CAD/CAM Integration Based on Machining Features for Prismatic Parts

By

Min Hou

Submitted to the Department of Mechanical Engineering and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy

____
Chairperson

Committee members: ______________________________

____

____

____

Date defended: ____________________

The Dissertation Committee for Min Hou certifies
that this is the approved version of the following dissertation:

CAD/CAM Integration Based on Machining Features for Prismatic Parts

Committee: ______________________________________
Chairperson

____

____

____

____

Date approved: __________________________

Abstract

The development of CAD and CAM technology has significantly increased efficiency in each individual area. The independent development, however, greatly restrained the improvement of overall efficiency from design to manufacturing. The simple integration between CAD and CAM systems has been achieved. Current integrated CAD/CAM systems can share the same geometry model of a product in a neutral or proprietary format. However, the process plan information of the product from CAPP systems cannot serve as a starting point for CAM systems to generate tool paths and NC programs. The user still needs to manually create the machining operations and define geometry, cutting tool, and various parameters for each operation.

Features play an important role in the recent research on CAD/CAM integration. This thesis investigated the integration of CAD/CAM systems based on machining features. The focus of the research is to connect CAPP systems and CAM systems by machining features, to reduce the unnecessary user interface and to automate the process of tool path preparation. Machining features are utilized to define machining geometries and eliminate the necessity of user interventions in UG. A prototype is developed to demonstrate the CAD/CAM integration based on machining features for prismatic parts. The prototype integration layer is implemented in conjunction with an existing CAPP system, FBMach, and a commercial CAD/CAM system, Unigraphics. Not only geometry information of the

product but also the process plan information and machining feature information are directly available to the CAM system and tool paths can be automatically generated from solid models and process plans.

Acknowledgements

My most sincere thanks go to my advisor, Dr. Terry Faddis, for guiding and encouraging me to finish this dissertation. He directed my dissertation research and other projects in the CAD/CAM field, which pose a great impact on my development. I would like to thank Dr. Carl Luchies, Dr. Robert Sorem, Professor Robert Umholtz and Dr. James Miller for their advice and support as members of my dissertation committee.

Special thanks go to Mr. Bill Simons and Dr. Steve Brooks in Honeywell FM&T for their help in the usage of FBMach, coordination on the research, and discussions about the details of my research.

I would like to thank the people at the Intelligent Systems and Automation lab of Mechanical Engineering for the pleasant research environment and their friendship, Bryce Baker, Feiqi Zhang, Pratya Levin, and Tejukiran Murthy. Bryce Baker helped me in many ways, discussing research work, managing the lab's network and talking about the culture. I enjoyed academic discussions on the research and everyday talks with Feiqi Zhang. Tejukiran Murthy worked with me to set the geometry of planar milling operations in the research.

Finally I would like to thank my parents, my husband and my daughter whose love, support and encouragement have helped me through the journey.

Table of Content

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

Computers have been greatly involved in product development throughout the product lifecycle, from design, analysis, to manufacturing. Since the concept of Computer Aided Design (CAD) emerged in early 1950s, [1] it has extended into every aspect of product design and development. CAD systems were developed from simple two-dimensional (2D) drafting tools to currently much more powerful systems based on solid modeling technology. Around the same time, Computer Aided Manufacturing (CAM) was inspired by Numerical Controlled (NC) machines. CAM systems were developed separately from CAD systems by different user groups. [1] The development of CAD and CAM technology significantly increased efficiency in each individual area. The independent development, however, greatly restrained the improvement of overall efficiency from design to manufacturing. The communication between CAD and CAM systems has become a bottleneck for further improvement of production efficiency.

The first effort to break the isolation of CAD and CAM systems was to reuse the product model designed in CAD systems in CAM systems. [2] It made CAM systems able to directly manipulate CAD models, either the wire frame or solid model. But the model given by CAD systems is only the product model of the final shape. The traditional CNC machines used in machine shops take NC programs, i.e.

G/M codes, as input to drive CNC machines. NC programmers need to know the intermediate shapes of the product to create machine codes for each machining step. Intermediate shapes of intermediate processes in manufacturing are not available from CAD systems and have to be obtained from creating new models or editing the final model. It requires a great amount of user interactions and knowledge of the CNC machine and processes to generate machine codes.

CAD/CAM technologies have continued to evolve. The current trend is feature-based systems. Features play a key role in the recent integration of CAD/CAM systems. Automatic feature recognition has been successful to a certain extent and applied to Computer Aided Process Planning (CAPP) systems. Feature-based CAPP interprets the product model in terms of machining features and uses the features to generate manufacturing instructions to produce the product. With the help of automatic feature recognition, CAPP systems can recognize features directly from solid models created by CAD systems and generate process plans for the solid models. A product has to be manufactured to bring it to life. However, the process plans generated from CAPP systems cannot serve as a starting point for CAM systems to generate tool paths and NC programs. The user still needs to manually create the machining steps and define geometry for each machining step. The situation must be improved to achieve a seamless CAD/CAM integration.

Research in the Intelligent Systems and Automation (ISA) laboratory of Mechanical Engineering at the University of Kansas with Honeywell has applied automatic feature recognition to solid models for creating process plans based upon the solid models and machine tool information. [3] This research is to link CAPP systems and CAM systems by machining features, to reduce the unnecessary user interface and to automate the process of tool path preparation. The system is expected to be easy to expand and to accurately transfer data.

1.2 Research Objectives

Product development is a long process from initial concept to final manufactured product involving design, planning and manufacture activities. There are many CAD systems to take care of product design, CAM systems to generate tool paths and NC programs and CAPP systems for process planning. However, the links between those different systems are weak. Currently the information from CAD and CAPP systems cannot be interpreted directly by CAM systems. It is the users who interpret the information, create desired machining operations and prepare geometry for tool path generation and NC part programs according to a process plan generated in CAPP systems. The emergence of STEP-NC [4] addresses this problem, but it is in its initial phase and there are many traditional NC machines currently being used that take NC programs as input so that the transition to STEP-NC will not be short.

The goal of this research work is to provide a seamless CAD/CAM integration through using machining features, to make product as well as process information available immediately in an electronic form for the preparation of NC part programs and to automate the process of tool path generation from the solid model of a part in a

CAM system. This will significantly reduce the user interactions and the amount of time it normally would have required to generate the information. A prototype has been implemented in conjunction with Feature-Based Machining Advisor (FBMach) [5] and Unigraphics (UG) [6]. An integration layer has been developed to read a process plan from FBMach, create machining operations, map the machine features to UG machining geometry and finally automatically generate tool paths for the part. All the information coming from the CAPP system (FBMach), through the integration layer, can be understood by the CAM system (UG), therefore there will be much less user interaction required to prepare NC part programs. The research focuses on prismatic milling and drilling machining features, including pocket, slot, step, periphery, cutout, planar face, general removal and hole features, and corresponding machining operations.

1.3 Thesis Organization

This thesis consists of six chapters and two appendices. Chapter 2 discusses the importance and current status of CAD/CAM integration, reviews the concept and usage of feature technology in CAD/CAM and overviews the information modeling and data transfer. Chapters 3 presents the architecture of the integrated CAD/CAM system based on machining features, analyzes the information of process plans and features available from FBMach and describes the process of tool path generation and the objects essential to tool path generation in UG. Chapter 4 discusses the issues of the system implementation, defines the two data models for the process plan and

CAM objects, and describes the mapping of process plan information to CAM objects. Chapter 5 demonstrates the integration with four examples and discusses the pros and cons of the system. Chapter 6 summarizes the research and identifies a number of research directions for future work. Appendix A lists a process plan file for one hole feature exported from FBMach. Appendix B illustrates the complete data model of UG CAM objects in EXPRESS-G.

2 Literature Review of Related Research Work

2.1 The integration of CAD/CAM systems

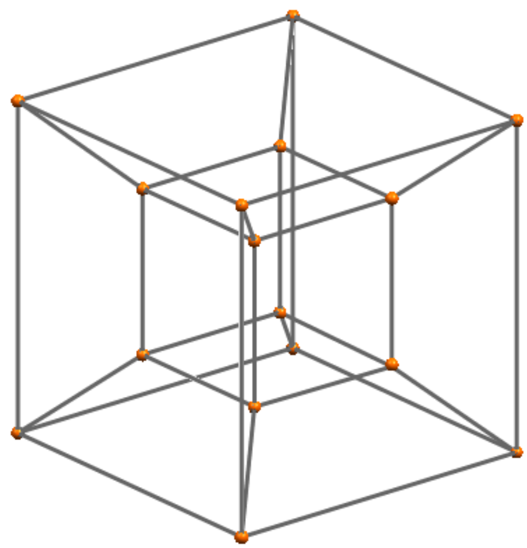
CAD/CAM systems have helped speed many tedious steps in the concept to production cycle since they were employed in industry. They replace or assist many decision-making functions and dramatically increase productivity. Soon after CAD and CAM systems were developed, users realized that the communication between CAD and CAM systems became the bottleneck for the improvement of productivity, so the integration of CAD and CAM systems started drawing researchers attention. For the purpose of the research conducted in this dissertation, only mechanical CAD and CAM systems and their integration are discussed.

2.1.1 Development of CAD technology

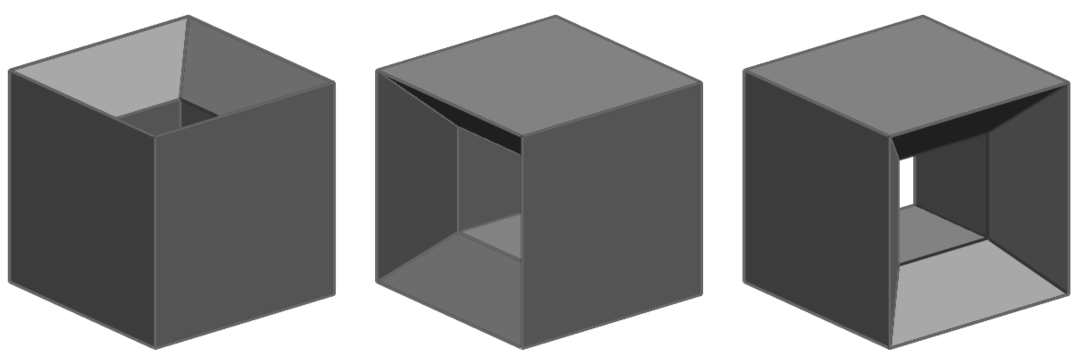
CAD system is widely used to describe any software system capable of defining components with geometry. [1] It uses computers to aid in the process of product design and development. CAD originated from early computer graphic systems, and evolved with the development of interactive computer graphics and geometric modeling technology. Computers have long been used for engineering calculations in batch job mode before the emergence of CAD systems. The development of Sketchpad system at MIT in 1963 by Dr. Ivan Sutherland was a turning point. [1] Sketchpad is considered to be the ancestor of modern CAD systems as well as a major breakthrough in computer graphics in general. Sketchpad was the first system that allowed a designer to interact with a computer graphically by drawing on a CRT monitor with a light pen. It presented a prototype of a graphical user interface, an indispensable feature of modern CAD systems.

