AN EXAMINATION OF FEATURE BASED MODELING AND SYSTEMS UTILIZING FEATURE TECHNIQUES

by

DENNIS JAY HANER

B.S., Kansas State University, 1986

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Approved by:

L. E. Grosh

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work represents one of the most significant accomplishments of my life. It marks the time when I, at least temporarily, have completed my formal education. I now embark on the ongoing educational experience of life. The time spent at Kansas State University has been fulfilling both educationally and personally, and I thank all of you at Kansas State.

Special thanks is offered to the Industrial Engineering Department, the College of Engineering, and the Graduate School. I would like to thank my graduate committee, Dr. L.E. Grosh, Dr. Doris Grosh, and Dr. Eddie Fowler. I would also like to thank Dr. Brad Kramer, and Jeff Silkman for their support on this project. Finally, I wish to thank the support of all my friends and family who have pushed and stood beside me throughout the years. A very special note of thanks should also be given to my fiance, Carrie Helmke.

CHAPTER 1: INTRODUCTION AND METHODOLOGY

INTRODUCTION

Much of the key work being done by manufacturing industry and academia today is an effort to improve automation levels and technology. The most severe obstacle to overcome is the problem of communications between the different computer controlled systems involved in automation. The initial phase of these automation processes is the design of a part for manufacture. Information produced by this operation is important in the effort to establish system integration. The problem may be formally stated as:

PROBLEM STATEMENT: The successful passage of manufacturing data from the designer's model (computer aided design) to the production facilities (computer aided manufacturing) is necessary to the implementation of automation ideas. Improvement is needed in modeling systems if this obstacle is to be overcome.

PROJECT GOAL: The goal of this paper is to evaluate an alternative method of modeling objects for production. This method is a subset of geometric modeling known as Feature Based Modeling. This work looks at feature based modeling from the manufacturing point of view.

The part is defined as a set of features to be removed from a raw block of some material. The design point of view looks at a part as being constructed with features. A feature may be defined as a set of faces on a geometric model which form a set of connected faces, edges, and vertices.

Chapter 2 of this work deals with providing the reader a review of what geometric modeling is, and of what it comprises. Definitions for geometric modeling, and geometric modeling systems are provided. A discussion of the role of geometric modeling in computer aided design is presented. The different methods of classifying a geometric modeling system are presented, and a discussion of some of the problems inherent in geometric modeling is presented.

Chapter 3 is concerned with providing an overview to feature based modeling. This chapter includes introductions to feature based modeling, definitions and descriptions, and a discussion of the different entities making up feature based modeling. A statement of the goal of feature based modeling and of the relationship between computer aided process planning and feature based modeling is also presented. Also included in Chapter 3 are discussions of the primary methods, data management structures, and roles of features in integrating the islands of automation.

Chapter 4 consists of descriptions of four different systems that in some way utilize feature technology.

Chapter 5 consists of conclusions drawn from this research.

METHODOLOGY

The materials used to produce this document were provided by three primary sources. The first source was the Kansas State University Libraries. Interlibrary loan was used in conjunction with the KSU libraries. The final source was instructors within the department. Many different materials were read and interpreted. While many of these were not directly useful, they did aid in providing a general picture of what the effort should involve.

CHAPTER 2:

GEOMETRIC MODELING: A REVIEW

INTRODUCTION

This chapter will provide the reader with a brief review of geometric modeling and the ideas of geometric modeling. Geometric modeling, computer geometric modeling for our purposes, can be defined as the process by which objects are represented and the shape of those objects.

This chapter will be composed of more formal definitions for geometric modeling and for geometric modeling systems. The role of geometric modeling in computer aided design, specifically the design process will be discussed, including methods and descriptions of classification techniques. Finally, some areas of shortcomings and how feature based modeling is directed to these problems will be presented.

DISCUSSION

Geometric modeling in its traditional form is illustrated by as an engineering drawing (Figure 1). These drawings provide the manufacturing personnel with information required for part production. The manufacturing engineer studies the drawing and derives the information needed for production. This information includes:

1- Feature Planning;

The part is decomposed into machinable features.

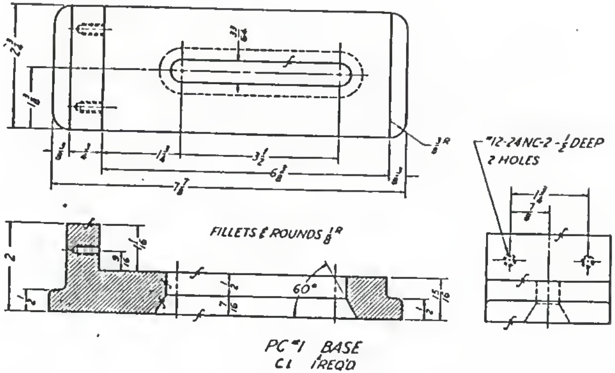
2- Cut Planning;

Each of the planned features is set with a machining sequence. This sequence is dependent upon the determination of what features must precede other features in production.

3- Tool Planning;

The required tools and fixtures for production are planned. This includes the determining of tool type based upon material types.

4- Cut Plan Optimization;



Hand clamp vise details.

Figure 1: Engineering Drawing (2)

The sequence of operations is set, considering tools and fixturing.

5- Detail and Machine Control Planning:

Speeds, feeds, number of cuts, finish, etc, are determined.

Consider the example of drilling a hole in a blank. The drawing shows the hole as two circles and two hidden lines. The engineer looks at the drawing and interprets the image in the blank to be a hole. This drawing provides the engineer with information concerning tolerances, finish, material, and specific feature characteristics (i.e. is the hole countersunk?). The trend has been away from hand drawing for the transmittal of such information. The use of computer aided design systems is becoming more frequent. The engineer must still interpret the images provided by these systems to determine process plans and computer aided manufacturing program functions.(1)

Many computer-based graphics systems have been developed since the introduction of Sutherland's system SKETCHPAD.(3) The development of SKETCHPAD and other systems has sparked the importance of modeling solids by computer, otherwise known as "Geometric Modeling".

"Geometric Modeling" focuses on the representation of shape and other spatial properties and the composition of physical objects in space. Solving a geometric modeling problem on a computer amounts to constructing a model in a representation with which the computer can work and constructing procedures to modify it interactively. Human beings are able to interpret a set of two-dimensional drawings to extract knowledge regarding three dimensions. Computational models for a computer must be based on internal representations. In a computer aided design (CAD) system, not only do human beings interpret models but so do automation systems such as numerical control (NC) tool path generator programs. The problem for conventional systems is the interpretation of the three-dimensional data. This is a problem which feature based modeling addresses directly.(3)

The above ideas lead to definitions for both geometric modeling and geometric modeling systems. Geometric Modeling may be considered to be a coherent collection of data structures used to represent physical shapes, i.e. geometric models of objects, and software modules used to construct such models. Feature based modeling is not an alternative to geometric modeling, it is merely another method of accomplishing feature based modeling.

A geometric modeling system (GMS) can be considered to be a computer-based system that provides facilities for the creation, modification, and access of object representations. Construction of current geometric modeling systems is usually due to the need for computer-aided design and computer-aided manufacturing facilities in industry. These "Geometric Modeling Systems" play a significant role in the effort to integrate CAD/CAM because of their ability to generate drawing displays or NC data for the control of manufacturing machines.(3,4)

Geometric modeling systems may be further defined as the part of a CAD system which is used for describing, editing, storing and distributing geometric data. Geometric modeling systems are characterized by the following properties:

1) Initially the object description must be entered in a form suitable for both humans and a computer. This is accomplished through the use of a definition language.

2) The information is transformed into what is known as a data representation. This allows

for the needed calculations for analysis, drawing, etc., to be made available to the user.

3) Interfaces to external programs and data bases must also be available. Translation of pre-defined entities to systems which already understand the needs of these entities is much more productive and efficient then transfer of points, lines, faces, vertices, etc., which are quite cumbersome and induce redundancy in transfer between different systems.(3,5)

Definition Language

The user must input the data description either in the form of specific data like vertex coordinates or in a format defined by a language. This language is known as the 'Definition Language'. Definition language is considered to be a procedural representation of an object. The data explicitly entered is incomplete by itself and has to be expanded by procedures into a complete data representation. Definition language consists of statements which define simple geometric primitives. The primitives are used in the geometric construction of more complex objects. The majority of the definition languages used today is based on the constructive solid geometry (CSG) representation which is the easiest to use for humans. Simple, understandable blocks used for part construction make CSG very user friendly.

The Role of Geometric Modeling in CAD

The tasks performed by a useful CAD system during the design phase of a physical object must be considered to more fully understand the idea of geometric modeling. The design phase can be broken down as follows.

1) Synthesis: Rough drafts of the object are entered into the CAD system. These descriptions will include information regarding materials, finish, and tolerances. As the object is edited so is this information. This provides a more accurate description of the object.

2) Analysis: Technical, economical, and engineering properties of the object are derived. Relationships between assembly components are evaluated. The design is then updated and improved as is feasible and possible based upon these results.

3) Manufacture Planning: Once a satisfactory image has been made available, the tasks needed for manufacture the part are determined. This phase is what is usually known as computer aided manufacturing. The first step is to determine the blank shape from which the part is to be cut. The next tasks include technical drawings, part lists, bill of materials, and NC-tapes. Mechanical objects to be assembled may also need robot programs, and quality control programs may also be generated.

4) Documentation: Documentation for the user should be generated from information provided by the system; and,

5) Redesign: The information within the CAD system should provide a basis for redesign as needed.(3)

These ideas illustrate the key principle of manufacture planning within the idea of geometric modeling and CAD. The principle is that the part should be described only once. The provided description will be used for the analyses and as a basis for the manufacture planning and other steps of the design process. The need is to provide a means so that once the initial model is provided, it can be used for the automatic generation of programs for other phases such as process planning, and NC machine programming. Systems today are unable to generate automatically fully usable NC programs in this method.(3,5,6)

Representation Schemes

The main problem of geometric modeling is to create methods for representing arbitrary physical objects in a computer. The representation scheme used in geometric modeling systems associates an object's physical model with its' representation. At the user level, the input data provided is by itself a representation. Representations inside a computer are data structures which contain geometric information about objects. The internal representation scheme used is one of the most common ways to characterize a geometric modeling system.

Geometric modeling systems are characterized as follows.

1) Domain - The set of objects that the system can represent.

2) Ambiguity - A representation is unambiguous if it corresponds to a single object.

3) Uniqueness - An object has a unique representation if it can be represented in only one way.

4) Conciseness - This refers to the elimination of redundancy in a representation. Redundancy results in the need for more storage and causes the database to be cumbersome.

5) Ease of Creation - A user should be able to easily create a representation.

6) Efficiency of Application - This is a measure of the system's ability to provide geometric data for an object for a wide variety of applications.

There are four main representation schemes for geometric modeling systems. They are: cell decomposition, constructive solid geometry, sweep representation, and boundary representation, otherwise known as wireframe. Constructive solid geometry and boundary representation will be the major points of interest because they are the best understood and most widely used methods for modeling 3-D solids.(4)

There are some general properties which may be applied to the representation schemes of any modeling system. These properties may be stated as follows.

1) Expressive Power: does the domain of the representation contain all desired objects?

2) Validity: do all admissible representations possess at least one corresponding object in the mathematical modeling space?

3) Unambiguousness: do all valid representations represent exactly one object in the modeling space?

4) Uniqueness: do all mathematical models in the domain have exactly one representation?

5) Definition Mechanism: what types of definition mechanisms may be used on objects within the scheme? How well do these definitions suit humans? How well do the definitions suit automation?

6) Conciseness: how large and cumbersome do the representations of useful objects become within the particular scheme? Is excessive redundancy present?

7) Computational Ease: what and how complex are

the computational algorithms for the scheme?

8) Closure of Operations: do the definition and manipulation operations leave the validity of the representation intact?

