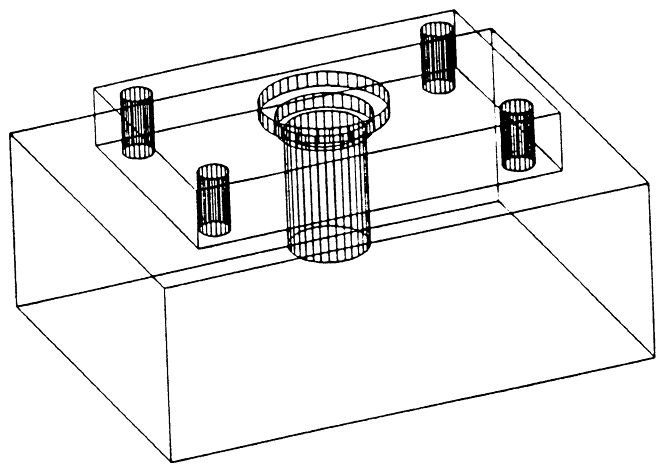

Figure 20: CAM-I ANC 101 test part

A rectangular array of simple blind holes is selected next. The designer is again queried for a number of parameters. These parameters, in this case, consist of hole diameter, effective depth, number of rows and columns, number of holes in each row and column, and elevation of the top of the holes. Following this, the system requests the location of the rectangular array of holes. The system creates a process plan fragment for the rectangular array of simple blind holes. The process plan fragment for the rectangular array of simple blind holes is to center drill then twist drill each hole.

The features in the center of the part are selected next as two simple circular pockets and a simple flat bottom hole. Process plan fragments (PPF) for each of these features are created in the same manner as above.

5.2.1 Generation of Code

All features necessary for the representation of the part have now been selected and the feature database contains six PPFs, one for each feature selected in the design. These PPFs are:

1. FEATURE: Raw stock

2. FEATURE: Exterior rectangular protrusion

   PPF: Plunge_Mill/Mill_Outside_Complete

3. FEATURE: Rectangular array of simple blind holes

   PPF: Center_Drill/Twist_Drill

4. FEATURE: Simple circular pocket

   PPF: Plunge_Mill/Circular_Interpolate_Complete

5. FEATURE: Simple circular pocket

   PPF: Plunge_Mill/Circular_Interpolate_Complete

6. FEATURE: Simple flat bottom hole

   PPF: Center_Drill/Twist_Drill/Plunge_Mill/
   Circular_Interpolate_Complete

Each of these PPFs contains the necessary information concerning geometry, material, and feature relationships for a finalized process plan for this part to be created. The "_Complete" suffix on the above PPFs signify that one tool does both the roughing and finishing cuts. Other suffixes possible in these cases are "_Rough" and "_Finish" which signify, respectively roughing and finishing tools.

To create the finalized, ordered process plan for this part, the process planning taxonomy first orders the features with the outermost, highest elevation features first; working towards the innermost, lowest elevation features last. Next, the rulebase is consulted by the process planner and a set of tools and numerical control programming statements are assembled for the machining of the part.

The process plan created for the part, as determined by the process planner module of the system, is as follows:

1. Machine the exterior rectangular protrusion using a
2 inch diameter end mill.
2. Machine the top two center cylinder features using an end mill and plunging cuts.
3. Machine the deep center cylinder by first drilling, then end milling to achieve a flat bottom hole.
4. Drill the four holes at the corner of the part.

The tools chosen by the process planner to perform the above operations are:

Tool 1 : 7/8 inch dia end mill.

Tool 2 : 11/16 inch dia end mill.

Tool 3 : # 8 center drill.

Tool 4 : 39/64 twist drill.

Tool 5 : 9/16 inch dia end mill.

Tool 6 : 2 inch dia end mill.

Tool 7 : # 7 center drill.

Tool 8 : D twist drill.