CAD technology has evolved dramatically since it was born. In the early 1970s CAD systems were little more than drafting software used to create 2D drawings similar to hand-drafted drawings. CAD was often referred to as Computer Aided Drafting at that time. The geometry available to the user was limited to simple geometry, like lines, circular arcs and ellipse arcs. Advances in programming and computer hardware, notably solid modelling in the 1970s, have allowed more versatile CAD applications in design activities. As the geometric modeling technology has progressed from simple 2D drafting, to three-dimensional (3D) wire-frame, to 3D surfaces and now 3D solid modeling, so have CAD systems. With the rapid development of CAD systems from simple 2D drafting systems to complex 3D modeling systems, they are now being used throughout the engineering process from conceptual design and detailed engineering, through strength, dynamic analysis of components and assembly planning.

The advent of 3D solid modeling marked the beginning of a new era in the 1970s in CAD. Solid modeling creates unambiguous and complete geometric representations of objects unlike wire-frame models. Wire-frame models are ambiguous in the sense that several interpretations might be possible for a single model. A well-known example [7] is shown in figure 2.1. The example has a beveled hole through its center, but we cannot tell the direction of the hole from the wire-frame model. There are three possibilities for the opening direction as shown. In addition, it is virtually impossible to find the volumetric information of the model



a. an ambiguous wire-frame model



b. three interpretations of the wire-frame model

Figure 2.1 A wire-frame model with multiple interpretations from wire-frame or surface models, while it is not a problem for solid models. Solid models have unambiguous representations and contain complete information, therefore not only can they be used to produce engineering drawings, but engineering analysis can be performed on the same models as well. There have been several different approaches for solid modeling. The two approaches most commonly adopted are [8] constructive solid geometry (CSG) and boundary representation (B-rep).

Solid modeling has become a mature and popular tool in the design of mechanical parts as well as in other areas. Today the use of solid modeling is a common practice in commercial CAD systems. The development of B-rep solid modeling kernels, like Parasolid and ACIS, at the end of the 1980s played an important role to the development of modern CAD systems. Nowadays CAD is not limited to drafting and rendering, and it ventures into many intellectual areas of a designer's expertise, such as engineering analysis and assembly simulation.

2.1.2 Development of CAM technology

CAM technology was sparked by the invention of NC machine tools. NC machine tools were developed to manufacture complex shapes in an accurate and repeatable manner. NC machines are directed by part programs following industrial data standard, RS274D, known as ISO 6983 internationally. [9] The standard defines a set of M and G codes which specify a sequence of cutting tool movements as well as the direction of rotation, speed of travel and various auxillary functions such as coolant flow. NC programs are lengthy and must specify each single movement of the machine tool. NC programs are difficult to create or edit by hand. An example of NC part programs for a drilling operation is given in figure 2.2. Simple NC programs, like point-to-point processes, can be created manually, usually with the aid of a calculator. For more complex programs, however, it is very time consuming and subject to human errors to manually generate NC programs from drawings. CAM systems were developed to use computers to prepare and generate part programs for NC machines.

```
N0010 G40 G17 G90 G70
N0020 G91 G28 Z0.0
N0030 T01 M06
N0040 G00 G90 X4.2445 Y-9.8098 S500 M03
N0050 G43 Z5.9207 H00 M08
N0060 G81 Z5.7644 R5.9207 F10.
N0070 G80
N0080 M09
N0090 G00 Z8.
N0100 X3.6 Y-9.2
N0110 M02
```

Figure 2.2 NC programs for a drilling operation

The first generation of CAM emerged when Automatically Programmed Tool (APT) was developed to help control NC machines at the Massachusetts Institute of Technology (MIT) in the 1950s. [11] APT is a universal programming language for NC machines and has been widely adopted in industry. It provides a convenient way to define geometry elements and generate cutter locations (CL) for NC programs by computers. At first, APT could handle only relatively simple geometry; points, lines, circles, planes, quadratic surfaces etc. Later on, the handling of more complex geometry was made possible. With the help of APT, early CAM systems can create a part drawing and convert the drawing into NC programs so that a NC machine can manufacture the part from the part programs. APT was created before graphical interfaces were available, so it relies on text to specify the geometry and toolpaths needed to machine a part.

Even though APT offers advantages over manual approach, using APT involves defining compreshensive geometries and tool positioning commands, which poses a significant potential for errors in the process. To overcome this problem, graphics based CAM was introduced and became popular in 1980s. [11] This allows part geometry to be described in the form of points, lines, arcs, and so on, rather than requiring a translation to a text oriented notation. The user can more rapidly define the geometry as well as use powerful graphics display capabilities to quickly define, verify, and edit the actual cutter motions. Graphics display also allows the system to display the resulting tool path on the monitor, making earlier verification of a program possible, which can avoid costly machine setups for prototype testing.

Despite starting independently of CAM, CAD had great influence to later development of CAM. CAM started reusing part geometric models from CAD very shortly after graphics based CAM was introduced. The geometric models that are used to generate tool paths and NC codes advanced from 2D drawing, 3D wire-frame, and surface models to solid models, along with CAD development. When the tool paths are generated using a wire-frame model, there is no way to detect interference between the cutting tool, the part and the fixture. When using a solid model, the entire machining environment, including stock, part volume, tools, and fixtures, can be modeled and used to generate a collision-free tool path. It is also make it possible to incorporate geometric reasoning rules and machining knowledge to select feeds, speeds, tools, and operation sequence.

With the continuing demand for ease of use and productivity improvement, greater automation is being embedded into all aspects of CAM products, from the user interface to post processors. Machining intelligence built into a CAM system makes the system produce more consistent NC code in less time. Computerized Numerical Control (CNC) has been challenged by the trend to develop an open CNC architecture. [12] Open architecture control has a common architecture of system components and interfaces, therefore would provide unconstrained access to all data within the control, a friendly configuration interface for users, and improved machine tool communication.

2.1.3 CAPP and CAD/CAM integration

As discussed early in this section the utilization of computers in design and manufacture started independently in their own area with no apparent link between them. [1] Initially the development of CAD systems had little effect on CAM development due to the different capabilities and file formats used by drawings and NC programs. The result was that a lot of CAM programming time was spent redefining the part geometry, which had already been defined in CAD. [2] The realization of this result led to the appearance of the first integrated CAD/CAM systems, in which the CAM system could begin work with the geometric model created by the CAD system and not require the part model to be recreated. The benefits of integrated CAD/CAM systems include decreased time to market, lower development and design cost and the ability to rapidly translate ideas into models. [11] Current major commercial CAD/CAM systems, such as Unigraphics, Pro/E, I-DEAS, CATIA, etc. have many specialized modules packed together and running on their own proprietary databases. The integration of these systems implies the system has both CAD and CAM capabilities and the geometric data from CAD can be used in the CAM module without conversion in addition to the uniform application interface.

Integration plays an increasingly important role in CAD/CAM systems while the technology of CAD/CAM is rapidly developing and widely spread in industry. Nowadays integrated CAD/CAM systems have the ability to share the same part model. CAM systems can import 3D models from CAD, either wire-frame, surface or solid models. However, a part represented in a solid model designed by a CAD system cannot be directly machined on a CNC machine. The current method to generate tool paths and NC programs needs extensive user interactions in CAM systems. To machine a part, users must create machining operations with appropriate parameters, prepare machining geometries manually and specify cutting tools, so that the CAM system can generate the tool paths and CNC programs. The whole process is time-consuming and error-prone. Very often, the time needed in generating tool paths and NC programs for a part is considerably longer than the actual machining time. [13] To improve the integration of CAD/CAM systems, it is essential to improve the communication between CAD and CAM systems and reduce data redundancy.

The bridge between design and manufacture is process planning. Process planning is the process of determining detailed operation instructions to transform an engineering design to a final part. [11] The size, shape, tolerances and finishes of the part all affect process plans. The process planning task involves many activities, like selection of machining operations, cutting tools, and cutting parameters, ordering of operations, and calculation of setup and fixtures. All the activities are closely related and dependent on each other. Problems with manual process planning, such as lack of expertise, inconsistency of the plans, and the need to automate the process planning task, have led to CAPP systems.

CAPP has evolved to simplify and improve process planning and use product information and manufacturing resources more effectively. The goal of CAPP is to generate a sequenced set of instructions used to manufacture the specified part, which then can be applied to downstream applications, like CAM. There are two fundamental methodologies used in CAPP systems, the variant and the generative approaches. [11] In the variant approach, a set of standard process plans is stored for the part families identified through group technology. The plans for new parts are derived from the modification of the standard process plans of similar part families.

In the generative approach, process plans will start from scratch instead of an existing plan. A new process plan is created based on the analysis of part geometry, material, and other factors that may influence the manufacturing decisions.

CAPP is seen as a communication agent between CAD and CAM systems. [14] To communicate effectively between CAPP and CAD/CAM systems it is necessary to provide transmission of two types of information: geometric data that describes the design of a part, and technological data that describes the way of machining the part. Feature-based CAPP quickly attracted researchers' attention and plays an important role in CAD/CAM integration. Feature technology is able to provide an adequate basis for the integration of design and subsequent applications such as engineering analysis, process planning, machining and inspection. Feature-based CAPP interprets the product model in terms of machining features and uses the features to generate manufacturing instructions to produce the product. For instance, CAPP typically generates drilling operations for hole features.

A great deal of research has been conducted and many results have been achieved on feature-based CAPP. [15] XCUT [16] is an expert process planning system that can analyze solid model product representations and recognize features required by subsequent manufacturing processes. Khoshnevis, et al., [17] developes a 3I-PP (Intelligent Integrated Incremental Process Planning) system, which applies a knowledge-based approach to feature completion and process selection, and the space search algorithm for process sequencing. The process planning system provides integration of CAPP with CAD and scheduling systems. Han, et al. [14] present the effort on feature-based machining sequence generation based on tool capabilities. Machining sequence is generated based on tool capabilities and is optimized with the aid of the feature dependencies and a manufacturing cost function. Eventually, a setup sequence is generated where an optimal machining sequence is determined per each setup. The system uses the Standard for Exchange for Product Model Data (STEP) as input and output formats and therefore can be ported to arbitrary CAD and planning systems.

Although there is a great deal of research on feature-based CAPP, which focuses on the link with design and process planning, not much research focuses on the link with process planning and manufacturing. Liang [13] and Miao [18] showed their progress of the research on both links. M. Liang, et al. [13] report the development of a STEP based tool path generation system in a Unigraphics environment for rough machining of planar surfaces. The system is featured with a data extraction module, a volume slicing module, a CL file generation module and an NC code generator. In the system, a STEP file of the design model is processed by the data extraction module and the internal and external features of the model are recognized. Then the features are handled by tool path generation algorithms. The tool path is automatically generated based on the STEP file and does not need intermediate data exchange. The system only deals with rough machining of planar surfaces. Though rough machining represents a significant amount of machining time, it needs to be extended to handle finish cutting and geometries other than planar surfaces.

Miao, et al. [18] demonstrate the use of features in automating certain process planning tasks and integrating CAD and CAM modules in a commercial CAD/CAM system (I-DEAS). In the system, CAPP is achieved by automatically extracting machining features from the CAD model and then using knowledge-based methods to prepare a process plan for the part. Set-up planning, operation sequencing and tool selection are performed automatically based on criteria such as feature shape, feature locations, tool access directions and feasibility of work piece locating and clamping. Features and manufacturing attributes are exported to I-DEAS for tool path generation and verification. A 3-axis feature taxonomy has been defined in the system, including inner profile, outer profile, through hole, blind hole and volume-clear features, as shown in figure 2.3. The feature taxonomy used was very broadly defined.