These characteristics serve to provide a means of evaluation for the various types of geometric modeling systems available.(4)

Constructive Solid Geometry

Constructive Solid Geometry (CSG) schemes describe a solid as a combination of given simpler objects called primitives. A solid is built by using these primitives, e.g. cubes, cylinders, etc., in varying sizes, orientations, and compositions. Shading is the primary means of 3D representation for this scheme. Many of the input facilities used for geometric modeling systems are based on the CSG scheme because of the building block approach.(4,5)

Boundary Representation

This scheme represents a solid by breaking down its boundaries into faces and representing each face by its edges and verticies (Figure 2). Two primary types of information are associated with the boundary

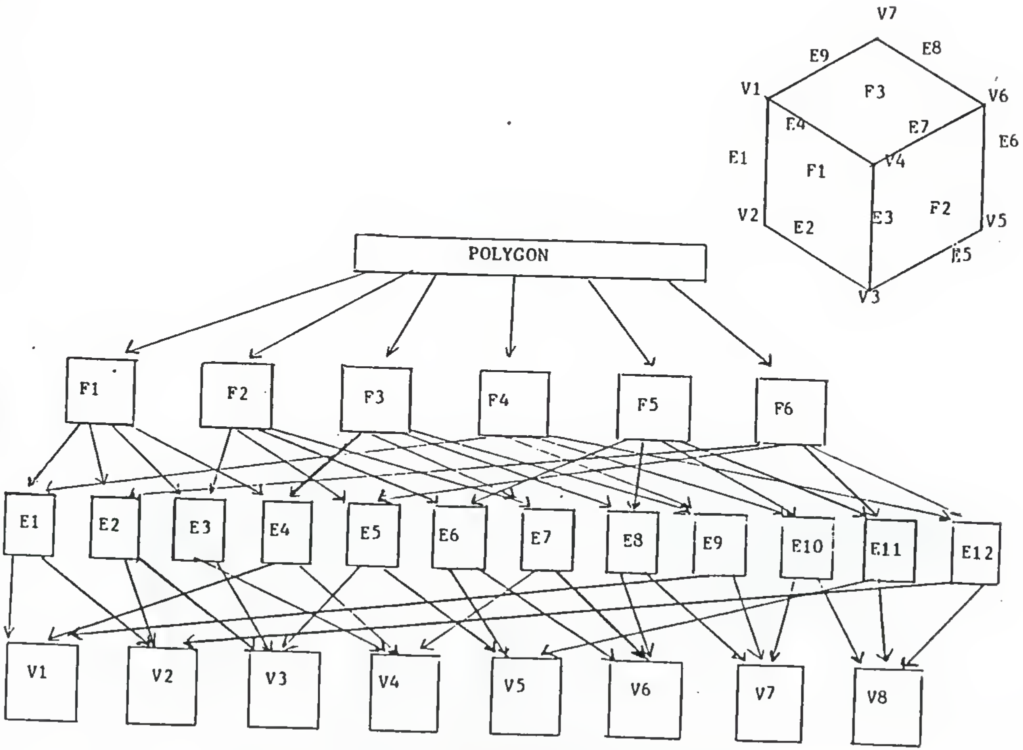


Figure 2: Boundary Representation Tree (4)

representation scheme. Geometric data defines an object's physical dimensions and locations in space, topological information describes the connections between faces, edges, and verticies. The definition of a curve can be accomplished by combining data provided by points, lines, and unit vectors. Defining the complete shape requires the combining of geometry and topology.

Boundary representation schemes are unambiguous and nonunique. They are not concise but capable of covering domains as rich as CSG schemes. The main advantage for boundary representation is that they can be compactly stored with high resolution and are very useful for graphics applications. Boundary representation is the most common internal representation used in today's geometric modeling systems.(4,5)

System Applications

The intended applications of most existing geometric modeling systems are for mechanical part design. The majority of systems use primitive blocks as input to describe solids. CAD, PADL, BUILD, and ROMULUS are geometric modeling systems that use this method of input and are used primarily for modeling the design of mechanical parts. These systems require the user to input the specifications of many parameters required to generate the curves of lines defined by the geometric functions. (Table 1) (4)

Geometric Modeling with Features

Feature based modeling is a type of geometric modeling that may be used to address the problems seen with many of today's principle systems. The key benefit we are looking for with feature based modeling is to provide design information in such a way that once the initial model is provided, it can be used to generate automatically the other phases of production. These phases include ideas such as process planning, and NC programming which includes all information needed for fabrication, not just tool path plotting.

The automatic interpretation by computers requires that the data be available in such a form as to allow 3-dimensional interpretation by the computer systems. Traditional modeling provides a 2-dimensional image which the human user is able to interpret as 3-dimensional. Feature based modeling provides design information internally to the manufacturing systems that require the information.

This is where we see the major difference, and need

Name	BUILD-1	CADD	PADL	ROMULUS	TIPS-1
Developer	Braid	–	Voelcker	–	Okino
Purpose	Research Engineering Design & NC	Commercial Drafting	Commercial Design	Commercial Design	Commercial Design & NC
Design Elements	Primitive Solids	Primitive Solids	Primitive Solids	Primitives, Sweeping	Primitive Solids

Table 1: Geometric Modeling Systems (1)

for modeling with features as opposed to modeling in the traditional sense. Many of the traditional CAM (computer aided manufacturing) ideas require human input, they cannot operate and generate the needed information without the aid of a human interpreter. These processes include operations such as turning, milling, and drilling. This is because it is very difficult for a CAM system to automatically determine the part definition data needed for part production. The feature idea is to use a predetermined feature and to attach the material, finish, tolerance, etc..., information with that feature and to have the other system components intuitively know all of the information about the production and use of each feature, group of features, and final assemblies.

The importance of the part definition data illustrates another major principle under the feature based modeling idea. As the part database progresses through the stages of production it will be constantly edited. This will allow changes in design or production sequence to be rapidly entered into the production flow. The idea is that much of this updating and alteration will be done automatically, and can be done within the structure of the predefined features of the system. This will serve to eliminate the incredible amount of redundancy for data entering and re-entering seen in today's industry.(3,4,6)

A key problem encountered is the structure of the database to be used with these modeling systems. Many database systems require the use of homogeneous records. They require:

1) fixed record lengths
2) fixed field size, type, and sequence.

This type of database structure is insufficient because the combination of all database fields is simply too large. The need is for either:

1) several different files with homogeneous records; or,
2) variable records in one file.

SUMMARY

This chapter has presented a review of geometric modeling and some of its characteristics. Geometric modeling can be defined as a coherent collection of data structures used to represent physical shapes. A geometric modeling system can be defined as a computer based system that may provide facilities for the creation, modification, and access of object relationships.

The role of geometric modeling in computer aided design was covered. The major role is in the actual design process of the part. Methods of classifying these systems were also reviewed. The primary method is according to the representation scheme that the system uses.

The last step was an evaluation of some of the problems with current geometric modeling techniques, and how modeling with features controls these problems. The two major problems are in the areas of manufacture planning, and in the high degree of redesign required. These topics shall be addressed in the following chapters.

CHAPTER 3:

FEATURE BASED MODELING: AN OVERVIEW

INTRODUCTION

Lack of communication is one of the major problems plaguing industry today. This is very apparent when considering manufacturing automation. The opening of communication lines between CAD (computer aided design) and CAM (computer aided manufacturing) systems is one of the most important requirements for successful factory automation.

Traditional CAD systems typically contain only geometric and topological information about the part to be manufactured. The manufacturing aspect of production needs the process plan to produce the part properly from the raw material envelope. Information about the part such as tolerances, finish, and material (the non-geometrical information) must be readily available to the CAD/CAM systems in order to provide the process plan.

The CAPP (computer aided process planning) systems in use today require some type of coding for part features. The process plan is generated from the interpretation of these feature specifications. The recognition and extraction of feature information from the CAD system can drive these CAPP systems. As the features are specified, consideration must be given to the order in which they will be produced. Features made up of multiple simpler features will require planning for proper production sequence. If a link is established which will be able to convert a CAD part description based on features to the manufacturing information needed by a CAM system, then a system providing a greater degree of manufacturing automation will be available.

DISCUSSION

Definition and Description

What is known as CAD (computer aided design) is usually simply computer aided drafting. Many of the CAD systems in use today use the computer aided engineering information concerning ideas such as stress calculations to produce a part image. Computer aided manufacturing often amounts to using a computer to run machine tools that are not integrated with other factory operations.

"The typical CAD model no more knows that a set of lines represent a hole than an ordinary word processor knows that a sequence of letters represent a specific sentence structure."(7) The target area of feature based modeling is to allow these systems to interpret a hole as a hole. Feature technology allows common entities such as holes, bevels, grooves, and notches to be defined so that they may be called and edited without dealing with all of the individual geometric entities, such as lines and points, of which they are composed. From these efforts has come what is known as the "design for manufacturability" concept.(7)

The "design for manufacturability" concept has gained much national attention over the last few years. The idea is to bridge the gap between the design and manufacturing stages of part production. The goal of a feature recognition scheme is to evaluate a part for manufacturability during the iterative design process. The use of a feature recognizer will tend to call attention to manufacturing concerns during the design of the part.(8)

Within the effort to "design for manufacturability" and to use feature based modeling, database structure is a key to successful system development. One of the goals of geometric modeling is to enable the construction of a control database for the information storage, retrieval, and updating of three dimensional mechanical components, assemblies and systems. Because such information is intended for a wide variety of purposes, such as documentation, drafting, engineering analysis, simulation, process planning, part programming, and automatic assembly, a geometric model must be not only complete with all necessary information, but also application independent. The idea behind using features for part definition is to attach the needed information for automatic documentation, drafting, engineering analysis, simulation, process planning, part programming, and automatic assembly directly to the features describing the part.(10)

Process planning appears to be a key in the efforts to integrate CAD/CAM. A feature as commonly defined by a process planner is based upon machine tool processes and are usually linked directly to the machines which perform those processes. A feature is defined as a set of faces forming a set of connected faces, edges, and verticies. Because the manufacturing process depends on the shapes of cavities, we further define features as recognizable volumes to be removed by machining (Figure 3). This points out the major difference between the design phase and manufacturing phase for a given part. The designer tends to look at a part as a series of features used to make-up the part. The manufacturing engineer looks at the part as the volumes/features which need to be removed from the raw stock to successfully fabricate the part. The two sides must develop a means of common definition if successful links between design and manufacturing are to be developed. The use of feature based modeling techniques is intended to provide a step in this direction.

There are two primary categories of feature definition: simple and compound. A simple feature cavity is made up of one feature, such as a hole or a

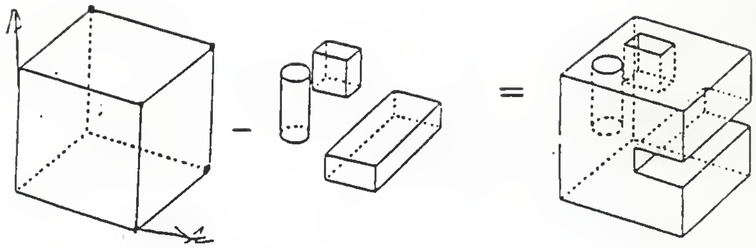


Figure 3: Feature Decomposition (8)

slot. A compound feature is comprised of a combination of simple features joined together (Figure 4). The goal is to observe all of the individual features and develop the sequence needed to produce the part. This goal is accomplished by defining the understood features as the simpler components such as points, arcs, and unit vectors that make up the feature. Based on the different characteristics of the defined features of a part, the computer aided manufacturing systems can extract the information needed to produce the different features.(9)

The goal of a feature modeling system is to provide sufficient information from the designer to drive the manufacturing process. The system should provide the necessary information to power the numerically controlled tools such as milling machines, lathes, and drill presses to make the part. A system being used by Deere & Co. is capable of just this. The computer asks the designer a series of questions about the part, in this case a hub, and responds by providing a picture, a process plan, and the NC program to make that part.(7)

CAPP (computer aided process planning) systems require information and data concerning part features. Turned parts have long been coded according to the

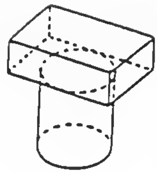


Figure 4: Compound Feature (9)

sequence of diameters, threads, holes and other features. Parts have also been classified by APT (automatically programmed tools) like statements which address geometry and structural information about the part. APT is an upper level language designed specifically for the programming of computer controlled machines. The standard procedure is that these statements are broken into "M" and "G" codes which control machine movement and settings. An example of the APT language and the corresponding "M" and "G" codes for machine control for face milling follows.

APT:

```
FMILL,102/DP(.125)/DI(0.7)/DAA(2.0,2.0,2.0)
RTO(4.0)DTO(0.0)LTO(2.0)
```

This statement is a face milling command, tool number 102, depth of cut .125, tool diameter 0.7, starting point (2.0,2.0,2.0), right to (4.0), down to (0.0), and left to (2.0).

The corresponding "M" and "G" codes are:

```
N0010  G00M33 **Positioning Mode, Spindle Start**
N0020  X015500 Y018500 **X & Y Position Commands**
N0030  G01      **Linear Interpolation Mode**
N0040  Z021000   **Z Position Command**
N0050  G00
```

```
N0060  Z-002250
N0080  X029000
N0090  G00
N0100  Y-005250
N0110  G01
N0120  X-029000
N0130  G00
N0140  Y-005250
NO150  G01
N0160  X029000
N0170  G00
N0180  Y-005250
N0190  G01
N0200  X-029000
```

It is not difficult to see the usefulness of the APT language. The "G" codes above represent machine movement control commands, the X, Y, and Z statement are positioning statements.

This is further decomposed into binary coded decimal format, which eliminates round-off problems. The process plan results from sequencing machining operations according to the features which compose the part. A CAD/CAM link should be able to convert part descriptions into feature expressions. (9)

This points out one of the problems concerning feature definitions. The manufacturing point of view prefers to model the part with consideration for features to be removed, while designers prefer to think of the part as being designed with solid features. When considering rotational parts, especially those containing threads, a problem arises of how to withdraw features from those threads. The method used is to specify a feature exclusively for threads.

Applications and Reasons for Use

Present CAD/CAM systems commonly rely on the human process planner to evaluate models of the desired part and to translate the geometry and topology into an ordered set of machinable features. This information provides NC path generation and assembly information for the part. The CAD/CAM purpose is to install a link of communication either in terms of faces, edges, and verticies or form features. The information needed to perform CAM tasks does exist in solid model images. The problem is that it is in an unusably low level. Manufacturing information is currently extracted by the human process planner, who scans the image and raises the level of information regarding part features so as to permit comprehension by CAPP programs.(9)

The problem to be addressed lies not so much in the ability of geometric modeling systems to decompose parts into features, but in the ability for the information to be passed or translated for different systems. Because of the unusably low information level for N different packages, it will take approximately N-squared interfaces. It is one of the goals of geometric modeling, and therefore feature based modeling, to provide a single description that is in a usable form for all systems. (8)

CAD/CAM has done little in past years to bridge the gap separating the many different systems. Solid modeling systems in use today are able to present very good images of the desired part, but they provide no actual link to the process to be carried out in the production of that part. It is very difficult to effectively translate part definition data needed for part production using current modeling methods.