Finally, the N.C. part program created by the system, in the COMPACT II programming language, is shown below in Figure 21. To aid in understanding the COMPACT II code, the process plan fragments are shown above the segment of the COMPACT II programming statements they generated.

```
$ Geometry Statements
DRAW,PEN3
DCIR1,PT(2.0XB,1.5YB),0.5R
DSET2,PT(2.0XB,1.5YB,0.0ZB),NOMORE
DCIR3,PT(2.0XB,1.5YB),0.375R
DSET4,PT(2.0XB,1.5YB,-0.125ZB),NOMORE
DCIR5,PT(2.0XB,1.5YB),0.3125R
DSET6,PT(2.0XB,1.5YB,-0.25ZB),NOMORE
DRAW,PEN2
DLN7,0.5YB
DLN8,3.5XB
DLN9,2.5YB
DLN10,0.5XB
```

```
DPB11,LN10.S(LN7);LN9;LN8;LN7,F(LN10),NOMORE
DRAW,PEN4,PB11
DSET12,PT(0.75XB,0.75YB,0.0ZB),&
RECT(5.0LX/2EQSP,3.0LY/2EQSP),1A
DRAW,PEN5
$ Tooling Statements
$machine exterior rectangular protrusion
$               PPF:       Plunge_Mill/Mill_Outside_Complete
ATCHG,TOOL6,TLCMP6,2.0TD,180TPA,400.0FPM,0.003IPR
MOVE,OFFLN10/.1XS,OFFLN7/.1YS,NOZ
MOVE,0.1ZB
CUT,-0.5ZB,CON
CUT,PB11/CL,.003STK,-0.5ZB
CUT,PB11/CL,0.0STK,-0.5ZB
RET,STOPS,COF,OSTOP

$machine largest interior cylinder
$               PPF:       Plunge_Mill/Circular_Interpolate_Complete
ATCHG,TOOL1,TLCMP1,0.875TD,180TPA,400.0FPM,0.032IPR
MOVE,2.0XB,1.5YB,NOZ
MOVE,0.1ZB,CON
CUT,-0.125ZB,0.032IPR
ICON360,CIR1,-0.125ZB,0.005STK,S(0),CCW
ICON360,CIR1,-0.125ZB,0.0STK,S(0),CCW
CUT,0.005Z
MOVE,2.0XB,1.5YB,NOZ
RET,COF,NOX,NOZ,OSTOP,STOPS,0.0STK

$machine medium interior cylinder
$               PPF:       Plunge_Mill/Circular_Interpolate_Complete
ATCHG,TOOL2,TLCMP2,0.6875TD,180TPA,400.0FPM,0.02IPR
MOVE,2.0XB,1.5YB,NOZ
MOVE,-0.025ZB,CON
CUT,-0.25ZB,0.02IPR
ICON360,CIR3,-0.25ZB,0.005STK,S(0),CCW
ICON360,CIR3,-0.25ZB,0.0STK,S(0),CCW
CUT,0.005Z
MOVE,2.0XB,1.5YB,NOZ
RET,COF,NOX,NOZ,OSTOP,STOPS,0.0STK

$machine deep interior cylinder
```

```
$    PPF: Center_Drill/Twist_Drill/Plunge_Mill/
$                              Circular_Interpolate_Complete
ATCHG,TOOL3,TLCMP,0.6144TD,180TPA,OFFSET1,250.0FPM,0.016IPR
DRL,SET6,0.667541DP,2RDWELL,CON
RET,STOPS,COF,OSTOP

ATCHG,TOOL4,TLCMP4,0.6094TD,180TPA,250.0FPM,0.016IPR
DRL,SET6,1.0DP,CON,0.9141DEEP
RET,STOPS,COF,OSTOP

ATCHG,TOOL5,TLCMP5,0.5625TD,180TPA,400.0FPM,0.02IPR
MOVE,2.0XB,1.5YB,NOZ
MOVE,-0.15ZB,CON
MOVE,-0.75ZB,0.02IPR
ICON360,CIR5,-0.75ZB,0.005STK,S(0),CCW
CUT,2.0XB,1.5YB,NOZ,50IPM
CUT,-1.25ZB,0.02IPR
ICON360,CIR5,-1.25ZB,0.005STK,S(0),CCW
ICON360,CIR5,-1.25ZB,0.0STK,S(0),CCW
CUT,0.005Z
MOVE,2.0XB,1.5YB,NOZ
RET,COF,NOX,NOZ,OSTOP,STOPS,0.0STK

$center drill and drill rectangular array of simple holes
$    PPF: Center_Drill/Twist_Drill
DRAW,PEN2
ATCHG,TOOL7,TLCMP7,0.255TD,180TPA,250.0FPM,0.012IPR
DRL,SET12,0.32939DP,2RDWELL,CON
RET,STOPS,COF,OSTOP
DRAW,PEN3

ATCHG,TOOL8,TLCMP8,0.246TD,118TPA,250.0FPM,0.007IPR
DRL,SET12,0.5DP,CON,0.375DEEP
RET,STOPS,COF,OSTOP
END
```