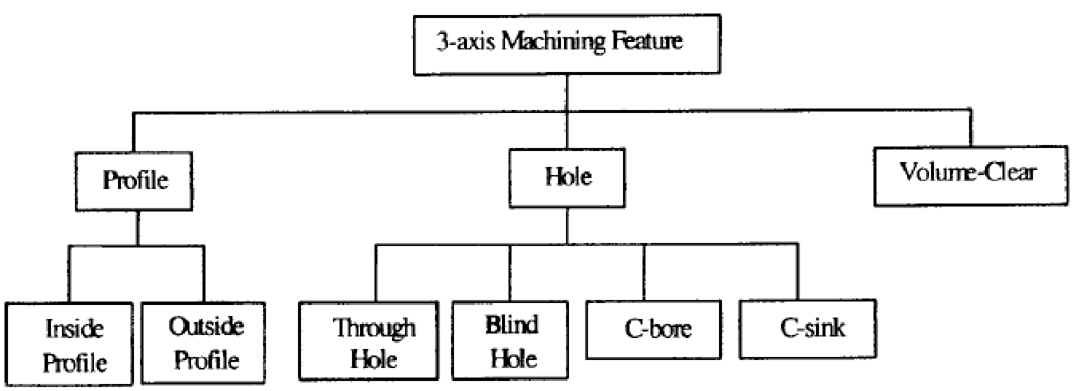


Figure 2.3 The feature taxonomy of Miao's system [18]

2.2 Feature technology in CAD/CAPP/CAM

It has been widely accepted that feature technology plays a key role in achieving CAD/CAM integration. A feature is a collection of topological and geometrical entities that are grouped together so engineers can associate knowledge useful for reasoning about the part. [19] Different users have different views of what is important for a given shape due to various functions, such as design, analysis, assembly, and manufacturing. A feature model can be constructed by two approaches, feature-based design or feature recognition.

2.2.1 Feature-based design

A product model can be built by using a set of design features, which is known as feature-based design. [20] A design feature is a shape that has significance to the design engineer. Design with pre-defined features can reduce the designer's work substantially. It is very difficult to provide a set of features for every conceivable situation and it would result a large unwieldy library. One advantage of feature-based design is that existing designs can be modified and reused more easily. The designer can simply add, delete, or modify the features of existing models to have a new design. The modification of features is more tractable than directly changing the underlying geometry. [21]

In the view of design, features can be addiction or subtraction to a solid model, which often differ from downstream application features. Design features cannot always be used in process planning and manufacturing directly. Figure 2.4 gives a frequently quoted example of different views on design features and machining features. The part may be modeled by adding ribs to the base block by designers. However, machining features of the same part would be slots and a step that correspond to the material to be removed from the stock.

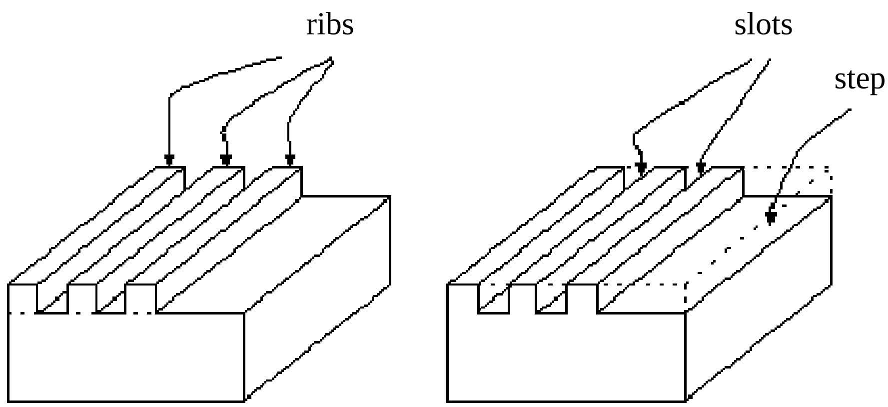


Figure 2.4 Design features vs. machining features [20]

One way to deal with the problem of different views of features in design and manufacturing is to use manufacturing features in both the design and manufacturing domain. However, it limits the freedom and flexibility of designers to design with only manufacturing features, especially for parts that need to be machined. [20] Moreover, to design with manufacturing features the designer must make detailed low level manufacturing decisions early in the design process. Another solution to different views of features is to use different sets of features in design and manufacturing and then convert the design features to manufacturing features, a process called feature model conversion. [22] Shah defines feature sets for converting between design and manufacturing. [23] In this way, the feature information that is generated during design is discarded and manufacturing related feature information is inferred from the solid model of the part. The conversion is not always possible and at best is limited.

2.2.2 Machining features

A machining feature is typically defined as a collection of related geometric elements that as a whole correspond to a particular machining method or process for creating the geometry. [19] Machining feature is a high-level product geometric representation and macro description of the product geometry. This high-level information allows engineers to realize some applications like design for manufacturing and automatic process planning. Typical machining features would indicate machinable features, such as holes, slots, and pockets. Examples of machining features are illustrated in figure 2.5.

The part AP224 of STEP defines a set of machining features for use in process planning. [24] AP224 facilitates the identification of computer-interpretable manufacturing shapes. In AP224, a machining feature is a type of manufacturing feature that identifies a volume of material that shall be removed from the initial stock to obtain the final part geometry. A manufacturing feature identifies the types of features necessary to manufacture a machined part. Each manufacturing feature is either a machining feature, a replicate feature, or a transition feature. Machining features require both direction and location to place them on a part. It is desirable to

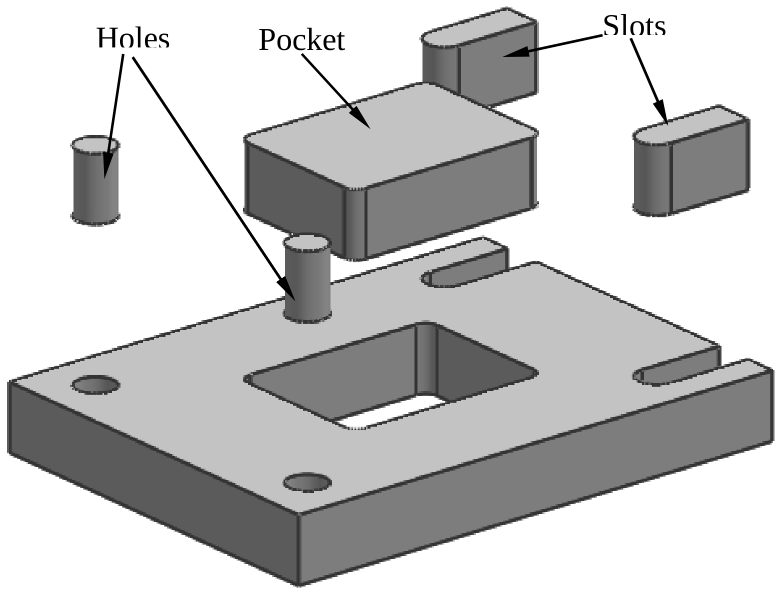


Figure 2.5 Examples of machining features

make the location of each feature in a common super-type with coordinates in the center of the feature and at the top of the stock because this is how the machinist will measure the feature when the part is manufactured.

The research on using machining features in CAPP systems has been discussed earlier. Feature technology has been extended to various manufacturing application domains besides the CAPP area, such as assembly, manufacturability evaluation, inspection, and cost analysis. Gupta and Nau [27] describe an approach for the analysis of the manufacturability of machined parts. The manufacturability rating is calculated when different operation plans are generated for the given part. Wang and Kim [28] use the form feature representation of the components to identify assembly mating relations between a set of assembly components. Han et al. [29]

research on manufacturing cost optimization. The system recognizes only manufacturable features by consulting the tool database, and simultaneously constructs dependencies among the features. An optimal machining sequence is found by the aid of the feature dependencies and a manufacturing cost function.

2.2.3 Feature representation

The manufacturing feature definition must contain the appropriate information to drive process planning and NC part programming application. Features are associated with the geometry and topology of a solid model and its representation relies heavily on the capabilities of solid modeling. There are two main methods to represent features, surface representation and volume representation. [3] Surface representation uses a collection of faces in the solid model to represent features. Machining features are represented as surface features in early work due to the limitations of the solid modeling systems at that time. Surface representation provides a natural way to associate important manufacturing information such as tolerances and surface roughness with the features. Sometimes, however, a feature cannot be fully represented by the existing faces from the solid model because feature interactions may change their topology and geometry and lead to information loss. Figure 2.6 shows a slot feature intersecting with a hole feature. The bottom face and side faces of the slot are divided by the hole feature and partial faces of the bottom and side are missing.

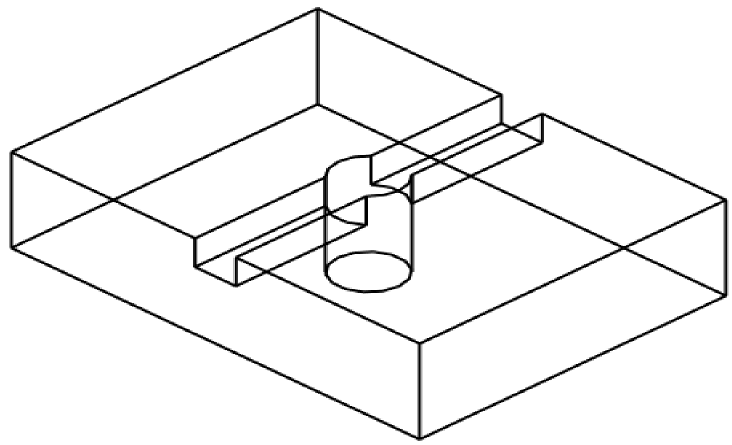
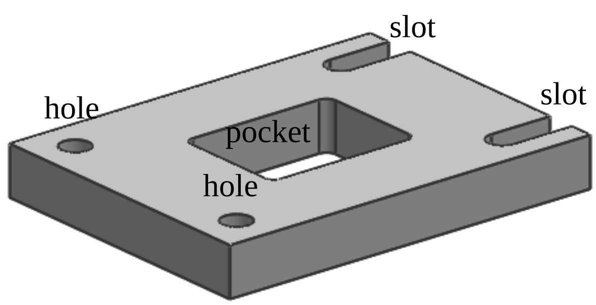


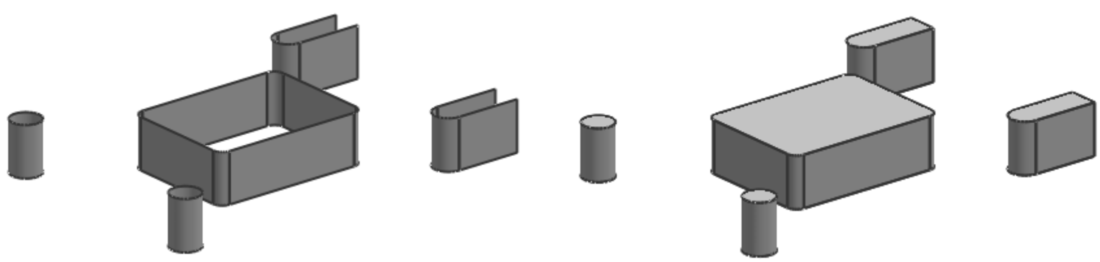
Figure 2.6 An example part with intersecting features

Volumetric representation represents features by using solid volumes that can be removed from the work piece in a machining operation. It is the basis of the volumetric decomposition feature recognition approach. There has been an increased use of volumetric representation in recent research as it provides a more comprehensive representation of the actual machining operations than surface representation. [30] Volumetric representations have great advantage in handling intersecting features. However, the pure volumetric representation is not suited to handle design and process changes and it cannot be clearly related to the associated faces of the design model. [3] Figure 2.7 illustrates examples of surface and volumetric representations of the same part.