Feature definition and modeling are part of a large issue affecting the issue of linking CAD/CAM. The goal is to be able to replace the engineering drawing as a method of passing information about the part. Features will allow information and manufacturing descriptions about the part to be passed more easily through exchange standards linking different CAD/CAM systems (IGES, PDDI, PDES, etc...).

Feature based modeling has a major role in two parts of the efforts to close the gap between CAD and CAM. The first result is that they will allow engineers to design parts by specifying readily understandable physical features. The second is that product definition databases will provide complete descriptions of components and in a form that can be transferred not only within the company, but throughout an interlinked network of organizations such as parts suppliers, service providers, and distributors.

Feature based modeling also allows for better quality control within the design phase of the project. The features are established on a basis of what is actually possible to manufacture at a given facility. If the designer specifies a feature that is not possible, then the system can provided him/her with a flag to re-think that step. Even if the feature is possible, very difficult features such as a square pocket can also be monitored for manufacturing and tooling costs. The designer will be aware of these factors while designing. Feature systems also provide a benefit in the area of component assembly. While designing the assembly, the engineer can call for the component by specifying the required features instead of looking up the required part identification numbers. One of the areas that may benefit greatly from this feature technology is group technology.

The biggest benefit of using the feature modeling idea is that it will provide a bridge for the gulf between design and manufacturing. A conventional engineering blueprint may contain a circle with two dotted lines intersecting at its center. A person must look at that drawing and deduce its meaning, and how to make that part. Feature modeling allows the image to be defined and interpreted as a hole. The computer is able to understand the concept of a hole and can automatically provide implicit instructions on how to make it. The higher level of knowledge provided by the feature allows the system to transform a design automatically into a manufacturing process, combining information on process planning as well as specification of the actual individual machining processes.

Ingersoll-Rand is currently working on the establishment of a feature based modeling system.

According to Ingersoll-Rand, such a system will allow them to halve the total time needed to set up production for a new part. It is estimated that there is a need for up 1000 different features to be defined for modeling purposes. A problem with this system is that a feature crucial to designing a part may be of little significance in actually manufacturing that part. A designer is typically more concerned about the material that exists in the finished part, its shape, and its dimensions. The manufacturer is more concerned with what material has to be removed to make the part. This is a problem spot with feature based modeling. Software is needed that could examine a computer model and identify features, much like human process planners do now. This software would also enable existing drawings to be interpreted in terms of their features. Feature recognition is still a research topic that must be resolved so that the design engineer can be free to think purely in terms of function and performance.(7)

Methods

There are two primary methods of feature based modeling under evaluation. The first method is to design the desired part from features defined and existing in a database. The second method is to evaluate a part model and extract the features from the existing image. The discussion in this work is limited to efforts of extracting features from existing models. These two methods point out the primary discrepancy between designer and manufacture needs. A designer classically thinks of a part in terms of the features which compose that part. A manufacturer/process planner thinks of a part in terms of the features that need to be removed for successful fabrication of the part.

One of the systems currently using this techniques is FEATURES. FEATURES is a system developed by Mark Henderson at Arizona State University. The system was first presented at the 1985 CAM-I annual proceedings. The FEATURES system is made up of a feature recognizer, extractor, and organizer (Figure 5). A designed manufacturable part contains various features such as holes, slots, and pockets. The recognizer identifies each of the features and the extractor separates them as volumes to be removed. The feature organizer arranges the features into a graph structure consisting of nodes for the feature and links for their relationships. The graph structure allows the part to be decomposed so that direct relationships exist between the features. The information about the stock material and all features is stored and organized in a form that can be

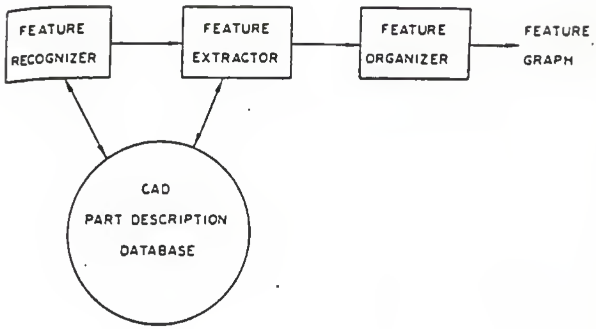


Figure 5: FEATURES System (9)

used in subsequent manufacturing processes.

The FEATURES system searches the provided part image for predefined features. These features are used to provide data relating to machining processes. These features are then recognized, extracted, and organized into a format that is appropriate for providing the required information for automated manufacturing programs.(9)

The FEATURES system addresses both simple and compound features. In order to create the feature graph the system must extract the features in such a manner as to allow machining. This system, along with a feature definer for the process planner XCUT, and a method of representing features known as the attributed adjacency graph (AAG) will be presented in Chapter 4 of this work.

Data Management

It has been stated that the key to successful development of feature based technology is the establishment of an expert system.(8) This expert system must be able to interpret not only the drawing but also all important information about manufacturing the part. Expert system structure would be applied to both the user interface and to the internal decision making process of the database.

There are three major reasons that an expert system is used for help in resolving the problem of feature based modeling. They are:

1) It attempts to simulate a human expert, the traditional CAD/CAM link.

2) It addresses the problem of knowledge representation, a very important consideration in the manipulation of semantic knowledge; and,

3) Analysis of non-analytical problems is possible.

The need for expert system technology illustrates the point that for true automation more is required than the translation of data. The system must be able to interpret and extract information about the part. The insight of the human interpreter must be present. Current requirement for passing part definition data is that several data files be operating simultaneously to define all characteristics about the part. The feature based idea is to define each feature and to have attached to that feature the needed information for production. The degree of difficulty for data translation will be simplified by such a capability. The CAD/CAM system will be able to understand that a hole is a hole, not two circles and two hidden lines.

The knowledge base of a feature based system will be made up of the CAD drawing description of the part. The information must include both geometric and topological information stored in the CAD database.

As an example, consider the general definition of a hole. The following parameters describe the hole:

1) The hole begins with an entrance face;
2) All subsequent faces of the hole share a common axis;
3) All faces of the hole are sequentially adjacent;
4) The hole terminates with a valid hole bottom. (8)

The database management systems currently available are primarily concerned with the manipulation of homogeneous, well-structured data such as numeric business data. The data from symbolic oriented/feature based modeling does not fall into this type. The basic problem is that in large, diverse manufacturing environments, little uniformity exists in the structure or content of data across different products. (11) This is a very significant problem that must be overcome for the successful use of features. Specialized CAD systems represent graphical entities most suitable for displaying graphical images. The problem is that any implied semantic data represented by these points, line segments, circles, or curve segments is lost.

The question becomes: what must the database incorporate in order that the features and the successful passage of their data is available? The database must be able to merge and communicate with different sources of data residing throughout the manufacturing process. This data is most likely scattered about in various files and databases throughout the corporation. The database used may be rule based. Many of the systems which currently rely on feature definition actually decompose the feature into simpler, more easily defined entities. This process is repeated as many times as possible to provide the database with an idea of the feature it is manipulating.

A good example of the desired database is available in a document provided by the Automated Manufacturing Research Facility of the National Bureau of Standards. Each feature used to describe the part is first defined. At this time the feature may be any one of several features that may be commonly defined by the human processor. After the feature has been identified, its type is generalized. According to the feature type definition the possible attributes are accessed in a rule based method. This allows for the development of needed product specifications (i.e. tolerances, finish, dimensions, etc...). The part is decomposed as a shell, it is then decomposed into a series of faces. The defined faces are then defined and analyzed as loops, which are then defined as a series of edges. The edges are then decomposed into vertices. These vertices are then used to define surfaces which may be decomposed into curves. The curves are defined by a series of points, lines, and unit vectors. The points are defined as absolute values in the database, as are the unit vectors. The features are then defined using the defined geometric entities. After this is accomplished the production data are attached to each respective feature for use by all operations concerned with part production (Appendix A).(12)

A second example of a feature based model database is available. This case is based on the general structure of the AMRF database (Appendix B). This example relies more on the feature idea then does the first example. The concept being addressed is that in order for a CAD system to provide sufficient information for automatic part production by the CAM system, information must be provided regarding feature characteristics as well as geometric information. The modeling system must be able to manipulate manufacturing information such as material specification, and finishes with each part feature.(1)

SUMMARY

This chapter has presented an overview of feature based modeling. A discussion has been developed concerning the definition and description of such a system. Practical applications and reasons for use of feature based modeling have been provided to show when, where, and how feature based modeling may be applied. The two basic methods of accomplishing feature based modeling are provided. They are:

1) Design with features; and,
2) Extract features from the design.

This work concentrated primarily of the idea of extracting the features from the designed parts.

Future work in this area will involve incorporating more and more features and their capabilities into production systems. Through the use of exchange standards such as PDDI, information transfer will be readily available to all parts of the production sequence.

CHAPTER 4: SYSTEMS USING FEATURE TECHNOLOGY

INTRODUCTION

This paper has been concerned with the presentation of the idea of feature based modeling. A discussion has been presented providing a background review of geometric modeling. Feature based modeling is a subset of geometric modeling.

This chapter is devoted to a more detailed explanation of feature based modeling, through the description of three modeling systems which currently use feature based part descriptions.

The first system examined will be FEATURES, which was devised and developed by Henderson at Purdue University.(4)

The second system to be examined is used in conjunction with the automated process planner XCUT. This system, developed by Hummel and Brooks, is being used at Allied Corporation, Bendix Kansas City Division.(13)

The third system using feature technology is known as attributed adjacency graph (AAG), it was developed by Joshi and Chang. (14)

DISCUSSION

FEATURES

The FEATURES system extracts and examines part features as volumes to be removed. These feature are extracted from a solid model geometry database and are arranged into a high-level graph structure for the purpose of process planning. The method consists of:

1) Searching the part description;
2) Recognizing the cavity features;
3) Extracting the features as solid volumes of material to be removed;
4) Arranging the features in a high level features graph, which is a data structure appropriate for manufacturing process planning;

The FEATURES system has been designed and developed to address some specific manufacturing problems. The problems being addressed are:

1) The CAD database does not provide sufficient manufacturing knowledge;
2) Automated part manufacturing is difficult because the CAM software is unable to extract the part description and manufacturing

information from the CAD database;

3) The expression of part features based on the given part description;

The FEATURES systems addresses these problems by attempting to simulate the human part interpreter through the use of "logic programming". Feature recognition is completed by if P1, P2, ... Pn then A. P1..Pn are conditions that if satisfied indicate that A is the resulting feature. This is what is meant by "logic programming".

The system results in a graph of features which represent volumes to be removed. The manufacturing features drawn from the CAD database are organized hierarchically according to their positions in the modeled part. This allows the semantic information associated with each of the features to be available for CAM operations.

Three major components make up the FEATURES system. The parts are a feature recognizer, a feature extractor, and a feature organizer. The feature recognizer scans the part model for each of the feature types. The feature extractor separates the features into entities and defines them as volumes to be removed. The organizer "constructs" the feature graph. Each feature is assigned a separate "node", and is linked to other features based upon their manufacturing relationships. The resulting graph formed organizes information about the stock material and the features for use in manufacturing processes.

The overall goal for the FEATURES system is to "precede the manufacturing planning by scanning the complete part to recognize features both in definition and juxtaposition and gather semantic information about the complete part before beginning the task of process planning."(9) The FEATURES system accomplishes this task by examining the part stock for patterns established by predefined generic features. This allows the feature graph to be established to meet the high level requirements of automated manufacturing programs. This allows for the separation of compound cavities into their feature components. The system is then able to determine the accessibility of each feature from the outside of the stock material.

The FEATURES system uses boundary representation (BREP) over constructive solid geometry (CSG) for its representation scheme. CSG requires that the designer not only design the part, but that he concentrate on the process planning task as well. It is not desirable to require the designer to create a process plan for each design iteration.

An example of how the feature recognizer works can be given for a cylinder. The logic programming process is:

"If a hole entrance exists
and the face adjacent to the entrance is
cylindrical
and the face is convex
and the next adjacent face is a plane
and the plane is adjacent only to the cylinder
then the entrance face, cylindrical face, and
plane comprise a cylindrical hole."(9)

In FEATURES, which uses the logic programming language PROLOG, this reads:

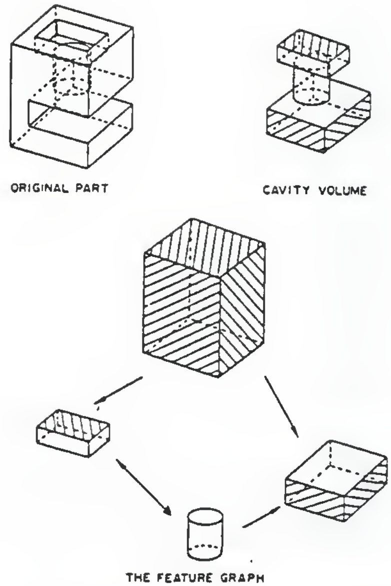
```
"cylindrical-hole(set-of-faces):-
     entrance(Face1);
     adjacent(Face1,Face2);
     cylindrical(Face2);
     convex(Face3);
     adjacent(Face2,Face3);
     not-equal(Face3,Face1);
     plane(Face3);
     adjacent-faces(Face3,Face1);
```

set-of-faces=[Face1,Face2,Face3]."(9)

Similar rules exist for the definition of other features. Using these rules, "a file containing the feature classification and its description is created, including the face list and axis. The faces and edges are listed in sequential order from entrance face to hole bottom according to their adjacency in the feature. This feature description is then used to create the feature graph."(9)

The second phase of this process is feature extraction. All of the cavities for extraction are classified and defined by the feature recognizer, which creates the previously mentioned feature graph. This step opens the door for process planning to use the feature graph (Figure 6) for sequence planning. These provide the necessary manufacturing information for all features, considering speeds, feeds, tool selection, fixturing, and cutter approach direction.