Figure 21: Example COMPACT II program

This COMPACT II program may now be postprocessed into a machine readable N.C. program for the manufacture of the CAM-I ANC 101 test piece.

5.3 MBB Test Piece

The MBB test piece is most frequently used as the benchmark for proving the efficacy of CAM systems (Pratt 1988). This check is used to show that a geometrically complex part can be easily handled in terms of programming the machine tool for manufacture if the features that contribute towards the geometry are simple and straightforward. The creation of the wireframe model based on the features concept is first described and then the process selection is attempted. Validation rules that are necessary for the proper use of features are also explained to demonstrate the power of the system. Figure 22 gives the top view and the isometric view of the test piece.

A wireframe model of the MBB test piece can be obtained directly from the user interface provided by the design with features system. As the features are created, the tooling and process rule bases take over to direct the designer to perform the correct sequence of operations. The steps required include the specification of the material type of the test piece and the selection of a variety of features that contribute towards the completion of the

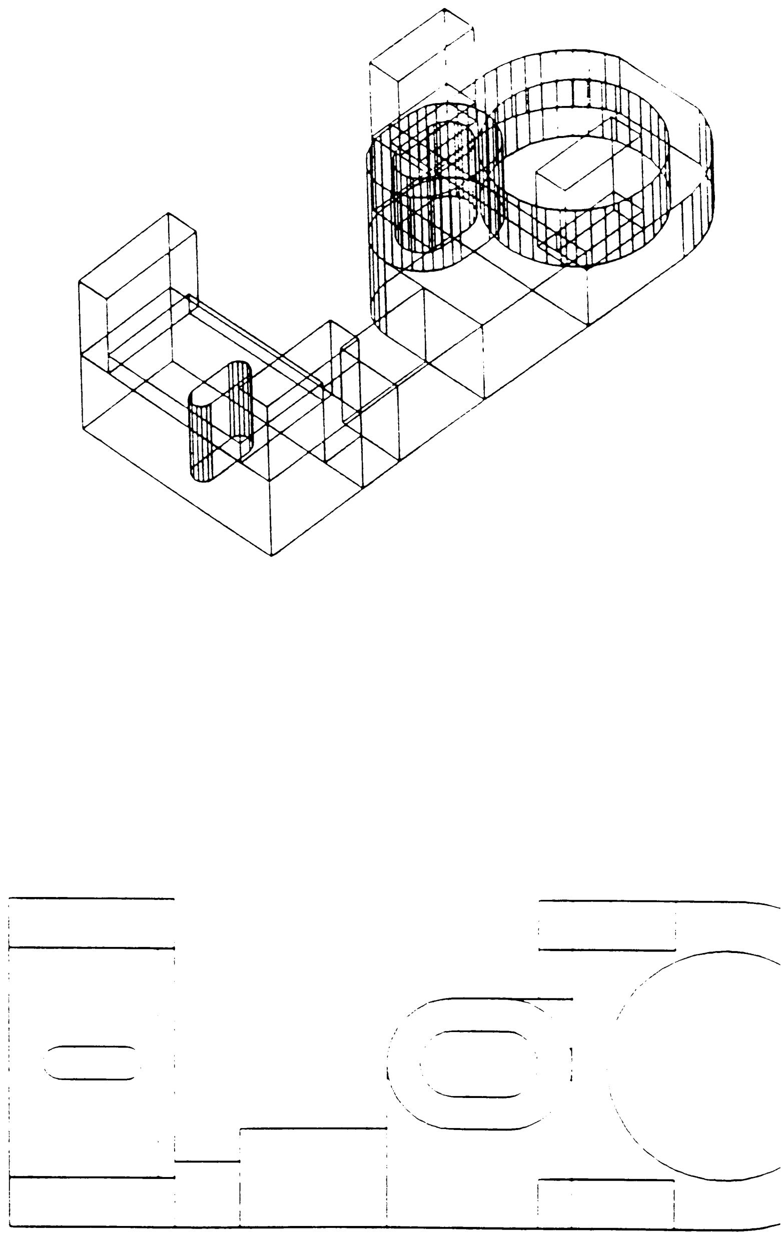

Figure 22: MBB test part

geometry. The selection of the features involves the specification of the vertices, elevations, and thicknesses of each feature. However, default values from the previous elevation and thicknesses are offered to the designer. In all, the creation of the test piece was found to take far less time than the conventional method of geometry-based design and required the specification of the following to completely generate the drawing:

1. Ten cubes of different sizes with no modification to their corners (no fillets or chamfers)
2. One through hole
3. One cube with filleted corners
4. One cube with two filleted corners
5. One cube with one filleted corner

Although the MBB test piece is complicated geometrically, the manufacturing processes necessary for its manufacture are simple. Two end mills are required for machining this piece from stock. The roughing end mill (which was selected by the system as being 1.25 inch in diameter) machines all accessible sides or faces of each feature. The roughing process leaves 0.005 inch of material on the sides of each feature. The finishing end mill (0.500 diameter) then removes the 0.005 stock left by the roughing end mill.

Specifically, during the roughing operation, the system tries to cut around each outside feature with a depth of cut equal to one-half the diameter of the end mill. The radial depth of cut is equal to one-half the radius of the end mill when possible. When a buried feed cut is necessary, the system specifies a slower feed rate and speed for the end mill because of the inferencing mechanism that is triggered once the feature information is known. The process parameters must then be selected on the basis of the facts and the rules that are in the knowledge base. Also, for cutting the inside features, a methodology of plunging and then cutting rapidly is employed. Since the plunge cut is in effect a buried feed cut, the feed rate and speed are reduced. The finishing end mill is required to achieve reasonably small corner radii at the intersections of the part features.

In generating this drawing the system must undertake a series of validity checks before beginning the process selection. These and other rules are checked at each moment a feature is created in the model. Figure 23 provides some of the possible scenarios and the following rules provide answers to solving these problems.