Each type of representation has its advantages under different situations. A system would benefit from the use of feature representations having a hybrid nature. The hybrid representation would convey two groups of information, the removed material volume and the connection with the associated face entities in the original input models. FBMach has been developed at Honeywell Federal Manufacturing and Technologies. [31] The system recognizes features by hints and creates both surface and volumetric machining features.



a. solid model and features



b. surface representations

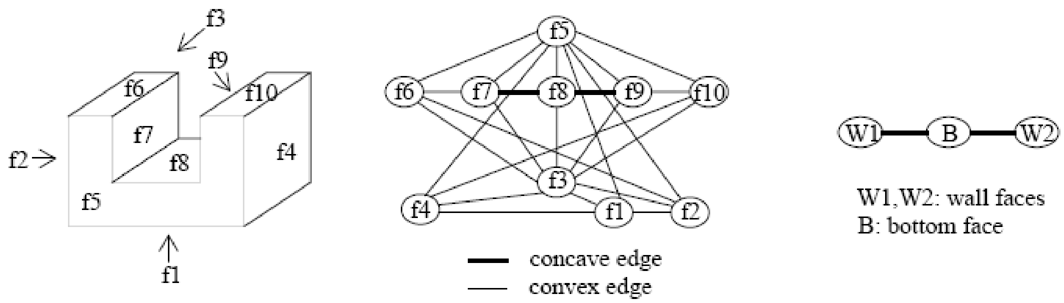
c. volumetric representations

Figure 2.7 Surface and volumetric representations of features

2.2.4 Feature recognition

Feature recognition is defined as deriving a feature model from a given geometry model. It can be performed either interactively by the user, or automatically by the software. Interactive feature recognition is very flexible since the user can explicitly define the feature. To integrate CAD/CAM systems seamlessly, however, automatic feature recognition is an important tool. So far, a great deal of research work has been performed on the automatic feature recognition from CAD solid models. Reviews of the state-of-the-art in feature recognition research are given in [22][32]. There are three currently dominant approaches, including the graph based approach, volume decomposition approach, and hint based reasoning.

The graph based approach makes use of a graph structure generated from the B-Rep of the solid model to represent the geometric and topological information of a part. Joshi and Chang [33] represent a part using the attributed adjacency graphs (AAG) where the nodes represent the faces of the part and the arcs represent the edges of the part. Nodes and arcs in the graph may have attributes indicating the convexity of the edges or the orientation of the faces. This graph representation is then searched for particular patterns that match with the feature template graphs to identify the features. Other graph representations, for example, Chuang and Henderson, [34] represent the object with a graph whose nodes are vertices of the object and whose arcs correspond to its edges. An example of graph based approach of a part is shown in Figure 2.8. [22] The part graph shown in figure 2.8.b is searched for subgraphs that match feature templates. Faces (f7, f8, f9) will be matched with the slot template as shown in figure 2.8.c. A primary problem with the graph based approach is the difficulty of recognizing intersecting features because intersections may damage the relationships between edges and faces beyond recognition. [22]



a. A part with a slot b. Graph representation of the c. Slot template

Figure 2.8 An example of graph based feature recognition [22]

The volume decomposition approach decomposes the input model into a set of primitive volumes and then manipulates them to specific features. The approach is based on the idea of finding the materials that must be removed from a raw stock to produce a part. Woo [35] uses convex hull and sets different operations to find general depression and protrusion features on a part. However, decomposition in his approach may not necessarily converge. Waco and Kim [36] propose the method of Alternating Sum of Volumes with Partitioning (ASVP) decomposition to avoid the non-convergence problem. The volume decomposition approach appears to be applicable to material removal operations in the machining domain. The problems associated with this approach are its restriction to polyhedral parts and the computational and representational limitations. Recently, Sundararajan and Wright [37] emphasize volumetric feature recognition for parts with freeform surfaces.

In the hint based approach, features are built from hints by searching the boundary representation of the part. Only hints, not full-fledged features, are found at first and then the missing portions of the feature are reconstructed. A hint implies the potential existence of a feature in the part. [38] For example, a cylindrical face can be the hint for a hole and two opposing parallel planar faces can be the hint for a slot as shown in figure 2.9. Han, et al. [39] design and implement a hint based reasoning system, called Integrated Incremental Feature Finder (IF^2). IF^2 recognizes holes, slots and pockets including floorless pockets. The system detects all hints at a time and assigns a heuristic strength to each hint. The hints are ranked in order of decreasing strength. The ranked hints are processed in a feature completer to generate volumetric features that satisfy machinability requirements. Hint based feature recognition is quite promising for recognizing intersecting features, but it is limited to predefined features and hints.

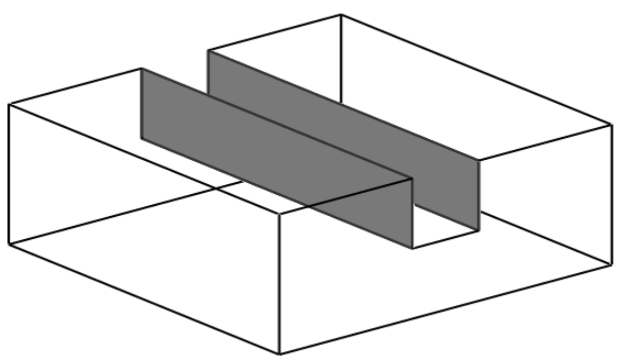


Figure 2.9 A slot hint indicated by the highlighted faces

Current research is looking at various hybrid methods combining basic recognition methods to improve the accuracy of feature recognition. Ye, et al. [40] propose a hybrid method taking advantage of graph based and hint based approaches to recognize undercut features from molded parts. Various undercuts are defined by corresponding undercut subgraphs. Face properties and parting lines are used as hints to recognize undercut features by searching for the cut-sets of the undercut subgraphs. Wang [41] combines the graph based and volume decomposition approaches for machining features. In the hybrid method, each technique recognizes features and successively simplifies the part model for the following methods. All recognized features are combined into a unified hierarchical feature representation, which captures feature interaction information.

Most important issues in automatic feature recognition have been the capability of recognizing intersecting features and handling multiple interpretations. [22] Many existing systems have limited capability to identify and account for feature intersections. Among the current approaches, the hint-based approach has demonstrated the most promise in dealing with the feature intersections. For machined parts, it is quite often that a part has more than one valid interpretation. Earlier research in feature recognition and process planning focused on generation of a single interpretation. Recently, generation of alternative interpretations has received a great deal of attention. Gupta, et al. [42] addressed the issue of multiple feature interpretations and introduced a method to generate all promising manufacturing plans from primary features.

2.3 Information sharing and transferring

Product data sharing and transferring across different systems is an important part to CAD/CAM integration. It is essential to the productivity of companies because efficient design and manufacturing require the coordination of many different processes and participants that rely on the efficient exchange of product data. In order to exchange product information we need not only the data representing the information but also the rules to interpret the data.

2.3.1 EXPRESS-G for Information modeling

In this thesis, the EXPRESS-G language is used to graphically present the information models of the system. An information model is a formal description of types of ideas, facts, and processes, and provides an explicit set of interpretation rules. [43] The information model specifies the objects within the domain, the relationships between the objects, the basic attributes of the objects and the constraints upon the objects and their relationships. EXPRESS-G is the graphical representation of an information model written in the EXPRESS lexical language. It provides a subset of the lexical modeling capabilities as it does not include the constraint portions of the lexical language. EXPRESS is an object-flavored information modeling language to specify the product information throughout its lifecycle. [43] It is a way to describe the information model and is computer interpretable. EXPRESS is defined and used in STEP as the formal specification of the required data and its relationships. EXPRESS is a language that was designed for domain experts rather than computer experts. Its graphical representation is great to display the structure of the model, and the associations between the items in the model are easy to follow by following the lines.

The EXPRESS language supports the definition of entities, attributes, basic types, inheritance relationships, properties, relationships, and constraints. The basic constructs of EXPRESS are entities and attributes, which represent relationships as well. Entities are identified from concepts in the interested domains. An entity is the basic object of the information model. It helps group the information in an intuitive way. An entity has attributes, describing the characteristics of the object. Attributes can be defined as either simple data types, e.g. integer, string or more complex types, e.g. other entities.

In EXPRESS-G a solid rectangle with a double vertical line at its right end represents a simple data type. There are seven predefined simple types in EXPRESS, namely binary, boolean, integer, logical, number, real and string. Solid rectangular boxes represent entities being defined in the information model. Lines with an open circle are used to show the relationships between entities or between an entity and its attributes. The lines are labeled with the name of the attribute with any cardinality constraints. Three different line styles are used in EXPRESS-G, dashed, thick and normal lines. An inheritance relationship is displayed as a thick solid line. An optional attribute of an entity is displayed as a dashed line. All other relationships are displayed as normal width solid lines. An EXPRESS-G model can reside on more than one page. If a relationship occurs between definitions on separate pages, the relationship line on each of the two pages is ended by a rounded box, which contains the page number, the reference number and the entity name referred to. The notion of EXPRESS-G is illustrated in figure 2.10.

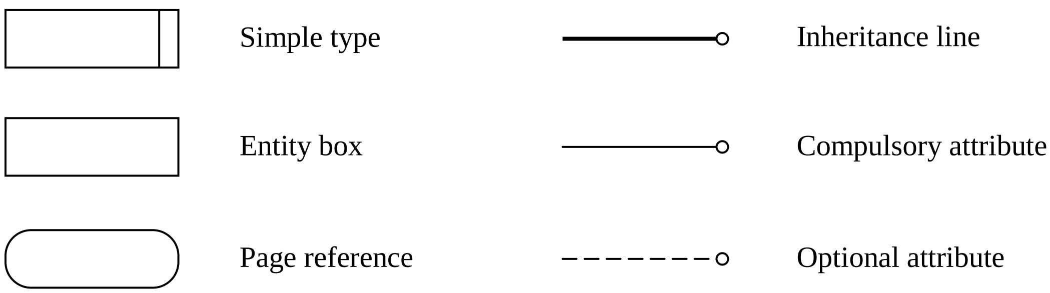
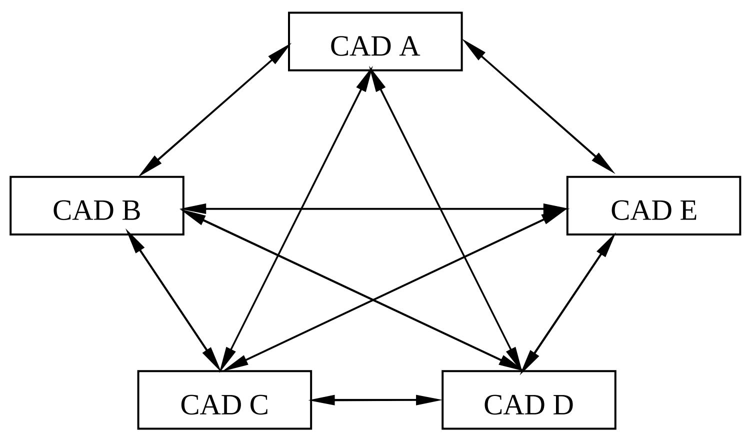