The features are subtracted from the part blank after recognition. The cavities formed by feature extraction are then checked and rechecked for more features. This process addresses the problem of compound features. The resulting set of cavities form



NOTE: ACCESS FACES ARE CROSS HATCHED

Figure 6: FEATURES Feature Graph (9)

the original compound cavity. All of the features are "tagged" with entrance face information and with axis definitions. This allows the system to determine the feature approach direction.

The feature graph (Figure 6) is constructed according to the existence and relationship of features recognized by the FEATURES system. The first layer of the graph consists of the features which contain access faces directly from the stock material, which is at the very peak of the graph. The other features are linked according to their relationship with the previously defined features. This determination of linkage occurs during the feature extraction process. The feature graph then provides the system with feature definition, adjacency, and accessibility.(9)

<u>XCUT</u>

Allied Corporation, Bendix Division Kansas City, is currently undertaking a project to develop an automatic process planner known as XCUT.(13) This system generates process plans given a feature-based part description. Bendix defines a manufacturing feature as "a structural entity whose attributes specify lower-level topology, geometry, or tolerance information."(13)

A feature is defined as "regions of a part that have some degree of manufacturing significance. Features form recurring geometric and technological patterns for which the process engineer has acquired years of manufacturing experience. (e.g.: thru-hole, slot, threaded hole, pocket, fillet, and notch)." (13) It is necessary that the feature data structure must "represent complex structural objects whose components may be either topological, geometrical, of technological in nature."(13)

XCUT utilizes object-oriented programming techniques extensively in the development of its feature representation scheme. This programming is performed using the "FLAVORS" a LISP based object-oriented language. FLAVORS presents the feature as a composite object made up of attributes and sub-objects, and the relations between these sub-objects. "The part representation structure is a graph whose nodes are instances of FLAVOR objects which denote features of that part."(13) A feature in the XCUT system represents a volume to be removed in the CAM processes. An example of this can be seen in Figure 7 and Figure 8.

The key to success for this system is its ability

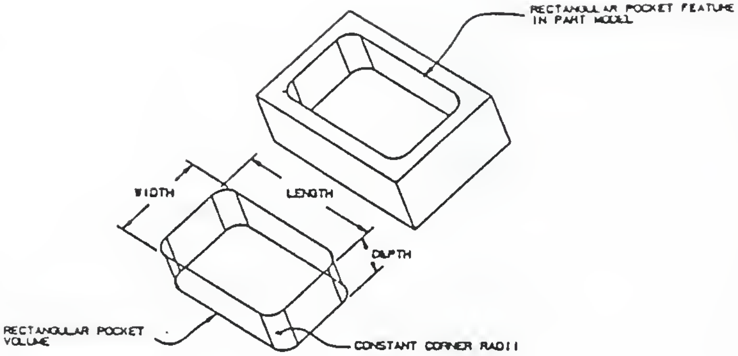


Figure 7: A Simple Rectangular Pocket (13)

```
(defflavor simple-rectangular-pocket
(width          ; width of pocket
 width-tol      ; width size tolerance
 length         ; length of pocket
 length-tol     ; length size tolerance
 depth          ; depth of pocket
 depth-tol      ; depth size tolerance
 corner-radius  ; radius of pocket corners
)
(pocket)        ; include all of pocket
                ; flavor definition.
:settable-instance-variables
:gettable-instance-variables
:initable-instance-variables)
```

Figure 8: Definition of Simple Rectangular Pocket (13)

to send messages throughout its internal structure. An example of this capability is the requesting of a list of all vertices for some feature. The message requesting the list of verticies is sent to the feature. The same message is then sent from the feature to each of the faces making up the feature. Each face then sends messages to each of its edges which return lists of vertices.

The concept of "feature access relation" is another very important issue in the XCUT scheme. "The access face is the face through which the cutting tool must pass in order to machine the feature." (13) Each of the explicit features are tagged with their access faces. They are also assigned with pointers to the features blocking those access faces (**Figure 9**). This creates what is known as a "parent-child" relationship between the features. Each of the parent features blocks at least one access face on a child feature.

Bendix has been developing a feature taxonomy for the clustering of features such as a depression, a protrusion, rectangular pocket, t-slot, notch, slab, and simple profile. A taxonomic description is "a descriptive generalization of a class of objects that sub-divides that class into sub-classes."(13) Figure 10

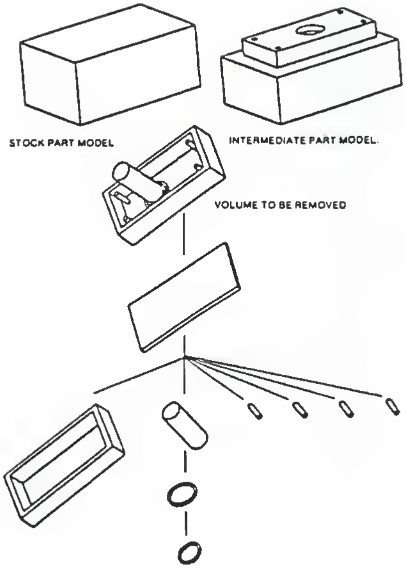


Figure 9: Feature Access Graph (13)

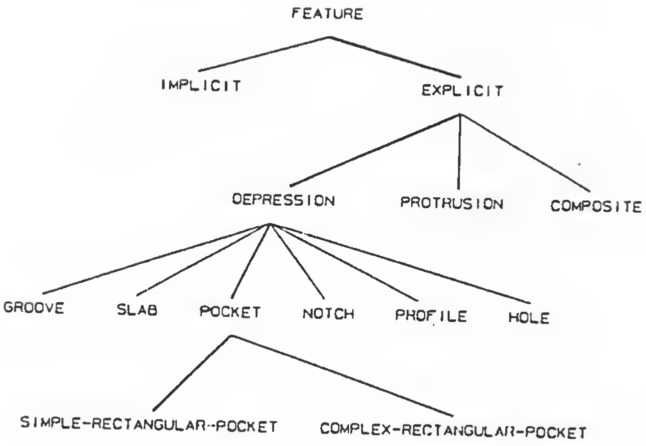


Figure 10: Example Taxonomic Breakdown (13)

gives an example of a taxonomic breakdown for a simple rectangular pocket. The "pocket" description appears as follows:

```
(defflavor pocket
     (sides      ; list of side planar faces
      corners    ; list of corner cylindrical faces
      fillets    ; list of fillet cylindrical faces
      top        ; pointer to top  entrance face
      bottom     ; pointer to bottom planar face
      islands    ; list of protrusions in pocket )
     (depression); include all of depression
                 ; flavor definition
     :settable-instance-variables
     :gettable-instance-variables
     :initable-instance-variables
```

The above inherits the characteristics of the "depression" description which appears as follows:

```
(defflavor depression
     (surface-faces         ;list of surface faces
      primary-entrance-faces;list of primary
                               entrance faces)
     (explicit)        ;include all of explicit
                       ;flavor definition.
     :settable-instance-variables
     :gettable-instance-variables
```

```
        :initable-instance-variables
```

The "depression" definition accepts properties from the "explicit" definition:

```
    (defflavor explicit
        (body)          ;pointer to feature body
        (feature)       ;include all of feature
                        ;flavor definition.
        :settable-instance-variables
        :gettable-instance-variables
        :initable-instance-variables
```

The "explicit" description inherits the properties of the "feature" description. The "feature" description appears as follows:

```
    (defflavor feature
        (tolerance    ; list of tolerance entities
        )             ; for feature.
        ()
        :settable-instance-variables
        :gettable-instance-variables
        :initable-instance-variables
```

This case shows that it is very important for any feature definitions to be extremely flexible. It would be very difficult to describe all of the features that any manufacturer would need for production. The manufacturer must have the ability to tailor these systems to meet his needs. The XCUT systems and its components are still in the developmental stage, but the key is to keep all components generic enough to allow a great deal of flexibility.(13)

Attributed Adjacency Graph

The Attributed Adjacency Graph (AAG) is a system designed for the recognition of machined features from a 3-dimensional boundary representation of a solid. The AAG approach is based upon the representation scheme used. The feature definition for the AAG are based on the representation scheme, and the inference procedures used.(14)

A boundary representation (BREP) scheme is used as the part representation scheme for the AAG. The boundary model of the object is defined B={F,E,V} where:

F={set of faces}

E={set of edges}

V={set of verticies}.

The BREP alone does not provide sufficient information for feature recognition; the system must also have additional information concerning face adjacencies and relationships. The low-level BREP, with its additional information is "converted into features that form a higher level structural entity whose primitives are the elements of a BREP."(14)

The AAG (**Figure 11**) "can be defined as a graph G=(N,A,T), where N=nodes, A=arcs, T=attributes to arcs defined in A, such that:

For all f in F, there exists a unique node n in N.

For every e in E, there exists a unique a in A, connecting n_i and n_j, corresponding to f_i and f_j, which share e.

Every a in A, has an attribute t, t=0 if the faces sharing the edge form a concave angle, t=1 if they form a convex angle." (14)

The exact definition of the feature is necessary for recognition of the feature to occur. The degree of detail needed determines the depth of the AAG hierarchial tree (**Figure 12**). This tree plays a key part in the reduction of the amount of time required for feature recognition.

AAG uses feature rules based upon the recognition of the general feature type. Specific sub-classes of these rules are obtained through the use of more

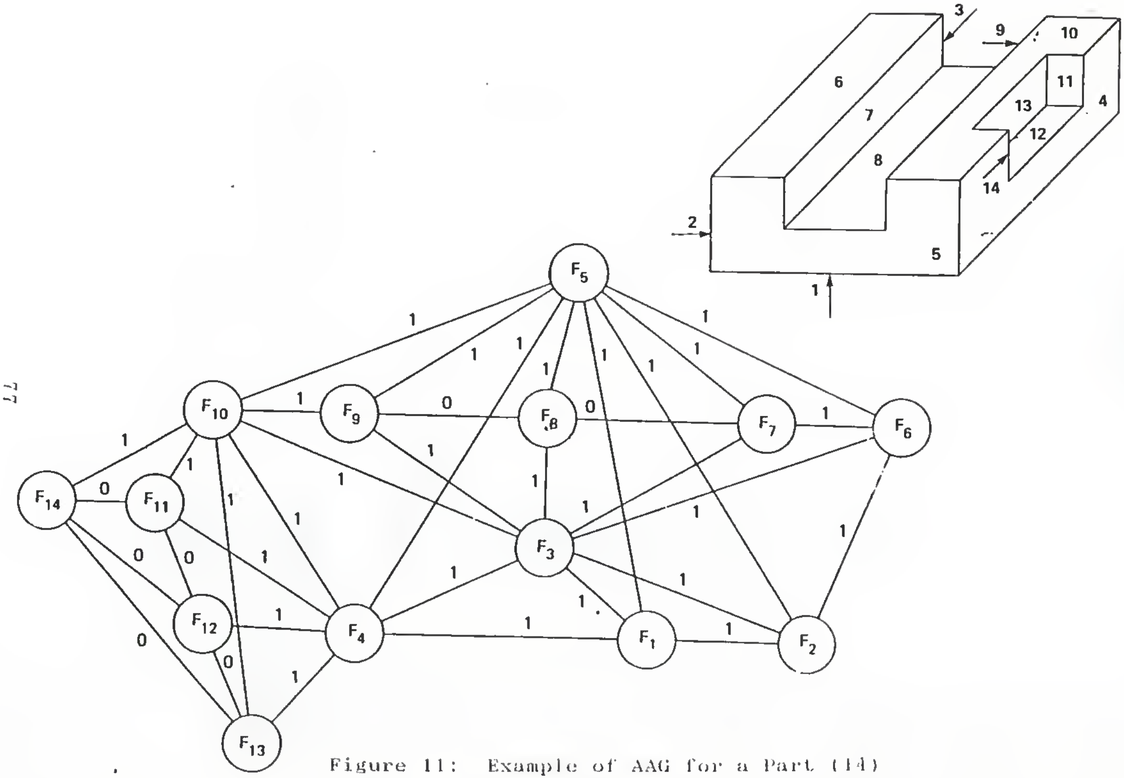


Figure 11: Example of AAG for a Part (14)

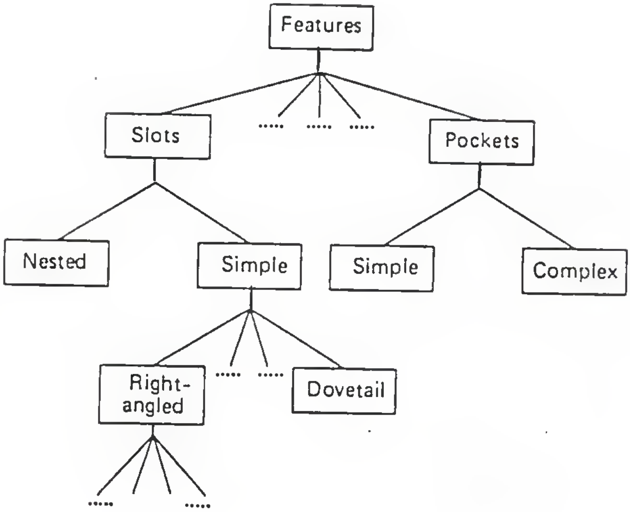


Figure 12: Hierarchial Organization of Features (14)

specific rules. The recognition rules are based upon AAG properties unique to one another. "To define a pocket, the AAG uses the following general rules:

- the graph is cyclic
- exactly one node n with number of incident 0 arcs is equal to the total number of nodes minus one.
- all other nodes have degree = 3
- the number of 0 arcs is greater than the number of arcs (after deleting node n)."(14)

All features are defined using such general rules. The rules must be unique for each feature defined, if not then an additional test is required.