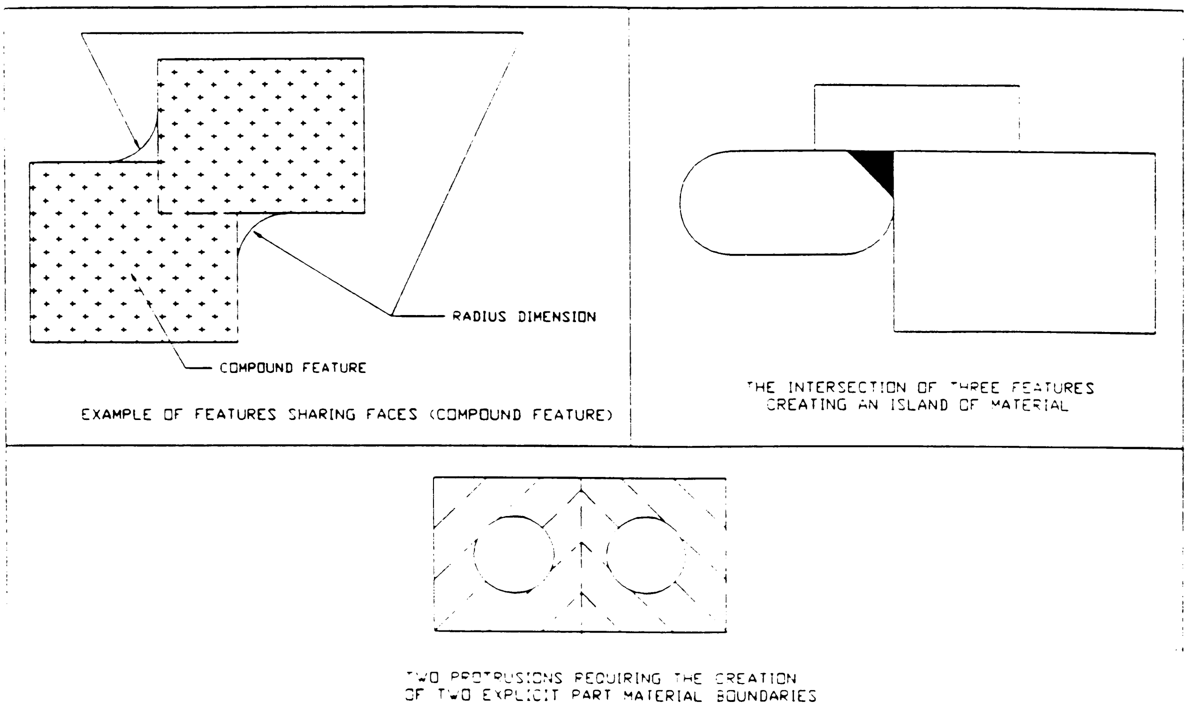

Figure 23: Feature interactions

RULE 1

If there is a feature B below a current feature A:

Machine feature A as a normal feature in the Z direction until the intersection with the feature B .

Then combine the features and define a new feature. Machine outside contour of the new feature.

RULE 2

If a feature A shares face(s) with another feature(s) B:

Combine the features A and B into a new compound feature. Also, determine the corner radii and define the part boundary.

RULE 3

When two features (or more) protrude from a single face:

Determine the minimum distance between the features. This would give the maximum size milling cutter that can be used.

Define a new raw material boundary and remove the difference.

Define raw material boundary perpendicular to the connecting points of the two protruding features.

RULE 4

If there is an interior island:

A new material boundary removal volume needs to be defined.

Thus, the problems which need to be addressed to machine the MBB test are: (i) concatenation of features, (ii) sharing of faces, (iii) protrusion of features, (iv) interior islands or pockets, and (v) feature growing.

5.4 Results and Discussion

5.4.1 Capabilities of the Integrated CAD/CAM Model

The system described herein is capable of handling many combinations of the individual features described in section 3.2 previously. There are many situations the system is not applicable to. Figure 23 shows examples of a few of these problem areas. Referring to Figure 23, the system will not handle the intersection of features (compound features), two or more protrusions from a plane or an island created by the boolean difference of three or more features.

These problems might seem to be a great detriment to the functionality of the system, however, they are not insurmountable. The algorithms for the solutions to these problems can be written in much the same manner as the algorithms for machining the individual features.

The attributes of this system as they exist at this point, are capable of creating a large number of different parts. The number of different parts is of the order of seven factorial or 5040. If the system as it exists currently were to be used in industry, it would be useful in two areas. First, many of the features created for use in this system are frequently encountered in fixture design, thus the applicability of the system to this area seems reasonable. The second area of applicability is for parts which need only to have holes drilled, counterbored or tapped.