Figure 2.10 The notations of EXPRESS-G language

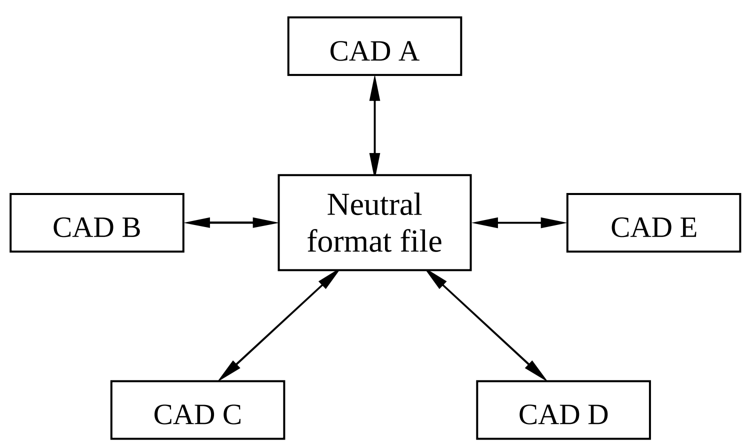
2.3.2 Product data transfer

There are numerous file formats on the market from different CAD/CAM systems, such as IGES, DXF, SAT, and PAR, to name a few. To integrate heterogeneous CAD/CAM systems, product data exchange and sharing between different systems is essential so that different CAD/CAM systems can communicate with each other. There are mainly two types of data translators: direct translators and neutral translators. [45]

Direct translators exchange data directly from one CAD/CAM system's proprietary format to another. Using direct translators requires a two-way translator for every two systems. The number of translators needed is $\frac{N!}{2!(N-2)!}$ to transfer data among N different CAD/CAM systems. Figure 2.11.a shows the situation using direct translators. A direct translator between two different systems normally works well and converts the product data to representation understandable by the receiving system. However, the implementations of direct translators are expensive and the number of translators needed increases exponentially when the number of involved



a. situation using direct translators



b. situation using neutral translators

Figure 2.11 Direct and neutral data transfer

CAD systems increases.

Neutral translators convert a proprietary CAD or CAM data format into an industry standard format that is then read by another CAD/CAM system and converted into its own format. The task of translating between various systems is reduced in complexity using neutral translators. Instead of defining a translator between every two systems, the translators only need to be defined between each system and the standard format. The number of neutral translators needed to transfer product data among N different CAD/CAM systems is just N. The industry standards are documented and available for anyone to use. The two primary neutral standard formats used today are IGES (Initial Graphics Exchange Specification) and STEP (Standard for the Exchange of Product Model Data) for mechanical product data exchange.

IGES (Initial Graphics Exchange Specification)

IGES is an industrial standard format for transferring CAD data to a dissimilar system. It was originally developed for the US Air Force, and was adopted as a national standard of the United States in 1980. [46] The IGES standard is a neutral file format that includes geometry including points, curves, surfaces and solids, and non-geometry information including annotation, definition and organization information. IGES was the first specification for CAD data exchange and today is supported by almost all CAD vendors. IGES was the dominating standard for CAD data exchange at that time.

IGES provides limited support for different data types and applications. Product data includes a very wide range of data types, not just CAD geometry. This limits the effective use of IGES except in relatively simple cases due to the large file size. Another big criticism that is made of IGES is that there is more than one way to describe some entities. [46] For instance, a cubic spline may be presented as IGES entity 112 or entity126 or even as a polyline of points (entity 106). The industrial community has recognized that a standard for CAD data exchange requires worldwide acceptance. Especially in the area of solid model exchange the IGES approach was recognized as excessively unsuitable. Hence, a completely new approach was needed.

STEP (Standard for the Exchange of Product Model Data)

STEP has been under development as an international data exchange standard since the 1990s. It became a full ISO (International Standards Organization) standard in 1994 and by now every major CAD system vendor has implemented STEP data translation. It is estimated that more than two million CAD stations now contain STEP data translators. [47] STEP overcomes the shortcoming of IGES and gives an explicit and complete representation of the product data model. STEP was designed to support a very wide variety of functional and business requirements. It contains the product data covering the entire product life cycle and has a neutral format that is independent of any software package and unrestricted to any particular hardware platform.

STEP is being maintained and extended by an international team of more than 200 experts who meet three times each year to design extensions and add technology. [47] It defines a data standard for CAD product design data including geometry, topology, tolerances, relationships, attributes, assemblies, configuration, and more. The STEP standard is intended for long-term development with modules or application protocols (APs) released as needed. The basic parts are complete and published, while more are under development. The completed parts cover general areas, such as testing procedures, file formats and programming interfaces, as well as industry-specific information. STEP has gained considerable popularity since the late 1990s, mainly due to the active support for STEP from the automotive and aerospace industries. According to a study of product data exchange in the automotive industry, the cost of interoperability problems involved in the automotive supply chain was estimated at about 1-billon dollars each year in the U.S., and the estimate was considered conservative. [48]

The most important aspect of STEP is extensibility. STEP already contains definitions for geometry, product identification, product structure (assembly), configuration control and manufacturing features. In recent years, work has been done to include information for tooling, manufacturing strategies, manufacturing processes and maintenance. The geometry model satisfies the requirements of the computer-based representation of the shape of a specific product, but it is unable to describe non-geometric product information. The integrated model is used to support the product development in the full life cycle, from product requirement analysis, conceptual design, detail design, process planning, CNC programming, machining, and assembly to quality assessment.

2.3.3 STEP-NC

The emergence of STEP-NC cannot be ignored when talking about an information model for better CAD/CAM information sharing, although it has not had any commercial usage. STEP-NC, the manufacturing extension of STEP, annotates the design information with manufacturing data. [4] It is being developed through international effort, following the success of the international standard STEP, to provide a data model for CNC machines. Currently, data models for basic milling, drilling and turning operations have been developed and interface schemes for other processes, such as contouring and wire EDM, are under development.

STEP-NC provides direct input for CNC machine tools, consisting of product information such as geometry, features, machining steps and tool paths. [49] Currently CNC machines have to be programmed using G-codes, which only describe the exact tool movements, without any information of the part being processed. Unlike G-codes, STEP-NC tells machines what to do instead of how to do it. Figure 2.12 contains an illustration of the information defined by STEP-NC in EXPRESS-G. [50] STEP-NC specifies machining processes using the concept of working steps, which describe a sequence of material removals, location, and associated process parameters. Each working step is related to an operation and a machining feature. CNC controllers are expected to translate working steps to axis motions and tool movements to machine the part.

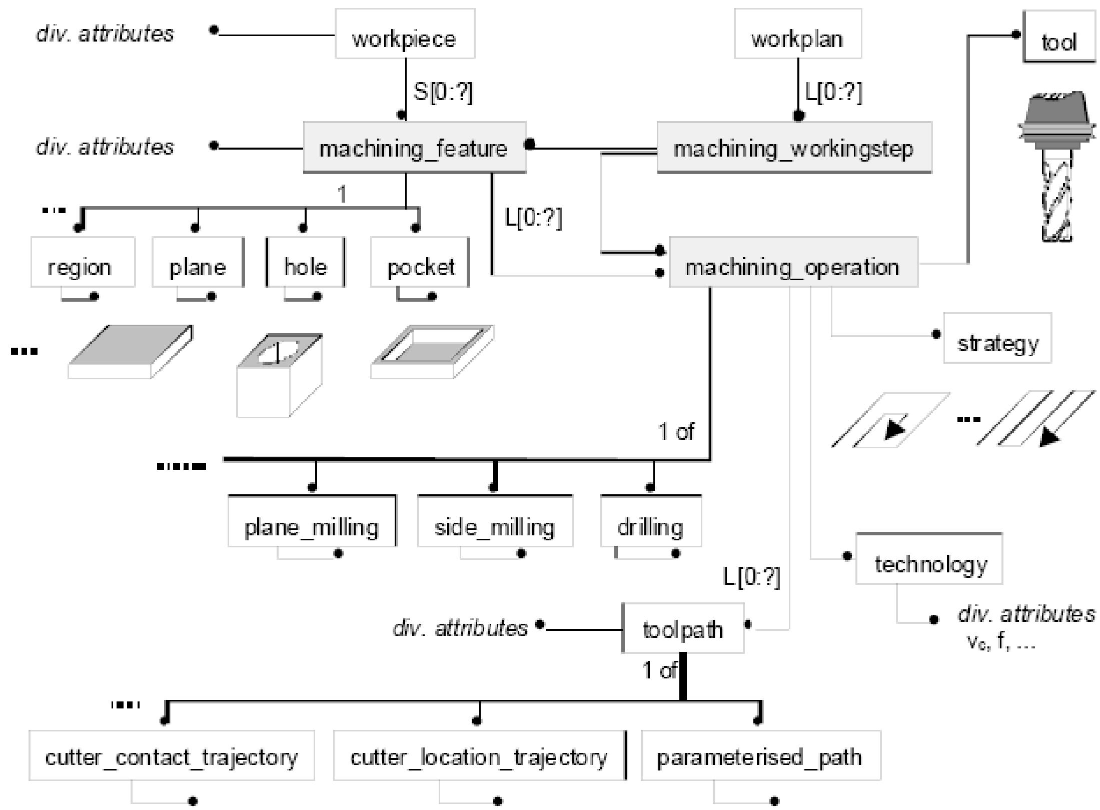


Figure 2.12 Information Diagram of STEP-NC [50]

The researchers have shown great interests in STEP-NC and its deployment towards CAD/CAM integration. [49]-[55] With the use of full fidelity STEP-NC data as direct input to a CNC machine, many bottlenecks to manufacturing productivity will be eliminated, such as cumbersome post processors, antiquated G & M codes, data redundancies, and more. [54] Today, CNC machines are not given any information about the part and process. With STEP-NC the controllers can be given complete information about the part, 3D model annotated with design tolerances, manufacturing features, process sequence and cutting tool requirements. STEP-NC will allow machine tools to deploy alternative strategies for making a part with the information given. Post processors will be embedded in the intelligent controllers so they will be transparent to users.

Although STEP-NC is appealing and gathering more and more interest from researchers, there are some challenges to overcome before it can be realized. [54] Certain capabilities of CAM systems are moved to NC controllers by adopting STEP-NC. It results in much more complicated controllers and requires more knowledge from the machine operators. Before STEP-NC can be practically used in manufacturing new STEP-NC compliant controllers for NC machines have to be developed first. NC controller manufacturers then have to redesign the structures and strategies of their controllers to make it happen. There are two versions of STEP-NC being developed concurrently by ISO, ARM (i.e. ISO 14649) and AIM (i.e. ISO 10303 AP-238). The main difference between these two models is the degree to which they use the STEP representation methods and technical architecture. Both versions can be viewed as different implementation methods of STEP-NC. More information about the two standards can be found in Feeny, et. al [55]. Industry is still debating on whether to use AIM or ARM. Furthermore, many inconsistencies among standards remain to be resolved to standardize the STEP-NC data model. Therefore, a large amount of systems-building work is still needed to make STEP-NC related technology commercially viable.