The feature recognition procedure in AAG works on the premise that each feature will make up a sub-graph of the entire graph. Recognition is dependent upon the identification of subgraphs that correspond to the features. A heuristic method is used to identify the different components that form the features. This operation is based on "a fact that is adjacent to all its neighboring faces with a convex angle does not form part of a feature."(14) The sub-graphs are analyzed and defined by the Recognizer to determine the feature types (Figure 13).(14)

```
Procedure Recognize_Features
   create AAG
   delete nodes (and the incident arcs at the nodes) such that
      for each node deleted, all incident arcs have attribute '1'
   form components of the graph
   for each component
      Call Recognizer
      if recognized then
         return (feature_type, comprising_faces)
      else
         call Split_Edges
            form subcomponents of the graph
            for each subcomponent
               Call Recognizer
               if recognized then
                  return (feature_type, comprising_faces)
               else
                  call Split_Nodes
                  form sub_components
                  for each sub_subcomponent
                     Call Recognizer
                     if recognized then
                        return (feature_type, comprising_faces)
                     else
                        Call Virtual_Pocket
                        if recognized
                           return (feature_type, comprising_faces)
                        else
                           return (Not_recognized)
                        endif
                     endif
                  next sub_subcomponent
               endif
            next subcomponent
         endif
   next component
   Call Join_Features
End
```

Figure 13: Recognize Features Procedure (14)

There are two types of interacting features that the AAG will address. Type 1 features are those which intersect such that they have only common edges between them. This condition is illustrated in Figure 14. The sets of {F1,F2} and {F3,F4} form split faces. The feature defined by {F10,F11,F12} make up the split.

During the recognition phase the split faces may need to be merged into a single node for the process to be successful. If this is the case, then the faces must:

- have the same equation
- have one face to which both are adjacent (F11 in this case).

The next step is that the feature sub-graph must be decomposed into primitive elements. This is accomplished by a "procedure known as 'split_edges' which separates the features as follows:

- identify all nodes with the number of 1 incident arcs > 1
- delete all "1 arcs" that emanate from such nodes
- form components of the graph
- add back "1 arcs" deleted within each component."(14)

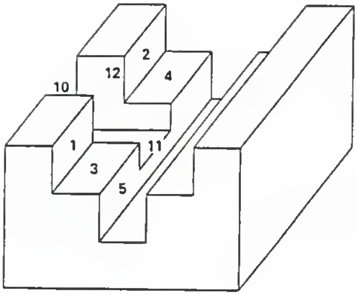


Figure 14: Type 1 Feature Interaction (14)

To insure that features are not improperly separated, procedure "Join_Features" is utilized. This procedure evaluates feature pairs to determine if they can be merged into a single feature. Local adjacency and split face information is used to determine if the features can be merged. "The process merges features by merging the list of faces that compromise the features, reconstructing the adjacencies for the merged list, and applies the Recognizer to determine if the merged features can be recognized as a single feature." (14)

The type 2 (**Figure 15**) feature addressed are "features which intersect such that they share a common face, and interaction between the features splits a face of the feature."(14) The procedure "split_nodes" performs this operation. With respect to **Figure 15**, the procedure works as follows:

- F3 and F4 are split faces and are assigned set {A}
- F2 is adjacent to {A} with 0 arcs and is assigned {B}
- F5 and F6 are the nodes creating the split. They are assigned {C}
- F2 is the node to be split
- {F5,F6} = {D}

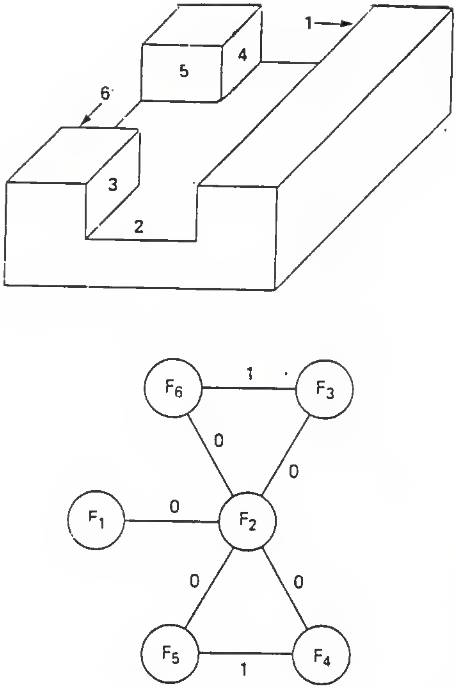


Figure 15: Type 2 Feature Interaction (14)

- {D} is set as F2' if adjacent to F2, and the 1 arcs between F6 and F3 and between F5 and F4 are deleted.

- F3 and F4 become one node. This forms surface 1 (S1) and surface 2 (S2).

  S1 = F1,F2,F3,F4

  S2 = F5,F6,F2'

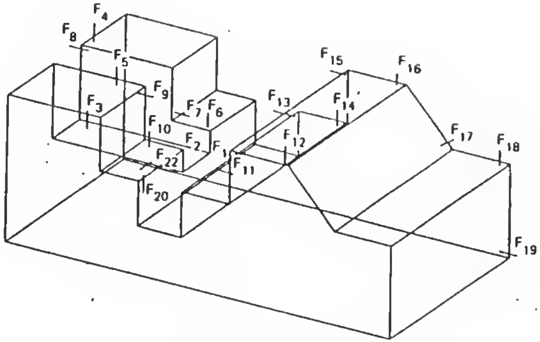
The AAG also contains a special case for addressing virtual pockets. A virtual pocket is made up of several smaller pockets. Feature interactions occur which cause the loss of some information critical to the recognition process. This occurs when several features combine to form to make unique classification difficult. Features which fall into this category are known as virtual pockets.

The virtual pockets are identified and processed after all individual and intersecting features have been identified. All remaining pockets are checked to see if they can be defined as virtual pockets, if not they are defined as individual pockets.

The AAG recognizer is interfaced with a Romulus solid modeller, which stores the image in the form of a boundary representation. The feature recognizer is programmed in Fortran 77 and used on a Sun 3/50 workstation. This arrangement was able to decompose completely Figure 16a in 2.5 seconds and Figure 16b in 2.0 seconds.

The use of the heuristic system is essential to avoid exponential process time growth with the addition of features to the part. The AAG provides a recognition of a superior number of feature types.

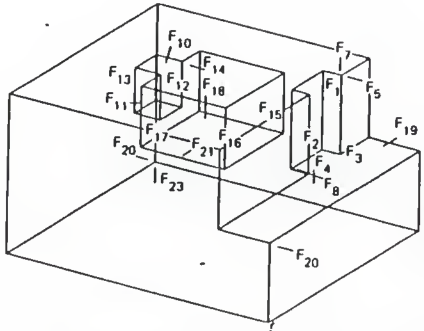
The application of this type of system "is seen in automated process planning, where information is used to determine tool approach, machining sequences, and the generation of tool paths."(14) Design for manufacturability and part classification also stand to benefit by the implementation of this technology. (14)



Feature 1
Feature type: Step
Comprising faces: 18,17

Feature 2
Feature type: Slot
Comprising faces: 15,14,13,12,11,10,9,7,6

Feature 3
Feature type: Slot
Comprising faces: 3,2,1



Feature 1
Feature type: Pocket
Comprising faces: 18,17,16,15,14

Feature 2
Feature type: Blind slot
Comprising faces: 13,12,11,10

Feature 3
Feature type: Step
Comprising faces: 5,4,8

Feature 4
Feature type: Blind slot
Comprising faces: 3,2,1,4

Figure 16: Test Part (a) and Test Part (b)

SUMMARY

The primary focus of this chapter has been to present examples of systems which utilize feature based modeling techniques. Three different systems have been examined and discussed.

The first system examined was FEATURES. This system was developed by Henderson at Purdue University. The goal of this system is to extract recognizable cavity features from the part description, and to organize the features into a higher level feature graph appropriate for process planning.

The second system examined was a feature recognizer to be used in conjunction with the process planner XCUT. This system is being developed by Brooks at Allied Bendix, Kansas City Division. The goal of the system is to provide to XCUT the part information in the form of a collection of features for manufacturing and process planning operations.

The third system evaluated is the concept of the attributed adjacency graph (AAG). This system has been developed so that the CAD part representation can be used to provide needed manufacturing information.

CHAPTER 5: CONCLUDING REMARKS

Feature based modeling is one of many areas being developed in the effort towards manufacturing automation. Industrial facilities today are occupied by islands of automation. Feature based modeling represents the attempt to bridge these islands.

The ability to provide computer aided manufacturing systems with pertinent and accurate data is paramount. Feature based modeling is a computer aided design technique which attempts to provide this data. The ability to provide manufacturing data (i.e., tolerance, finish, etc...) along with the design data is very important to closing the gap between computer aided design and computer aided manufacturing.

The development and further implementation of feature based modeling is dependent upon its users. A major obstacle to be overcome exists between the design process and the manufacturing process. The designer tends to think in terms of building the desired part with the defined features. The manufacturer thinks in terms of a blank piece of stock from which volumes defined by features are to be removed. The future success of efforts to integrate computer aided design and computer aided manufacturing will be dependent upon designers and manufacturers resolving this issue.

Designing for manufacturing is the key concept. The part to be produced should be designed with the manufacturing process in mind. The designer must take into account the needs of the manufacturer. Feature based modeling may provide a solution to this problem. Feature modeling allows the manufacturing information to be part of the specified feature. The designer may still perform the design phase by constructing with features. The final step of the design phase would then be to specify the raw stock needed to produce the part. This blank could then be used to extract the features for the manufacturing process. The part definition data with the features will allow the computer aided manufacturing systems to perform the production operations.

Feature based modeling allows for the definition and transfer of part definition data. This is important for the successful development of automated manufacturing and production systems. Feature based modeling is still in its very early development stages. One of the major needs is the definition of more features. Another need is for design and manufacturing to find a common ground of cooperation to facilitate development. The feature recognition process may be the key to resolving this issue. The process may provide a complement of the design features with respect to the raw material envelope. The recognition process would convert the positive features used for design and the raw material envelope to the negative features defined as volumes to be removed. If this problem can be overcome, feature based modeling will play a major role in the advancement of automation.

The "total-factory" concept represents the future of factory automation. The idea is that part production be totally automated, including supporting services such as shipping, receiving, and ordering of raw materials to meet customer orders. Feature based modeling may play a very major role in the realization of this concept. Non-manufacturing information, such as unit cost, could be transferred with the features. This would permit the supporting services for the manufacturing processes to provide input to the production system.

GLOSSARY

CAD - Computer Aided Design

CAM - Computer Aided Manufacturing

Feature - Geometric entity such as a hole, a grove, or a pocket

System - A grouping of devices intended to perform some task as a unit.

NC - Numerically controlled machine tools

GMS - Geometric Modeling System

CSG - Constructive Solid Geometry

APT - Automatically Programmed Tools computer language

Exchange Standards - Standards of communication between manufacturing components.

REFERENCES

1) Jeff Silkman, "Development of an Integrated CAD/CAM System for Automated NC Program Generation", M.S. Thesis, Industrial Engineering Department, Kansas State University, May 1988

2) Warren J. Luzadder, "Fundamentals of Engineering Drawing", Purdue Univesity, September 1947.

3) Roy Tong, "Geometric Modeling by Computer and Implementation of TIPS-1", Electrical Engineering Dept. Univ. of Washington, December 1, 1981

4) Martti Maetyl,"Geometric Modeling of Solid Objects", AE Helsinki Univ. of Technology, ESPOO (Finland), Lab of Information Processing Science, 1981

5) Bjoerke, O., "Geometric Modeling as an Integral Part of Computer Aided Design and Manufacturing Systems", Norges Tekniske Hoegskole, Trondheim. Verkstedteknisk Lab., July 1983

6) Garth Thompson,<u>Topics of Automated Factory Concept</u>, Presentation by Mechanical Engineering Department, Kansas State University, March 1988

(7) Brody, H., "CAD Meets CAM," <u>High Technology</u>,,May 1987, pp. 12-18

(8) Henderson, M., "Feature Recognition in Geometric Modeling," CAM-I Annual Proceedings, 1985, pp. 5-1 to 5-11.