5.4.2 Comparison of Standard CAM Package with this Model

CAM packages available today are capable of creating N.C. programs for virtually all 2 1/2 dimensional parts. Two and one-half dimensional parts are parts which are described fully with a plan view and a depth specification for each feature. A smaller number of standard CAM packages available are capable of processing fully three dimensional parts including complex sculpted surfaces defined by higher order mathematical functions. The design with features system will handle only the 2 1/2 dimensional features available from the user interface.

Assuming a part is to be manufactured which is in the domain this system encompasses, the design with features system can create a process plan and an N.C program much faster than a standard CAM package. The sequence of operations a user of a standard stand alone CAM package (the user cannot download a CAD drawing into the CAM package) must perform are as follows:

1. Create a geometrical representation of the part in the CAM system.

2. Select the correct cutting tools for the job.

3. Select the speeds and feeds for those cutting tools.

4. Create an N.C. program which describes the sequence of processes necessary for fabricating the part.

With the design with features system, the user must only create the graphical representation of the part. The selection of tooling, speeds, feeds and the actual ordering of process and the creation of the N. C. program is performed by the system. Thus, in the domain this system addresses, time savings can easily be realized.

6 CONCLUSIONS

This work successfully demonstrated that a wire frame modeling system can be used to create a design with features system aided by a simultaneous representation of the process description in a hierarchical format. This approach provides a distinct advantage over a pure feature-based system such as freeing the process planner from routine tasks, speeding up the design/manufacturing cycle and helping to ensure design for manufacturability at the earliest stage possible in the design.