3 Integration of FBMach and Unigraphics

3.1 Architecture of the integrated system

As discussed earlier the current method to generate tool paths and NC programs in CAM systems is a lengthy process and needs extensive user interactions. The process planning information generated in a CAPP system cannot be used in CAM system directly. This research utilizes machining features to transfer geometry information from the CAPP system to the CAM system and makes information intermediate available to the CAM system. A prototype has been implemented in conjunction with FBMach and Unigraphics to realize the integration of CAD/CAM systems based on machining features and automate the process of tool path generation. The integration of feature-based CAD/CAM will allow manufacturing automation by enabling automatic setup selection, automatic processes planning, tool selection and tool parameter selection, and automatic sequencing of operations.

FBMach, a feature-based process planner, is used as the CAPP system for the integration. FBMach utilizes hint-based automatic feature recognition technology to recognize features and create process plans from CAD solid models. FBMach defines micro planning level processes and associated resources, like cutting tools and machining parameters to implement process plans. UG, a commercial CAD/CAM software package, is used as the CAM system for the integration. It is utilized for its capability of tool path generation and post processing. Moreover, UG/OPEN [56] provides an application program interface (API) to allow user-developed programs to access and manipulate the geometry and manufacturing information of a model, which allows the integration of user developed systems with UG. The interface of FBMach and UG is developed to connect the two systems. All the information coming from the CAPP system (FBMach) is understandable in the CAM system (UG) through the interface, therefore the user interaction required to generate tool paths is significantly decreased.

The architecture of the integrated CAD/CAPP/CAM system is illustrated in figure 3.1. A part is designed in any CAD system and saved as a STEP file. FBMach reads in the solid model in STEP, defines the stock, then automatically recognizes machining features and generates the process plan. All the information of the part, the stock, features and process plan are read in and processed by the interface of FBMach and UG. The interface imports solid models of the part and stock, retrieves the

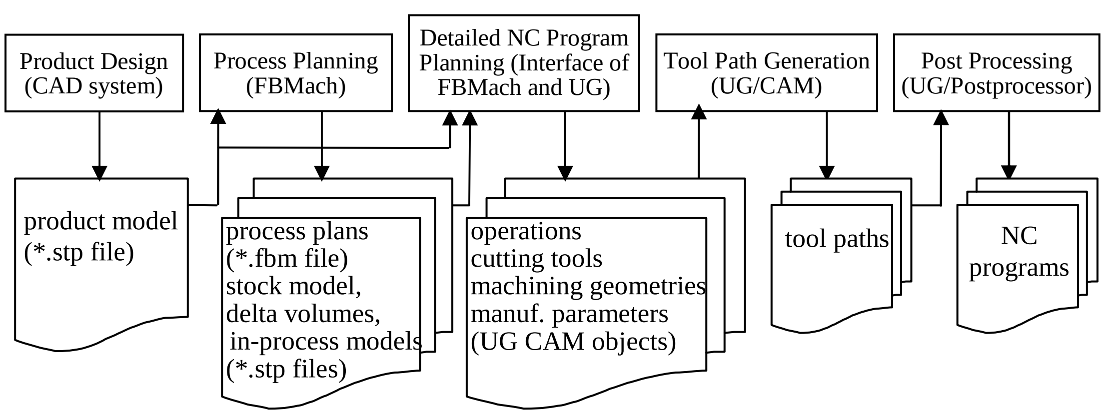


Figure 3.1 The architecture of the integrated CAD/CAM system

information of features and the process plan, creates corresponding operations for cut actions, finds machining geometries from machining features, specifies cutting tools and defines all the machining parameters available in the process plan for the operations. Then tool paths can be generated in UG according to the automatically created operations and the associated parameters. After tool paths are simulated and verified the NC programs can be created through post-processing and fed into NC machines. In the integrated system the process of tool path generation from solid models is automated with no user interaction necessary.

3.2 Process planning information from FBMach

To automate the process of tool path generation in UG, product and process planning information are needed from FBMach. FBMach takes advantage of existing solid model geometry to automate the process of defining features and identifying how each feature should be removed. FBMach defines the stock, recognizes machining features, defines cut actions for each machining feature, and determines the sequence of cut actions, the associated cutting tool and machining parameters for cut actions, all of which comprise a process plan file.

3.2.1 Material removal features and NC features

FBMach uses hybrid representation and defines both surface features, called material removal features, and volumetric features, called Numerical Control (NC) features. A material removal feature consists of a combination of faces and/or edges on the part to which geometrical and technical information is associated, which represents the feature on the final part, whereas a NC feature is a volumetric feature that represents the material volume to be removed from the stock or an intermediate work piece to produce a material removal features in a certain cut. For instance, if there is a rectangular pocket, the piece of material removed will be shaped like a block. This block would be an example of a NC feature and is called a delta volume. In some cases, a cut will produce more than one disjoint piece of material. Therefore, a NC feature may consist of one or several delta volumes. The delta volume contains only stock faces to be removed and part faces to be generated, whereas the part model may contain faces that are not machined and therefore are of no interest. [57] The in-process status of the part is described at different stages of the machining process through the use of in-process models. In-process models show the exact shape of the part after a given material removal operation. See figure 3.2 for examples of in-process models and delta volumes of a part. [5] The bodies in the left and right sides are delta volumes that show the material to be removed and the bodies in the center are in-process models that show the intermediate stages of the part at each step. The first body in the center represents the initial stock and the last body in the center represents the final part.

3.2.2 Features dealt with in the research

The purpose of machining features in the research is to facilitate the identification of machining shapes that are human and computer interpretable.

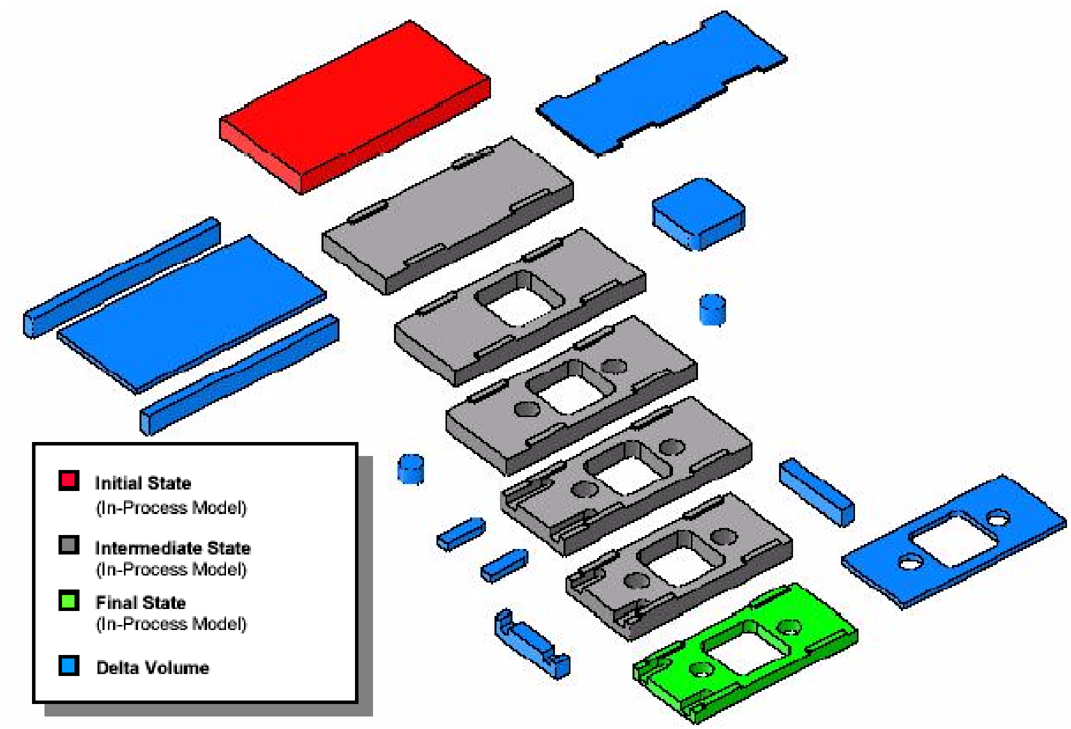


Figure 3.2 In-process models and delta volumes in FBMach [5]

Machining features allow information about the part shape to be used for decision-making in process planning and transferred to a CAM system along with process information. Feature type, feature dimension, and feature orientation are needed for tool selection, operation creation and sequencing. Some key common attributes of a feature include geometry of the feature, type of the feature, location and orientation of the feature. This research is limited to prismatic machining features and associated 2.5-axis milling and drilling operations. Machining features explored in the research are classified into round hole, pocket, slot, step, periphery, cutout, planar face, and general removal features. The description of each feature is given in table 3.1.

Table 3.1 Description of machining features dealt with in the research

Feature Type	Feature Description
Round hole feature	An axis-symmetric depression in the work piece where the side faces are represented by cylindrical/conical surfaces.
Planar face feature	An unbounded plane of the work piece. There is no any side faces found in the planar face feature.
Periphery feature	A feature that defines a portion of, or the entire periphery of the work piece. It does not have bottom faces and only consists of some side faces.
Cutout feature	A hole of arbitrary shape on the work piece with only side faces and no bottom faces.
Slot feature	A depression on the work piece that is characterized as a channel that can be created by sweeping a symmetric silhouette along a path. It may be closed on one or both sides and cannot have islands.
Step feature	A depression on the work piece where bottom faces share the same planar surface definition and side faces are contiguous. All side and bottom faces are perpendicular, and the corners between each side face are convex.
Pocket feature	A depression on the work piece that has one opening. A pocket will have only one bottom surface and have minimum three side faces.
General removal feature	A depression on the work piece that is in general shape and has side openings. It can have arbitrarily complex profile with a planar bottom face.

3.2.3 Exported process plan files

FBMach defines machining features, cut actions for machining features, the sequence of cut actions, associated cutting tools and machining parameters of cut actions. All of which are exported to process plan files. The portion of the exported file for a hole removal feature is listed in table 3.2. For round hole features, besides the common attributes of machining features, some unique attributes of holes features, such as bottom condition, hole depth, hole diameter, and taper angle, are exported into the process plan file as well.

Table 3.2 The portion of the process plan for hole removal feature

```
START_INSTANCE
MATL_REM_FEATURE_KCP "Hole-01"
type = ROUND_HOLE_MATL_REM_KCP
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
bottom_condition = "through"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.09
max_diameter = 0.161
taper_angle = 0
END_INSTANCE
```

For each recognized feature a set of feature cuts will be created and associated with the feature. A feature cut defines one cut for a material removal feature. For example, a round hole feature may have a center-drill cut, a drill cut, a ream cut, and a bore cut associated with it. A feature cut action is a step in the process plan that corresponds with a feature cut. A feature cut action has information of cutting tool, material removal feature through associated feature cut, NC feature, and some machining data as listed in table 3.3. A delta volume and in-process model are attached to the NC feature and given in STEP format as shown in table 3.4.