(9) Henderson, M. and Anderson, D., "Computer Recognition and Extraction of Form Features: A CAD/CAM Link," Computers in Industry, 1984, pp. 329-339.

(10) Woo, T.C.,"Interfacing Solid Modeling to CAD and CAM: Data Structures and Algorithms for Decomposing a Solid," Computer, December 1984, pp. 44-49.

(11) Cammarata, S.J. and Melkanoff, M., "An Information Dictionary for Managing CAD/CAM Databases," Database Programming and Design,March 1988, pp. 26-35.

(12) Hopp, T., "AMRF Database Report Format Part Model," National Bureau of Standards Automation Manufacturing Research Facility, October 22, 1986.

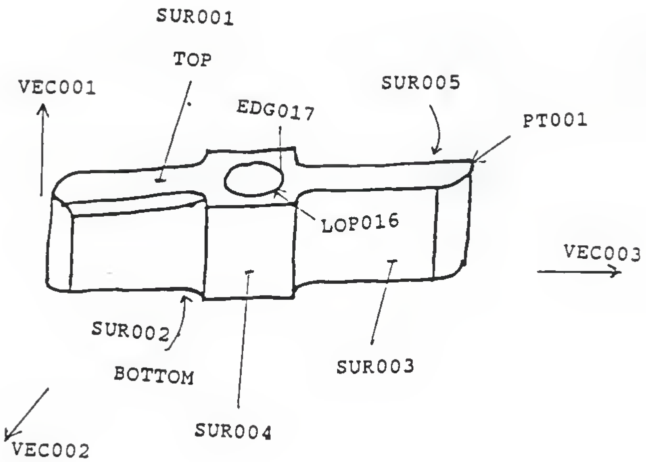
(13) Hummel, K. and Brooks, S., "Symbolic Representation of Manufacturing Features for an Automated Process Planning System", Allied Corporation, Bendix Kansas City Division, Kansas City, Missouri.

(14) Joshi, S. and Chang, T., "Graph-Based Heuristics for Recognition of Machined Features from a 3D Solid Model", <u>Computer Aided Design</u>, vol. 20, number 2, March 1988, Pages 59-66.

APPENDICES

APPENDIX A

DOG



SUR001
TOP
VEC001
SUR005
EDG017
PT001
LOP016
VEC003
SUR002
BOTTOM
SUR003
SUR004
VEC002

```
/PART_MODEL
/HEADER
  PART_NAME = 'DOG' .
/END_HEADER
/TOPOLOGY
/SHELLS
        SHELL001 ; TOP    , BOTTOM , FAC003 , FAC004 , FAC005 ,
                   FAC006 , FAC008 , FAC009 , FAC011 , FAC012 ,
                   FAC013 , FAC014 , FAC015 , FAC016 , FAC017 ,
                   FAC018 , FAC019 , FAC020 , FAC021 .
/END_SHELLS
/FACES
       TOP    ; LOP001 , LOP016 ; SUR001 + .
       BOTTOM ; LOP002 , LOP017 ; SUR002 - .
       FAC003 ; LOP003 ; SUR003 + .
       FAC004 ; LOP004 ; SUR004 + .
       FAC005 ; LOP005 ; SUR005 - .
       FAC006 ; LOP006 ; SUR006 - .
       FAC008 ; LOP008 ; SUR008 + .                        FACES
       FAC009 ; LOP009 ; SUR009 + .
       FAC011 ; LOP011 ; SUR011 + .
       FAC012 ; LOP012 ; SUR012 + .
       FAC013 ; LOP013 , LOP018 ; SUR013 + .
       FAC014 ; LOP014 ; SUR003 + .
       FAC015 ; LOP015 ; SUR005 - .
       FAC016 ; LOP019 ; SUR016 + .
       FAC017 ; LOP020 ; SUR017 + .
       FAC018 ; LOP021 ; SUR018 - .
       FAC019 ; LOP022 ; SUR019 + .
       FAC020 ; LOP023 ; SUR020 + .
       FAC021 ; LOP024 ; SUR021 - .
/END_FACES
                                                      LOOPS
/LOOPS
      LOP001 ;  EDG006 - , EDG008 - , EDG053 + , EDG069 + ,
                EDG057 + , EDG058 - , EDG059 - , EDG014 + ,
                EDG016 - , EDG054 + , EDG075 + , EDG070 - ,
                EDG071 - , EDG078 - ,
      LOP002 ;  EDG025 + , EDG024 + , EDG022 + , EDG076 - ,
                EDG035 + , EDG033 - , EDG032 + , EDG030 - ,
                EDG055 + , EDG027 + .
      LOP003 ;  EDG053 + , EDG052 - , EDG055 - , EDG051 + .
      LOP004 ;  EDG006 + , EDG042 - , EDG025 - , EDG041 + .
      LOP005 ;  EDG030 + , EDG067 + , EDG061 - , EDG064 - .
      LOP006 ;  EDG014 - , EDG048 - , EDG033 + , EDG047 + .
      LOP008 ;  EDG008 + , EDG041 - , EDG027 - , EDG052 + .
      LOP009 ;  EDG078 + , EDG072 + , EDG044 - , EDG024 - ,
      LOP011 ;  EDG016 + , EDG047 - , EDG035 - , EDG046 + .
      LOP012 ;  EDG032 - , EDG048 + , EDG059 + , EDG060 + ,
      LOP013 ;  EDG017 + ,
      LOP014 ;  EDG073 + , EDG043 - , EDG022 - , EDG044 + .
      LOP015 ;  EDG054 - , EDG046 - , EDG056 + , EDG045 + .
      LOP016 ;  EDG017 - .
      LOP017 ;  EDG036 + .
      LOP018 ;  EDG036 - .
```