The validity and usefulness of a database using frames has been shown to be a suitable method for storing geometry information in a flexible manner. The inheritance of information has been shown and a taxonomy together with a rulebase for the creation of process plans has been shown to work quite well. The generality of the database structure allows other applications to be added to the system at a later time. The knowledge gained from the creation of this system will allow other more complete applications to be developed and added to a later generation of this system, allowing a more automated approach to the complex task of design and manufacture.

Human experts in process planning and machining knowledge have contributed heavily to the creation of this system and are in agreement as to the validity of the process plans derived from the system. However, no formal testing has yet been done with the system in an industrial environment.

One of the greatest developmental problems encountered thus far is the representation of the geometric relationships between features. This difficulty is one of the drawbacks of working with a wireframe modeler because the features must be represented as vertices and faces, not as closed volumes. Thus, the problem of determining when a feature's face is shared with another feature or when a feature intersects another feature is difficult to detect reliably in all circumstances. A number of algorithms have been tested for detecting intersections, shared faces and the general problem of feature growing. However, a set of algorithms that will be successful in all applications has not yet been developed.

Another general problem that is inherent in designing with features is the incomplete set of features available to the designer. No complete set of features can be identified which will work in all circumstances. Therefore, the practicality of this system in an industrial environment is questionable. In light of this, perhaps the best way to utilize a system of this type is to accept the fact that a complete set of features will never be identified and to concentrate on the common features which are found in industry. Process planning knowledge for these common features could then be completely identified and optimized. Viable process plans for a portion of the part encompassed by the common features could then be developed simultaneously with the design phase. The result of this process will be a hybrid geometric model that will be composed of features, on one hand, and of simple lines, arcs, points, etc., on the other.

By using a hybrid model, the designer will realize a time savings by using features for some portion of the product design and the manufacturing engineer, process planner, or both would benefit by receiving a partial process plan that has the repetitive, simple features pre-planned and pre-programmed. Graphical icons could indicate to the process planner which feature have been processed, or conversely, which entities need to be addressed to complete the process plan.

6.1 Recommendation for Future Work

This work is prototype in nature and, as such, only presents concepts concerning methods for the construction of a commercial product. There are many areas of research which still need to be addressed. These include the following:

A. The treatment of the various tolerance types (ANSI and conventional) and their effect on the process selection.

B. Facilities to allow user definable features and their process fragments to be easily added to the system.

C. Treatment of different types of machine tools and processes such as lathes, lasers, punches or grinders.

D. Fixturing analysis and selection.

E. Process parameter optimization.

F. Learning from previous experience.

7 BIBLIOGRAPHY

Bond, A. H., and Chang, K. J., "Feature-Based Process Planning For Machined Parts", ASME International Computers in Engineering Conference, pp 571-576. San Francisco, Calif., August, 1988.

Cunningham, J. J., and Dixon, J. R. "Designing with Features: The Origin of Features". ASME International Computers in Engineering Conference, pp 237-243, San Francisco, Calif., August, 1988.

Cutkosky, M. R., Tannenbaum, J. M., and Muller, D. "Features in Process-Based Design." ASME International Computers in Engineering Conference, San Francisco, Calif., August, 1988.

Devgun, M. S., and Padmanabhan, S. "Feature Representation and Extraction in CAD Models For Manufacturing: Methods and Issues." Proceedings of X International Conference on Production Research, Nottingham, England, August, 1989.

Dixon J. R. "Designing with features: Building Manufacturing Knowledge into More Intelligent CAD Systems." Manufacturing International 88. Vol 1. Product & Process Design, Atlanta, Georgia, April, 1988.

Dixon, J. R., Libardi, E. C., Luby, S. C., and Vaghul, M. "Expert Systems for Mechanical design: Examples of Symbolic Representations of Design Geometries." Engineering With Computers, 2, (1987): 1-10.

Gindy, N. N. Z. "A Hierarchical Structure for Form Features." International Journal of Production Research, 27, No. 12, (1989): 2089-2103. 1989.

Henderson, M. R. Extraction of Feature Information From Three Dimensional CAD Data. Ph.D. Thesis. Purdue University, May 1984.