Table 3.3 The portion of a process plan for feature cut and feature cut action

```
START_INSTANCE
FEATURE_CUT_KCP "Hole-01:Micro-1:Drill-1"
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_ACTION_KCP "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
description = "Drill 0.161 in. Dia. Hole-01 to 0.141 in. Dia. and a Depth of 0.340 in."
FEATURE_CUT_KCP = "Hole-01:Micro-1:Drill-1"
working_step_offset_type =drilling_workingstep
MANUF_TOOL_RESOURCE_KCP "tl302"
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Drill-2:Hole-01:Micro-1:Drill-
1:Cut-1:1"
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Drill-2:Hole-01:Micro-
1:Drill-1:Cut-1:1"
MANUF_DATA_RESOURCE_KCP = "manuf_data:Drill-2:Hole-01:Micro-1:Drill-
1:Cut-1:1"
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Drill-2:Hole-01:Micro-
1:Drill-1:Cut-1:1"
NC_FEATURE_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1:NC Hole-3"
END_INSTANCE
```

Table 3.4 The portion of the process plan for NC feature

```
START_INSTANCE
NC_FEATURE_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1:NC Hole-3"
type = NC_ROUND_HOLE_KCP
FEATURE_CUT_ACTION_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
description = "flat"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.16
max_diameter = 0.141
taper_angle = 0
delta_volume = DV1_Hole-01Drill-1.stp
in_process_model = IP1_Hole-01Drill-1.stp
END_INSTANCE
```

A complete file of an exported process plan from FBMach is listed in Appendix A. Only the information related to the hole feature is included in the process plan file. All machining features of the part except one hole feature are suppressed and all feature cut actions except the feature cut actions for the hole are suppressed as well.

3.3 Tool path generation in Unigraphics

UG supports a wide range of manufacturing processes, including machining operations planning, NC programming, post processing, and NC program verification. An advantage of UG is UG/OPEN that allows user-developed programs to access and manipulate the geometry and manufacturing information of a model. It gives users the flexibility to create user-developed systems in UG and makes the integration with UG possible.

UG Manufacturing module allows users to interactively program and post process milling, drilling, turning and wire EDM (Electrical Discharge Machine) tool paths. The supported milling operations include face milling, planar milling, cavity milling, 3 axis contour milling and multi-axis milling. UG has its own file format of geometric models, but can successfully import a STEP file through translators. UG newly added holemaking module provides semi automatic drilling of hole features. [6] Users need to rely on user defined rules, attributed geometry, and CAM Templates to define the manufacturing process. The extent that UG uses features for NC machining is very limited.

3.3.1 Process of tool path generation in CAM

To interactively generate tool paths of a part in UG, many steps and user actions are involved. The overall process of creating any operation is basically the same, although some of the parameters for milling, drilling, turning, and wire EDM operations differ. Before creating any operation, the user needs to determine the sequence for machining the part based on experience, and then create operations representing the machining sequence. To describe the process of creating an operation and its NC programs, a standard drill operation is used as an example. The steps to generate NC programs for a standard drill operation are listed below:

1. Open an existing part or create a new part.
2. Initialize Machining Environment by selecting CAM Session Configuration and CAM Setup.
3. Define a standard drilling operation. Choose drill from type menu, choose drilling icon for subtype, and specify the operation name, geometry group, cutting tool, program group, and milling method if necessary.
4. Select the drill geometry such as selecting points or holes, optimizing their order of machining, and avoiding obstacles.
5. Specify a drilling tool or retrieve a drilling tool from a tool library.
6. Specify cycle types and set the parameters for the cycle.
7. Set any additional machining parameters necessary for the operation, such as, tool axis, avoidance geometry, machine control etc.
8. Generate tool paths for the operation. After generation, the tool paths can be displayed, listed, or deleted.
9. Post process the tool paths to generate NC programs for the operation.

All the steps can be done through the graphic user interface as shown in figure 3.3 and currently need to be done by the user manually. The process to manually

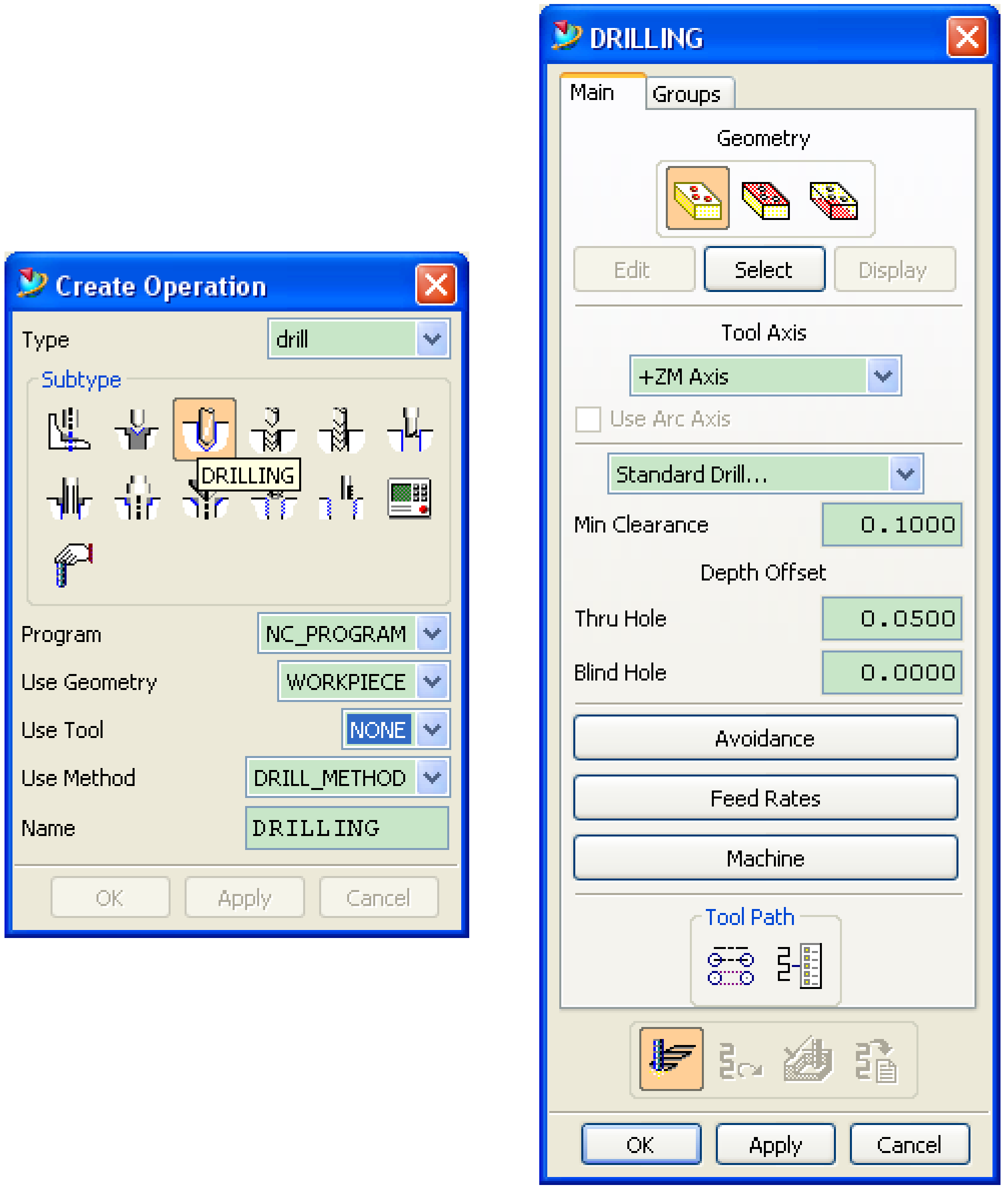


Figure 3.3 User input requirements for a drilling operation

create an operation and specify all necessary parameters is a time consuming and error prone task, especially preparing the geometry for in-process status and specifying the variable machining parameters. It heavily depends on the knowledge and expertise of the manufacturing engineer.

3.3.2 Machining operations

The machining operations available in UG are classified into drilling, milling, turning, and wire EDM operations. Drilling operations are machining processes for hole features, such as spot drilling, standard drilling, boring, reaming, counterboring, countersinking, and tapping. Milling operations are machining processes for milling features, which can be further classified into planar milling, cavity milling, contour milling and multi-axis milling operations. For prismatic machining features only 2.5 axis operations, drilling, planar milling, and cavity milling, are involved in this research.

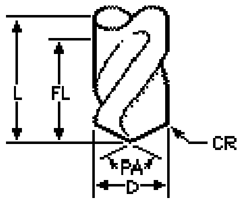
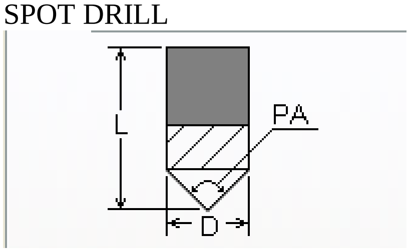
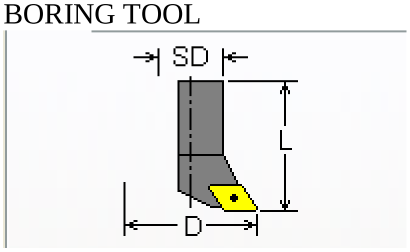
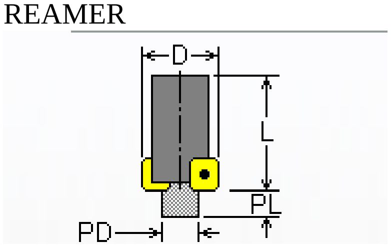
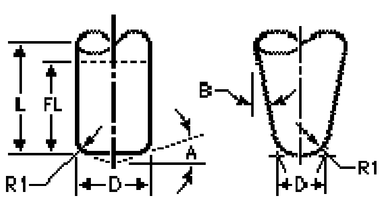
The spot drilling operation is used to provide starting holes for other drilling operations. The drill operation is the primary hole making operation used to drill basic holes. The boring operation is to enlarge a previously drilled hole with a single-point tool and produces a close tolerance and fine finish. The reaming operation is used to smooth and accurately size a previously drilled hole with a reamer.

The planar milling operation creates tool paths that remove material in planar layers by cutting levels perpendicular to the tool axis. Planar milling uses boundaries to define part geometry. It is intended for parts that have vertical walls and planar islands and whose floors are normal to the tool axis. Planar milling can also perform single and multiple pass profile machining of open and closed boundaries. Face milling is a special case of planar milling and designed specifically to rough and finish the planar faces of a part. It allows users to specify the face geometry by simply selecting the faces to be machined. The tool axis is automatically defined as the normal of the selected face plane.

Cavity milling removes material in cut levels that are perpendicular to the tool axis. It is similar to planar milling in the way that it uses a fixed tool axis and removes the material in planar cut levels. Cavity milling uses bodies, faces, or curves to define part geometry. Therefore it can be used for parts with tapered walls and contoured floors.

3.3.3 Cutting tool definition

To successfully generate tool paths, tooling information is required before calculation. The cutting tools related to the machining features are classified into two groups, drilling tools and milling tools. Drilling tools are used to machine drilling features and are further categorized into spot drill, twist drill, reamer, and boring tools. Milling tools are used to machine milling features and are categorized into face mill and end mill. End mill tools are the most common tools used in the milling process, they have cutting teeth at the end face, as well as on the periphery. Face mill tools usually have bigger sizes and more cutting teeth and most of the cutting is done at the end face of the cutting tool. The parameters to define a drilling or milling tool used in this research are listed in table 3.5.