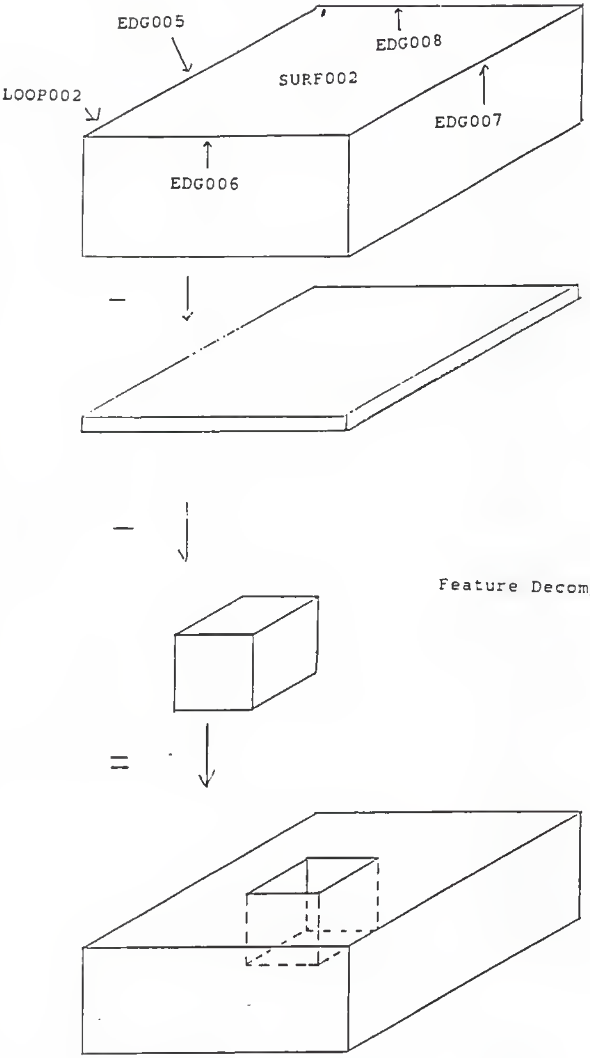
```
        LOP019 ;  EDG061 + , EDG066 - , EDG057 - , EDG062 + .
        LOP020 ;  EDG060 - , EDG058 + , EDG066 + .
        LOP021 ;  EDG051 - , EDG063 + , EDG064 + , EDG062 - ,
        LOP022 ;  EDG073 - , EDG077 + , EDG070 + , EDG074 + .
        LOP023 ;  EDG071 + , EDG077 - , EDG072 - .
        LOP024 ;  EDG076 + , EDG043 + , EDG074 - , EDG075 - ,
/END_LOOPS
/EDGES
       EDG006 ; VTX009 , VTX006 ; CRV006 + .
       EDG008 ; VTX010 , VTX009 ; CRV008 + .
       EDG014 ; VTX018 , VTX021 ; CRV014 + .
       EDG016 ; VTX022 , VTX021 ; CRV016 + .
       EDG017 ;  ;  CRV017 + .
       EDG022 ; VTX034 , VTX032 ; CRV022 + .
       EDG024 ; VTX035 , VTX034 ; CRV024 + .
       EDG025 ; VTX038 , VTX035 ; CRV025 + .
       EDG027 ; VTX039 , VTX038 ; CRV027 + .                EDGES
       EDG030 ; VTX044 , VTX046 ; CRV030 + .
       EDG032 ; VTX047 , VTX046 ; CRV032 + .
       EDG033 ; VTX047 , VTX050 ; CRV033 + .
       EDG035 ; VTX051 , VTX050 ; CRV035 + .
       EDG036 ;  ;  CRV036 + .
       EDG041 ; VTX038 , VTX009 ; CRV041 + .
       EDG042 ; VTX035 , VTX006 ; CRV042 + .
       EDG043 ; VTX032 , VTX063 ; CRV043 + .
       EDG044 ; VTX034 , VTX065 ; CRV044 + .
       EDG045 ; VTX030 , VTX001 ; CRV045 + .
       EDG046 ; VTX051 , VTX022 ; CRV046 + .
       EDG047 ; VTX050 , VTX021 ; CRV047 + .
       EDG048 ; VTX047 , VTX018 ; CRV048 + .
       EDG051 ; VTX042 , VTX013 ; CRV051 + .
       EDG052 ; VTX039 , VTX010 ; CRV052 + .
       EDG053 ; VTX013 , VTX010 ; CRV003 + .
       EDG054 ; VTX022 , VTX001 ; CRV011 + .
       EDG055 ; VTX042 , VTX039 ; CRV022 + .
       EDG056 ; VTX051 , VTX030 ; CRV030 + .
         EDG057 ; VTX056 , VTX058 ; CRV057 + .
         EDG058 ; VTX058 , VTX060 ; CRV058 + .
         EDG059 ; VTX018 , VTX060 ; CRV059 + .
         EDG060 ; VTX060 , VTX064 ; CRV060 + .
         EDG061 ; VTX062 , VTX064 ; CRV061 + .
         EDG062 ; VTX056 , VTX062 ; CRV062 + .
         EDG063 ; VTX044 , VTX042 ; CRV063 + .
         EDG064 ; VTX044 , VTX062 ; CRV064 + .
         EDG066 ; VTX058 , VTX064 ; CRV066 + .
         EDG067 ; VTX046 , VTX064 ; CRV067 + .
         EDG069 ; VTX056 , VTX013 ; CRV069 + .
         EDG070 ; VTX059 , VTX055 ; CRV070 + .
         EDG071 ; VTX061 , VTX059 ; CRV071 + .
         EDG072 ; VTX061 , VTX065 ; CRV072 + .
         EDG073 ; VTX065 , VTX063 ; CRV073 + .
         EDG074 ; VTX063 , VTX055 ; CRV074 + .
         EDG075 ; VTX001 , VTX055 ; CRV075 + .
         EDG076 ; VTX030 , VTX032 ; CRV076 + .
         EDG077 ; VTX065 , VTX059 ; CRV077 + .
         EDG078 ; VTX006 , VTX061 ; CRV078 + .
```

```
/END_EDGES
/VERTICES
        VTX001 ;  PT001 .
        VTX006 ;  PT006 .
        VTX009 ;  PT009 .
        VTX010 ;  PT010 .        VERTICES
        VTX013 ;  PT013 .
        VTX018 ;  PT018 .
        VTX021 ;  PT021 .
        VTX022 ;  PT022 .
        VTX030 ;  PT030 .
        VTX032 ;  PT032 .
        VTX034 ;  PT034 .
        VTX035 ;  PT035 .
        VTX038 ;  PT038 .
        VTX039 ;  PT039 .
        VTX042 ;  PT042 .
        VTX044 ;  PT044 .
        VTX046 ;  PT046 .
        VTX047 ;  PT047 .
        VTX050 ;  PT050 .
        VTX051 ;  PT051 .
        VTX055 ;  PT055 .
        VTX056 ;  PT056 .
        VTX058 ;  PT058 .
        VTX059 ;  PT059 .
        VTX060 ;  PT060 .
        VTX061 ;  PT061 .
        VTX062 ;  PT062 .
        VTX063 ;  PT063 .
        VTX064 ;  PT064 .
        VTX065 ;  PT065 .
/END_VERTICES
/END_TOPOLOGY
/GEOMETRY
/SURFACES                                            SURFACES
       SUR001 ; PLANE ; VEC001 ;  0.96875 .
       SUR002 ; PLANE ; VEC001 ;  0.00000 .
       SUR003 ; PLANE ; VEC003 ;  0.78125 .
       SUR004 ; PLANE ; VEC003 ;  1.00000 .
       SUR005 ; PLANE ; VEC003 ;  0.21875 .
       SUR006 ; PLANE ; VEC003 ;  0.00000 .
       SUR008 ; CYLINDER ;  PT041 ; VEC001 ;  0.37500 .
       SUR009 ; CYLINDER ;  PT037 ; VEC001 ;  0.37500 .
       SUR011 ; CYLINDER ;  PT053 ; VEC001 ;  0.37500 .
       SUR012 ; CYLINDER ;  PT049 ; VEC001 ;  0.37500 .
       SUR013 ; CYLINDER ;  PT057 ; VEC001 ;  0.31250 .
          SUR016 ; PLANE ; VEC008 ;  0.46404 .
          SUR017 ; CONE ; PT068 ; VEC001 ; 0.70711 .
          SUR018 ; CYLINDER ;  PT045 ; VEC001 ;  0.28125 .
          SUR019 ; PLANE ; VEC013 ;  1.17115 .
        . SUR020 ; CONE ; PT069 ; VEC001 ; 0.70711 .
          SUR021 ; CYLINDER ;  PT033 ; VEC001 ;  0.28125 .
/END_SURFACES
```

```
/CURVES
        CRV003 ; LINE ; PT013 ; VEC002 .
        CRV006 ; LINE ;  PT009 ; VEC002 .
        CRV008 ; CIRCLE ; PT012 ; VEC001 ; PT009 .
        CRV011 ; LINE ;  PT022 ; VEC002 .
        CRV014 ; LINE ;  PT018 ; VEC002 .
        CRV016 ; CIRCLE ; PT024 ; VEC001 ; PT021 .      CURVES
        CRV017 ; CIRCLE ; PT028 ; VEC001 ; PT025 .
        CRV022 ; LINE ; PT042 ; VEC002 .
        CRV024 ; CIRCLE ; PT037 ; VEC001 ; PT034 .
        CRV025 ; LINE ;  PT038 ; VEC002 .
        CRV027 ; CIRCLE ; PT041 ; VEC001 ; PT038 .
        CRV030 ; LINE ;  PT044 ; VEC002 .
        CRV032 ; CIRCLE ; PT049 ; VEC001 ; PT046 .
        CRV033 ; LINE ;  PT047 ; VEC002 .
        CRV035 ; CIRCLE ; PT053 ; VEC001 ; PT050 .
        CRV036 ; CIRCLE ; PT057 ; VEC001 ; PT054 .
        CRV041 ; LINE ;  PT038 ; VEC001 .
        CRV042 ; LINE ;  PT035 ; VEC001 .
        CRV043 ; LINE ;  PT032 ; VEC001 .
        CRV044 ; LINE ;  PT034 ; VEC001 .
        CRV045 ; LINE ;  PT030 ; VEC001 .
        CRV046 ; LINE ;  PT051 ; VEC001 .
        CRV047 ; LINE ;  PT050 ; VEC001 .
        CRV048 ; LINE ;  PT047 ; VEC001 .
        CRV051 ; LINE ;  PT042 ; VEC001 .
        CRV052 ; LINE ;  PT039 ; VEC001 .
           CRV057 ; LINE ; PT056 ; VEC002 .
           CRV058 ; CIRCLE ; PT075 ; VEC001 ; PT060 .
           CRV059 ; CIRCLE ; PT020 ; VEC001 ; PT060 .
           CRV060 ; LINE ; PT064 ; VEC006 .
           CRV061 ; LINE ; PT062 ; VEC002 .
           CRV062 ; CIRCLE ; PT076 ; VEC008 ; PT062 .
           CRV063 ; CIRCLE ; PT045 ; VEC001 ; PT042 .
           CRV064 ; LINE ; PT044 ; VEC001 .
           CRV066 ; LINE ; PT064 ; VEC007 .
           CRV067 ; LINE ; PT046 ; VEC001 .
           CRV069 ; CIRCLE ; PT016 ; VEC001 ; PT013 .
           CRV070 ; LINE ; PT059 ; VEC002 .
           CRV071 ; CIRCLE ; PT077 ; VEC001 ; PT059 .
           CRV072 ; LINE ; PT065 ; VEC010 .
           CRV073 ; LINE ; PT065 ; VEC002 .
           CRV074 ; CIRCLE ; PT078 ; VEC013 ; PT055 .
           CRV075 ; CIRCLE ; PT004 ; VEC001 ; PT055 .
           CRV076 ; CIRCLE ; PT033 ; VEC001 ; PT032 .
           CRV077 ; LINE ; PT065 ; VEC008 .
           CRV078 ; CIRCLE ; PT008 ; VEC001 ; PT061 .
/END_CURVES
/POINTS
        PT001 ;  3.46875 ,  0.21875 ,  0.96875 .     POINTS
        PT004 ;  3.46875 ,  0.50000 ,  0.96875 .
        PT006 ;  2.25000 ,  1.00000 ,  0.96875 .
        PT008 ;  2.59090 ,  1.15625 ,  0.96875 .
        PT009 ;  1.50000 ,  1.00000 ,  0.96875 .
        PT010 ;  1.15910 ,  0.78125 ,  0.96875 .
```

```
        PT012 ;  1.15910 ,  1.15625 ,  0.96875 .
        PT013 ;  0.28125 ,  0.78125 ,  0.96875 .
        PT016 ;  0.28125 ,  0.50000 ,  0.96875 .
        PT018 ;  1.50000 ,  0.00000 ,  0.96875 .
        PT020 ;  1.15910 , -0.15625 ,  0.96875 .
        PT021 ;  2.25000 ,  0.00000 ,  0.96875 .
        PT022 ;  2.59090 ,  0.21875 ,  0.96875 .
        PT024 ;  2.59090 , -0.15625 ,  0.96875 .
        PT025 ;  2.18750 ,  0.50000 ,  0.96875 .
        PT026 ;  1.87500 ,  0.50000 ,  0.96875 .
        PT030 ;  3.46875 ,  0.21875 ,  0.00000 .
        PT032 ;  3.46875 ,  0.78125 ,  0.00000 .
        PT033 ;  3.46875 ,  0.50000 ,  0.00000 .
        PT034 ;  2.59090 ,  0.78125 ,  0.00000 .
        PT035 ;  2.25000 ,  1.00000 ,  0.00000 .
        PT037 ;  2.59090 ,  1.15625 ,  0.00000 .
        PT038 ;  1.50000 ,  1.00000 ,  0.00000 .
        PT039 ;  1.15910 ,  0.78125 ,  0.00000 .
        PT041 ;  1.15910 ,  1.15625 ,  0.00000 .
        PT042 ;  0.28125 ,  0.78125 ,  0.00000 .
        PT044 ;  0.28125 ,  0.21875 ,  0.00000 .
        PT045 ;  0.28125 ,  0.50000 ,  0.00000 .
        PT046 ;  1.15910 ,  0.21875 ,  0.00000 ,
        PT047 ;  1.50000 ,  0.00000 ,  0.00000 .
        PT049 ;  1.15910 , -0.15625 ,  0.00000 .
        PT050 ;  2.25000 ,  0.00000 ,  0.00000 .
        PT051 ;  2.59090 ,  0.21875 ,  0.00000 .
        PT053 ;  2.59090 , -0.15625 ,  0.00000 .
        PT054 ;  2.18750 ,  0.50000 ,  0.00000 .
            PT055 ;  3.67638 ,  0.68750 ,  0.96875 .
            PT056 ;  0.07162 ,  0.31250 ,  0.96875 .
        PT057 ;  1.87500 ,  0.50000 ,  0.00000 .
            PT058 ;  1.15910 ,  0.31250 ,  0.96875 .
            PT059 ;  2.59090 ,  0.68750 ,  0.96875 .
            PT060 ;  1.36931 ,  0.15430 ,  0.96875 .
            PT061 ;  2.38069 ,  0.84570 ,  0.96875 .
            PT062 ;  0.28125,  0.21875, 0.875 .
            PT063 ;  3.46875,  0.781250, 0.875 .
            PT064 ;  1.15910 ,  0.21875 ,  0.875 .
            PT065 ;  2.5909 ,  0.78125 ,  0.875 .
            PT068 ;  1.15910 ,  0.09375 ,  0.7500 .
            PT069 ;  2.59090 ,  0.90625 ,  0.7500 .
        PT075 ;  1.159100 , 0.09375 ,  0.96875 .
        PT076 ;  0.321170 , 0.41483 ,  1.07108 .
        PT077 ;  2.590900 , 0.90625 ,  0.96875 .
/END_POINTS
/UNIT_VECTORS
        VEC001 ;  0.00000 ,  0.00000 ,  1.00000 .       UNIT
        VEC002 ;  1.00000 ,  0.00000 ,  0.00000 .
        VEC003 ;  0.00000 ,  1.00000 ,  0.00000 .
            VEC006 ;  -0.259802 , 0.381694 , -0.887024   VECTORS
            VEC007 ;  0.00000 , -0.707107 , -0.707107 .
            VEC008 ;  0.00000 , -0.707107 , 0.707107 .
            VEC009 ;  -0.676490 , -0.73645 , 0.00000 .
            VEC010 ;  0.879462 , -0.269642 , -0.392225 .
            VEC013 ;  0.00000 , 0.707107,  0.707107 .
/END_UNIT_VECTORS
/END_GEOMETRY
/END_PART_MODEL
```