Hummel K. E., and Brooks S. L. "Symbolic Representation of Manufacturing Features for an Automated Process Planning System." Proceedings: Knowledge Based Expert Systems for Manufacturing, ASME Winter Annual Meeting, December, 1986.

Jared, Graham E. "Shape Features in Geometric Modeling." Solid Modeling by Computers. Proceedings General Motors Solid Modeling Symposium, Warren, Michigan, September 1983. Plenum Press, New York.

Juri, A. H., Saia, A., and De Pennington, A. "Reasoning about Machining Operations Using Feature-Based Models." International Journal of Production Research, 28, No. 1 (1990): 153-171.

Kyprianou, L. K. Shape Classification in Computer Aided Design. Ph.D. Thesis. Cambridge University, July 1980.

Lee, Y. C. and Fu, K. S., "Machine Understanding of CSG: Extraction and Unification of Manufacturing Features." IEEE Computer Graphics and Applications 7 (1987): 143-157.

Pratt M. J. "Synthesis of an Optimal Approach to Form Feature Modeling", ASME International Computers in Engineering Conference, San Francisco, Calif., August, 1988.

Pratt M. J., and Wilson, P. R. "Requirements for the Support of Form Features in a Solid Modeling System", Report No. R-85-ASPP-01, CAM-I Inc., Arlington, Texas, 1985.

Machinability Data Center, Machining Data Handbook, Volumes 1 and 2, Cincinnati, Ohio, 1980

Miner, R. H. A Method For The Representation and Manipulation of Geometric Features in a Solid Model, M.S. Thesis, Department of Mechanical Engineering, Massachusets Institute of Technology, May, 1985.

Ranyak, P. S. and Fridshal, R. "Features For Tolerancing A Solid Model", ASME Computers in Engineering Conference, San Francisco, Calif., August 1988.

Shah, J. J., and Bhatnagar, A. S. "Group Technology Classification From Feature-Based Geometric Models", Manufacturing Review, 2, No. 3, (1989): 204-213.

Shah, J. J., and Miller, D. "A Structure For Supporting Geometric Tolerances For Computer Integrated Manufacturing", Transactions of the North American Manufacturing Research Institution of SME, May, 1989.

Shah, J. J., and Rogers, M. T. "Functional Requirements and Conceptual Design of the Feature-Based Modeling System", Computer-Aided Engineering Journal, 3 (1989): 9-15.

Shah, J. J., and Wilson, P. R. "Analysis of Knowledge Abstraction, Representation and Interaction Requirements For Computer Aided Engineering", ASME International Computers In Engineering Conference, San Francisco, Calif., August, 1988.

Shah, J. J., Hsiao, D., and Robinson, R. "A Framework for Manufacturability Evaluation in a Feature-Based CAD System", Proceedings of NSF Design and Manufacturing Systems Conference, Arizona State University, Tempe, January, 1990.

Staley, S. M., Henderson, M. R., and Anderson, D. C. "Using Syntactic Pattern Recognition to Extract Feature Information From a Solid Geometric Database", Computers in Mechanical Engineering, 2, No. 3 (September 1983): 87-102.

Vaghul M., Dixon J. R., Zinsmeister G. E., Simmons M. K., "Expert Systems in a CAD Environment: Injection Molding Part Design as an Example", ASME International Computers in Engineering Conference, Boston, Mass., August, 1985.

Woo, T. C., "Feature Extraction by Volume Decomposition", Proceedings of Conference on CAD/CAM Technology in Mechanical Engineering, Massachusetts Institute of Technology. March 1982. MIT Press, Cambridge, Massachusetts.

8 ACKNOWLEDGEMENTS

The author would particularly like to thank the members of the program of study committee, namely, Mohan Devgun, Ralph Scrutton and Douglas Gemmill for their guidance and help with this project. The author would also like to thank Srikanth Padmanabhan for his numerous insightful suggestions and criticism. Finally, the author would like to thank Kari for her love and understanding during the completion of this research.

9 APPENDIX

The Appendix, which is a complete listing of all AutoLISP routines used in the construction of the afore described system, is not included here due to its extensive length. This appendix is on deposit with Mohan Devgun, Professor of Mechanical Engineering, Iowa State University and may be procured with permission of the author and Professor Devgun.