APPENDIX B



Feature Decomposition

```
/PART_MODEL
/HEADER
   PART_NAME = 'SIMPLE_A'
/END_HEADER
/BLANK_DATA
   BLANK_TYPE = 'RECTANGULAR'
/BLANK_TOPOLOGY
/BLANK_SHELLS
    SHELL001 :: TOP, BOTTOM, FAC003, FAC004, FAC005,
                   FAC006 .
/END_BLANK_SHELLS
/BLANK_FACES
    BOTTOM :: LOOP001, SURF001 - .
    TOP    :: LOOP002, SURF002 + .
    FAC003 :: LOOP003, SURF003 - .
    FAC004 :: LOOP004, SURF004 + .
    FAC005 :: LOOP005, SURF005 + .
    FAC006 :: LOOP006, SURF006 - .
/END_BLANK_FACES                                          LOOPS
/BLANK_LOOPS
    LOOP001 :: EDG001 +. EDG002 +, EDG003 +, EDG004 + .
    LOOP002 :: EDG005 +, EDG006 +, EDG007 +, EDG008 + .
    LOOP003 :: EDG001 +, EDG010 +, EDG005 -, EDG009 - .
    LOOP004 :: EDG002 +, EDG011 +, EDG006 -, EDG010 - .
    LOOP005 :: EDG003 +, EDG012 +, EDG007 -, EDG011 - .
    LOOP006 :: EDG004 +, EDG009 +, EDG008 -, EDG012 - .
/END_BLANK_LOOPS
/BLANK_EDGES
    EDG001 :: VTX001, VTX002, CRV001 + .
    EDG002 :: VTX002, VTX003, CRV002 + .
    EDG003 :: VTX003, VTX004, CRV003 + .
    EDG004 :: VTX004, VTX001, CRV004 + .
    EDG005 :: VTX005, VTX006, CRV005 + .  EDGES
    EDG006 :: VTX006, VTX007, CRV006 + .
    EDG007 :: VTX007, VTX008, CRV007 + .
    EDG008 :: VTX008, VTX005, CRV008 + .
    EDG009 :: VTX001, VTX005, CRV009 + .
    EDG010 :: VTX002, VTX006, CRV010 + .
    EDG011 :: VTX003, VTX007, CRV011 + .
    EDG012 :: VTX004, VTX008, CRV012 + .
END_BLANK_EDGES
/BLANK_VERTICES
    VTX001 :: POINT; PT001 .
    VTX002 :: POINT; PT002 .
    VTX003 :: POINT; PT003 .
    VTX004 :: POINT; PT004 .     VERTICES
    VTX005 :: POINT; PT005 .
    VTX006 :: POINT; PT006 .
    VTX007 :: POINT; PT007 .
    VTX008 :: POINT; PT008 .
/END_BLANK_VERTICES
/END_BLANK_TOPOLOGY
/BLANK_GEOMETRY
/BLANK_SURFACE
    SURF001 :: PLANE; VEC003, 0.0 .
```

```
     SURF002 :: PLANE; VEC003, 2.2 .
     SURF003 :: PLANE; VEC002, 0.0 .
     SURF004 :: PLANE; VEC001, 4.0 .          SURFACES
     SURF005 :: PLANE; VEC002, 4.0 .
     SURF006 :: PLANE; VEC001, 0.0 .
/END_BLANK_SURFACE
/BLANK_CRV
     CRV001 :: LINE; PT001, VEC001 .
     CRV002 :: LINE; PT002, VEC002 .
     CRV003 :: LINE; PT003, VEC001 .
     CRV004 :: LINE; PT004, VEC002 .   CURVES
     CRV005 :: LINE; PT005, VEC001 .
     CRV006 :: LINE; PT006, VEC002 .
     CRV007 :: LINE; PT007, VEC001 .
     CRV008 :: LINE; PT008, VEC002 .
     CRV009 :: LINE; PT001, VEC003 .
     CRV010 :: LINE; PT002, VEC003 .
     CRV011 :: LINE; PT003, VEC003 .
     CRV012 :: LINE; PT004, VEC003 .
/END_BLANK_CRV
/BLANK_COORD
     PT001 :: 0.0, 0.0, 0.0 .
     PT002 :: 4.0, 0.0, 0.0 .
     PT003 :: 4.0, 4.0, 0.0 .
     PT004 :: 0.0, 4.0, 0.0 .   POINTS
     PT005 :: 0.0, 0.0, 2.2 .
     PT006 :: 4.0, 0.0, 2.2 .
     PT007 :: 4.0, 4.0, 2.2 .
     PT008 :: 0.0, 4.0, 2.2 .
/END_BLANK_COORD
/BLANK_UNIT_VECTORS                  UNIT
     VEC001 :: 1.0, 0.0, 0.0 .          VECTORS
     VEC002 :: 0.0, 1.0, 0.0 .
     VEC003 :: 0.0, 0.0, 1.0 .
/END_BLANK_UNIT_VECTORS
/END_BLANK_GEOMETRY
/BLANK_MANUFACTURE
/BLANK_TOLERANCE
     TOL001 :: TOP; 0.000 + .
     TOL002 :: BOTTOM; 0.000 + .
     TOL003 :: FAC003; 0.000 - .          TOLERANCES
     TOL004 :: FAC004; 0.000 + .
     TOL005 :: FAC005; 0.000 + .
     TOL006 :: FAC006; 0.000 - .
/END_BLANK_TOLERANCE
/BLANK_MATERIAL
     MAT001 :: SHELL001; (xxx) .  MATERIALS
/END_BLANK_MATERIAL
/END_BLANK_MANUFACTURE
/END_BLANK_DATA
/FEATURE_SPECIFICATION
   FEATURE_TYPE = 'SLAB'      SLAB DEFINITIONS
/SLAB_DATA
/SLAB_TOPOLOGY
/SLAB_SHELLS
```

```
    SHELL001 :: TOP, BOTTOM, FAC003, FAC004, FAC005,
                FAC006 .
/END_SLAB_SHELLS
/SLAB_FACES
    BOTTOM :: LOOP001, SURF001 - .
    TOP    :: LOOP002, SURF002 + .
    FAC003 :: LOOP003, SURF003 - .    FACES
    FAC004 :: LOOP004, SURF004 + .
    FAC005 :: LOOP005, SURF005 + .
    FAC006 :: LOOP006, SURF006 - .
/END_SLAB_FACES
/SLAB_LOOPS                                          LOOPS
    LOOP001 :: EDG001 +, EDG002 +, EDG003 +, EDG004 + .
    LOOP002 :: EDG005 +, EDG006 +, EDG007 +, EDG008 + .
    LOOP003 :: EDG001 +, EDG010 +, EDG005 -, EDG009 - .
    LOOP004 :: EDG002 +, EDG011 +, EDG006 -, EDG010 - .
    LOOP005 :: EDG003 +, EDG012 +, EDG007 -, EDG011 - .
    LOOP006 :: EDG004 +, EDG009 +, EDG008 -, EDG012 - .
/END_SLAB_LOOPS
/SLAB_EDGES
    EDG001 :: VTX001, VTX002, CRV001 + .
    EDG002 :: VTX002, VTX003, CRV002 + .
    EDG003 :: VTX003, VTX004, CRV003 + .   EDGES
    EDG004 :: VTX004, VTX001, CRV004 + .
    EDG005 :: VTX005, VTX006, CRV005 + .
    EDG006 :: VTX006, VTX007, CRV006 + .
    EDG007 :: VTX007, VTX008, CRV007 + .
    EDG008 :: VTX008, VTX005, CRV008 + .
    EDG009 :: VTX001, VTX005, CRV009 + .
    EDG010 :: VTX002, VTX006, CRV010 + .
    EDG011 :: VTX003, VTX007, CRV011 + .
    EDG012 :: VTX004, VTX008, CRV012 + .
END_SLAB_EDGES
/SLAB_VERTICES
    VTX001 :: POINT; PT001 .
    VTX002 :: POINT; PT002 .   VERTICES
    VTX003 :: POINT; PT003 .
    VTX004 :: POINT; PT004 .
    VTX005 :: POINT; PT005 .
    VTX006 :: POINT; PT006 .
    VTX007 :: POINT; PT007 .
    VTX008 :: POINT; PT008 .
/END_SLAB_VERTICES
/END_SLAB_TOPOLOGY
/SLAB_GEOMETRY
/SLAB_SURFACE
    SURF001 :: PLANE; VEC003, 2.0 .
    SURF002 :: PLANE; VEC003, 2.2 .   SURFACES
    SURF003 :: PLANE; VEC002, 0.0 .
    SURF004 :: PLANE; VEC001, 4.0 .
    SURF005 :: PLANE; VEC002, 4.0 .
    SURF006 :: PLANE; VEC001, 0.0 .
/END_SLAB_SURFACE
/SLAB_CRV
    CRV001 :: LINE; PT001, VEC001 .
```

```
     CRV002 :: LINE; PT002, VEC002 .
     CRV003 :: LINE; PT003, VEC001 .
     CRV004 :: LINE; PT004, VEC002 .
     CRV005 :: LINE; PT005, VEC001 .
     CRV006 :: LINE; PT006, VEC002 .     CURVES
     CRV007 :: LINE; PT007, VEC001 .
     CRV008 :: LINE; PT008, VEC002 .
     CRV009 :: LINE; PT001, VEC003 .
     CRV010 :: LINE; PT002, VEC003 .
     CRV011 :: LINE; PT003, VEC003 .
     CRV012 :: LINE; PT004, VEC003 .
/END_SLAB_CRV
/SLAB_COORD
     PT001 :: 0.0, 0.0, 2.0 .
     PT002 :: 4.0, 0.0, 2.0 .
     PT003 :: 4.0, 4.0, 2.0 .
     PT004 :: 0.0, 4.0, 2.0 .   POINTS
     PT005 :: 0.0, 0.0, 2.2 .
     PT006 :: 4.0, 0.0, 2.2 .
     PT007 :: 4.0, 4.0, 2.2 .
     PT008 :: 0.0, 4.0, 2.2 .
/END_SLAB_COORD
/SLAB_UNIT_VECTORS                 UNIT
     VEC001 :: 1.0, 0.0, 0.0 .    VECTORS
     VEC002 :: 0.0, 1.0, 0.0 .
     VEC003 :: 0.0, 0.0, 1.0 .
/END_SLAB_UNIT_VECTORS
/END_SLAB_GEOMETRY
/SLAB_MANUFACTURE
/SLAB_TOLERANCE
     TOL001 :: TOP; 0.000 + .
     TOL002 :: BOTTOM; 0.000 + .
     TOL003 :: FAC003; 0.000 - .
     TOL004 :: FAC004; 0.000 + .
     TOL005 :: FAC005; 0.000 + .
     TOL006 :: FAC006; 0.000 - .
/END_SLAB_TOLERANCE
/SLAB_MATERIAL
     MAT001 :: SHELL001; (xxx) .
/END_SLAB_MATERIAL
/END_SLAB_MANUFACTURE
/END_SLAB_DATA               POCKET DEFINITIONS
   FEATURE_TYPE = 'POCKET'
/POCKET_DATA
/POCKET_TOPOLOGY
/POCKET_SHELLS
     SHELL001 :: TOP, BOTTOM, FAC003, FAC004, FAC005,
                    FAC006 .
/END_POCKET_SHELLS
/POCKET_FACES
     BOTTOM :: LOOP001, SURF001 - .
     TOP    :: LOOP002, SURF002 + .         FACES
     FAC003 :: LOOP003, SURF003 - .
     FAC004 :: LOOP004, SURF004 + .
     FAC005 :: LOOP005, SURF005 + .
```

```
    FAC006 :: LOOP006, SURF006 - .
/END_POCKET_FACES                                        LOOPS
/POCKET_LOOPS
    LOOP001 :: EDG001 +, EDG002 +, EDG003 +, EDG004 + .
    LOOP002 :: EDG005 +, EDG006 +, EDG007 +, EDG008 + .
    LOOP003 :: EDG001 +, EDG010 +, EDG005 -, EDG009 - .
    LOOP004 :: EDG002 +, EDG011 +, EDG006 -, EDG010 - .
    LOOP005 :: EDG003 +, EDG012 +, EDG007 -, EDG011 - .
    LOOP006 :: EDG004 +, EDG009 +, EDG008 -, EDG012 - .
/END_POCKET_LOOPS
/POCKET_EDGES
    EDG001 :: VTX001, VTX002, CRV001 + .
    EDG002 :: VTX002, VTX003, CRV002 + .
    EDG003 :: VTX003, VTX004, CRV003 + .
    EDG004 :: VTX004, VTX001, CRV004 + .      EDGES
    EDG005 :: VTX005, VTX006, CRV005 + .
    EDG006 :: VTX006, VTX007, CRV006 + .
    EDG007 :: VTX007, VTX008, CRV007 + .
    EDG008 :: VTX008, VTX005, CRV008 + .
    EDG009 :: VTX001, VTX005, CRV009 + .
    EDG010 :: VTX002, VTX006, CRV010 + .
    EDG011 :: VTX003, VTX007, CRV011 + .
    EDG012 :: VTX004, VTX008, CRV012 + .
/END_POCKET_EDGES
/POCKET_VERTICES
    VTX001 :: POINT; PT001 .
    VTX002 :: POINT; PT002 .
    VTX003 :: POINT; PT003 .
    VTX004 :: POINT; PT004 .     VERTICES
    VTX005 :: POINT; PT005 .
    VTX006 :: POINT; PT006 .
    VTX007 :: POINT; PT007 .
    VTX008 :: POINT; PT008 .
/END_POCKET_VERTICES
/END_POCKET_TOPOLOGY
/POCKET_GEOMETRY
/POCKET_SURFACE                            SURFACES
    SURF001 :: PLANE; VEC003, 1.0 .
    SURF002 :: PLANE; VEC003, 2.0 .
    SURF003 :: PLANE; VEC002, 1.0 .
    SURF004 :: PLANE; VEC001, 3.0 .
    SURF005 :: PLANE; VEC002, 3.0 .
    SURF006 :: PLANE; VEC001, 1.0 .
/END_POCKET_SURFACE
/POCKET_CRV
    CRV001 :: LINE; PT001, VEC001 .   CURVES
    CRV002 :: LINE; PT002, VEC002 .
    CRV003 :: LINE; PT003, VEC001 .
    CRV004 :: LINE; PT004, VEC002 .
    CRV005 :: LINE; PT005, VEC001 .
    CRV006 :: LINE; PT006, VEC002 .
    CRV007 :: LINE; PT007, VEC001 .
    CRV008 :: LINE; PT008, VEC002 .
    CRV009 :: LINE; PT001, VEC003 .
    CRV010 :: LINE; PT002, VEC003 .
```

```
    CRV011 :: LINE; PT003, VEC003 .
    CRV012 :: LINE; PT004, VEC003 .
/END_POCKET_CRV
/POCKET_COORD
    PT001 :: 1.0, 1.0, 1.0 .
    PT002 :: 1.0, 3.0, 1.0 .
    PT003 :: 3.0, 3.0, 1.0 .
    PT004 :: 1.0, 3.0, 1.0 .        POINTS
    PT005 :: 1.0, 1.0, 2.0 .
    PT006 :: 1.0, 3.0, 2.0 .
    PT007 :: 3.0, 3.0, 2.0 .
    PT008 :: 1.0, 3.0, 2.0 .
/END_POCKET_COORD
/POCKET_UNIT_VECTORS
    VEC001 :: 1.0, 0.0, 0.0 .        UNIT
    VEC002 :: 0.0, 1.0, 0.0 .        VECTORS
    VEC003 :: 0.0, 0.0, 1.0 .
/END_POCKET_UNIT_VECTORS
/END_POCKET_GEOMETRY
/POCKET_MANUFACTURE
/POCKET_TOLERANCE
    TOL001 :: TOP; 0.000 + .
    TOL002 :: BOTTOM; 0.000 + .
    TOL003 :: FAC003; 0.000 - .
    TOL004 :: FAC004; 0.000 + .
    TOL005 :: FAC005; 0.000 + .
    TOL006 :: FAC006; 0.000 - .
/END_POCKET_TOLERANCE
/END_POCKET_MANUFACTURE
/END_POCKET_DATA
/END_FEATURE_SPECIFICATION
/END_PART_MODEL
```

AN EXAMINATION OF FEATURE BASED MODELING
AND SYSTEMS UTILIZING FEATURE TECHNIQUES

by

DENNIS JAY HANER

B.S., Kansas State University, 1986

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

ABSTRACT

This document addresses the problem of successful passage of manufacturing data from the designers' model (computer aided design) to the production facilities (computer aided manufacturing). The goal of this document is to present Feature Based Modeling, an alternative method of modeling objects for production.

Chapter 1 of the paper provides the reader with topic introduction and methodology. Chapter 2 provides a review of geometric modeling. Feature based modeling is a subset of geometric modeling. Chapter 3 provides a general discussion of feature based modeling. Chapter 4 discusses three systems which utilize feature based modeling. The three are: FEATURES, XCUT, and Attributed Adjacency Graph. Chapter 5 provides concluding remarks to this work.

The key issue is to resolve information content problems between the design and manufacturing processes. The design point of view considers a part to be constructed with features. The manufacture point of view considers a part to be a block of material from which features have been removed. Information content within the features may provide the means to resolve data problems.