<table>
<tr><td colspan="2">Table 3.5 Tool parameters of cutting tools in UG</td></tr>
<tr><th>Tool type</th><th>Tool parameters</th></tr>
<tr><td>TWIST DRILL

</td><td>(D) Diameter
(L) Length
(CR) Corner Radius
(PA) Point Angle
(FL) Flute Length
Number of flutes
Direction</td></tr>
<tr><td>SPOT DRILL

</td><td>(D) Diameter
(L) Length
(PA) Point Angle
Number of flutes
Direction</td></tr>
<tr><td>BORING TOOL

</td><td>(SD) Shank Diameter
(D) Diameter
(L) Length
(CR) Corner Radius
Number of flutes
Direction</td></tr>
<tr><td>REAMER

</td><td>(D) Diameter
(L) Length
(CR) Corner Radius
(PD) Pilot Diameter
(PL) Pilot Length
Number of flutes
Direction</td></tr>
<tr><td>FACE MILL/END MILL

</td><td>(D) Diameter
(R1) Lower radius
(L) Length
(B) Taper angle
(A) Tip angle
(FL) Flute length
Number of flutes
Direction</td></tr>
</table>

3.3.4 Machining geometry

Different types of operations need different machining geometry to generate tool paths. For drilling operations the required geometry is pretty simple, hole features that can be represented by cylindrical holes, conic holes, arcs or circles. The center locations of the holes or circles are extracted to generate tool paths or the center location can be given directly by points.

Planar milling operations require boundaries while face milling operations require faces. In face milling the faces define the floors to be faced off. Face geometry consists of closed boundaries whose inside material indicate the areas to be machined. In planar milling boundaries define areas that constrain cutting moves. Boundaries are used to define part, blank, and check geometry and the floor is used to define the lowest cut level for planar milling operations. All cut levels are generated parallel to the floor plane. Cavity milling operations use bodies, faces, or curves to define part and blank geometry. The difference between part and blank geometries defines the cut volume.

An example of machining geometries for cutting a part is illustrated in figure 3.4. The part may be machined using planar milling or cavity milling in UG with different geometry specifications and parameter sets. Figure 3.4.a shows the part and blank boundary and the floor for a planar milling operation. Figure 3.4.b shows the part body and blank body for a cavity milling operation.

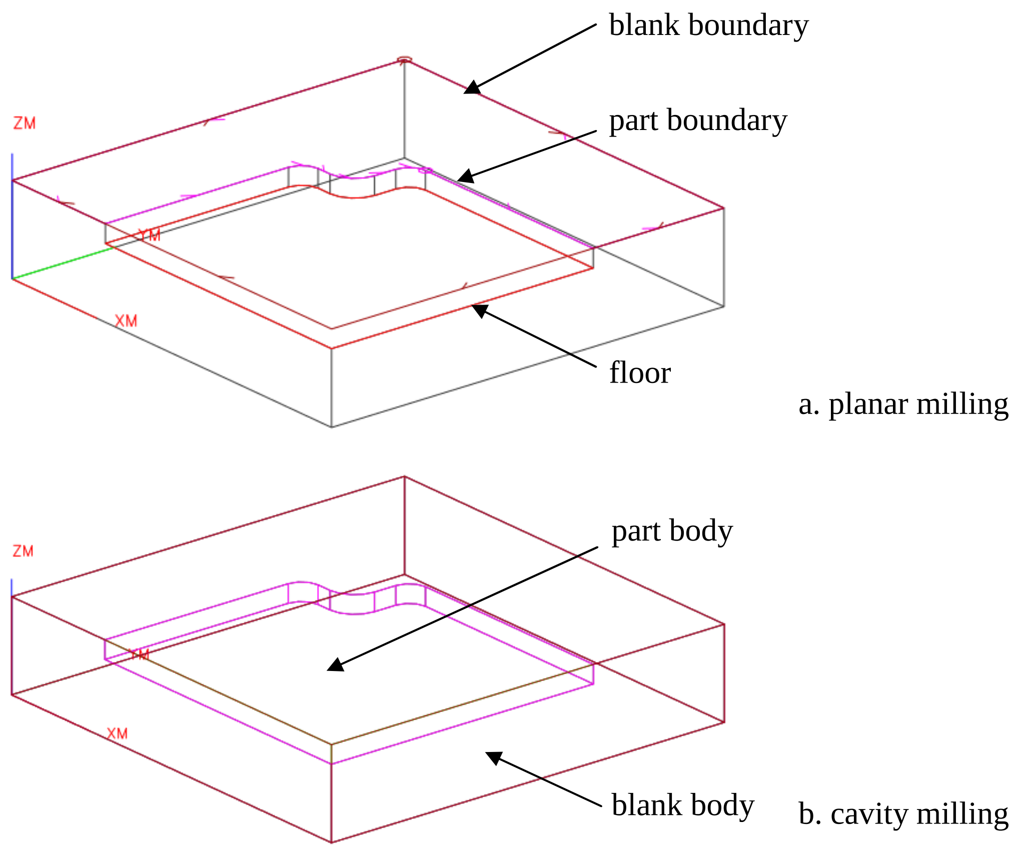


Figure 3.4 Machining geometries for planar milling and cavity milling

3.3.5 Machining parameters

A manufacturing application has its own coordinate system representing the reference position, the Machine Coordinate System (MCS). Machining operations in different set-ups needs different MCS. All of the coordinate values in the tool paths are output relative to the MCS. MCS can be defined as a group to include a list of operations in one set-up.

Some machining parameters are common to drilling and milling operations, such as engage and retract, tool axis, feed rates and speeds. Engage and Retract options are used to establish the direction and distance that the tool moves into or out of cutting movements within the tool path. Cutting feed rate and speed can be determined based on the part material, tool material, cut method and cut depth. Feed rate and speed can be retrieved from feeds and speeds database in UG once part material, tool material, cut method and cut depth are specified, or the user can manually specify the values of feed rate and speed.

Cycle parameter and minimum clearance are unique machining parameters to drilling operations. Cycle parameters define tool motions and conditions for cycle operations such as depth, feed rate, dwell times, and cutting increments. The depth and feed rate in cycle parameters have priority over those defined in the operation as common parameters. Minimum clearance distance specify the distance that cutting tool maintains above the part before beginning the cut feed rate. The cutting tool will position to each hole at the rapid or engage feed rate along the minimum clearance above the part surface.

For milling operations some frequently used machining parameters are cut method, cut depth, stepover and clearance plane. Cut method determines the tool path pattern used to machine cut areas. Available cut methods include zig-zag, zig, follow part, and profile as shown in figure 3.5. The part in the figure has a pocket with an island. Cut depth determines the cut levels of a multi-depth operation. Stepover

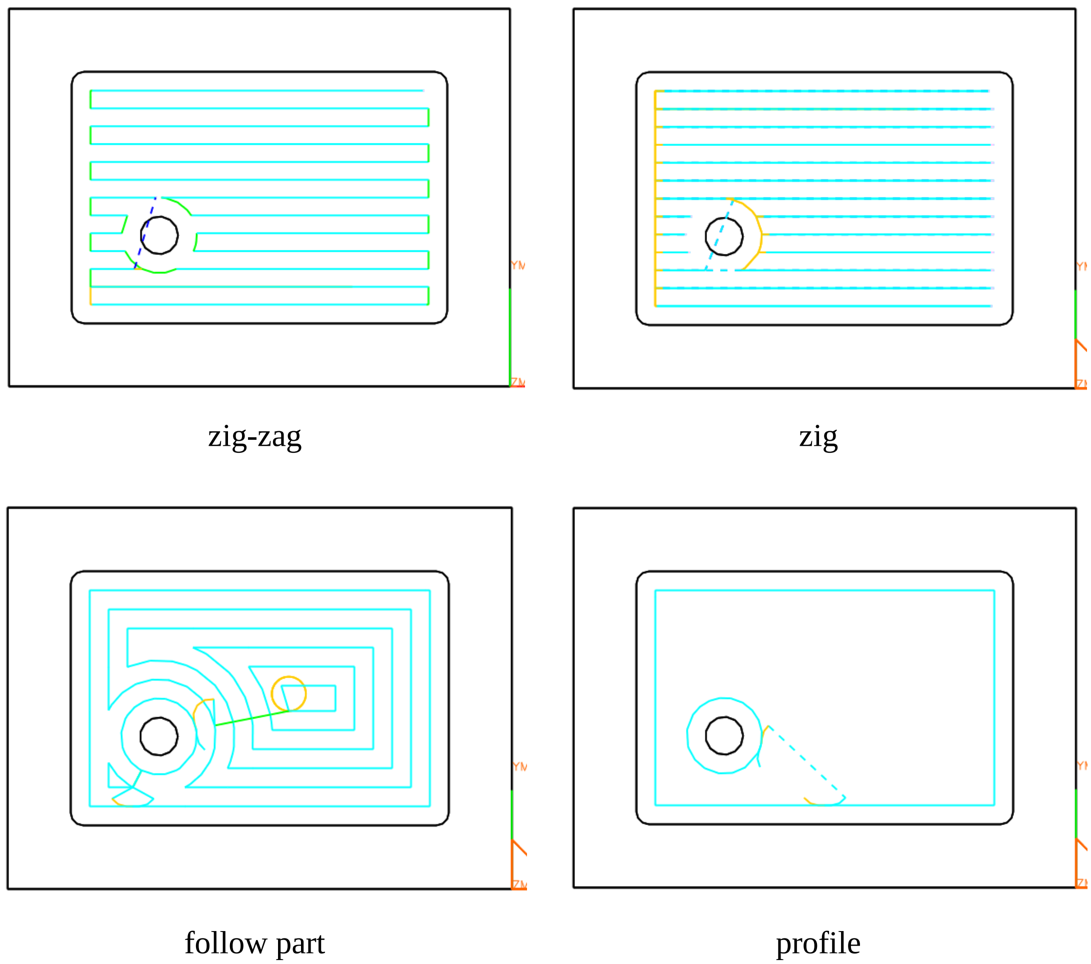


Figure 3.5 Cut methods for milling operations

specifies the distance that the cutting tool travels between successive cut passes. Clearance plane defines a safe clearance distance for tool motion before and after an operation and during any programmed obstacle avoidance moves between points. When the clearance plane is used, a rapid move to the clearance plane before the engage move is generated and the tool retracts to the clearance plane at the end of the operation.

4 Software Implementation

4.1 The prototype of integrated CAD/CAM system

The prototype of feature based CAD/CAM integration has been implemented in conjunction with FBMach and UG. An interface between FBMach and UG is developed as the integration layer to transfer and share the information between the two systems and automate the process of tool path generation. What the interface does is to read in the geometry models of the part and the stock and the process plan file of the part. The interface initializes UG CAM session and sets the appropriate machining environment to make it ready for operation creation. Then it creates machining operations according to the process plan, specifies machining geometry, defines a cutting tool, sets appropriate parameters for each operation and finally generates and displays tool paths for the operations. The flow chart of the integration layer is shown in figure 4.1. The prototype consists of three parts: retrieving information from process plan files produced by FBMach, building a class library for UG CAM objects and mapping information from FBMach to UG to prepare for tool path generation.

The interface is developed as an UG internal program to work within UG. The majority of the program is written in Visual C++ with UG/Open, while a couple of functions are written in UG GRIP (GRaphics Interactive Programming) because of the limitation of UG/Open. The UG/Open allows full access to the UG object

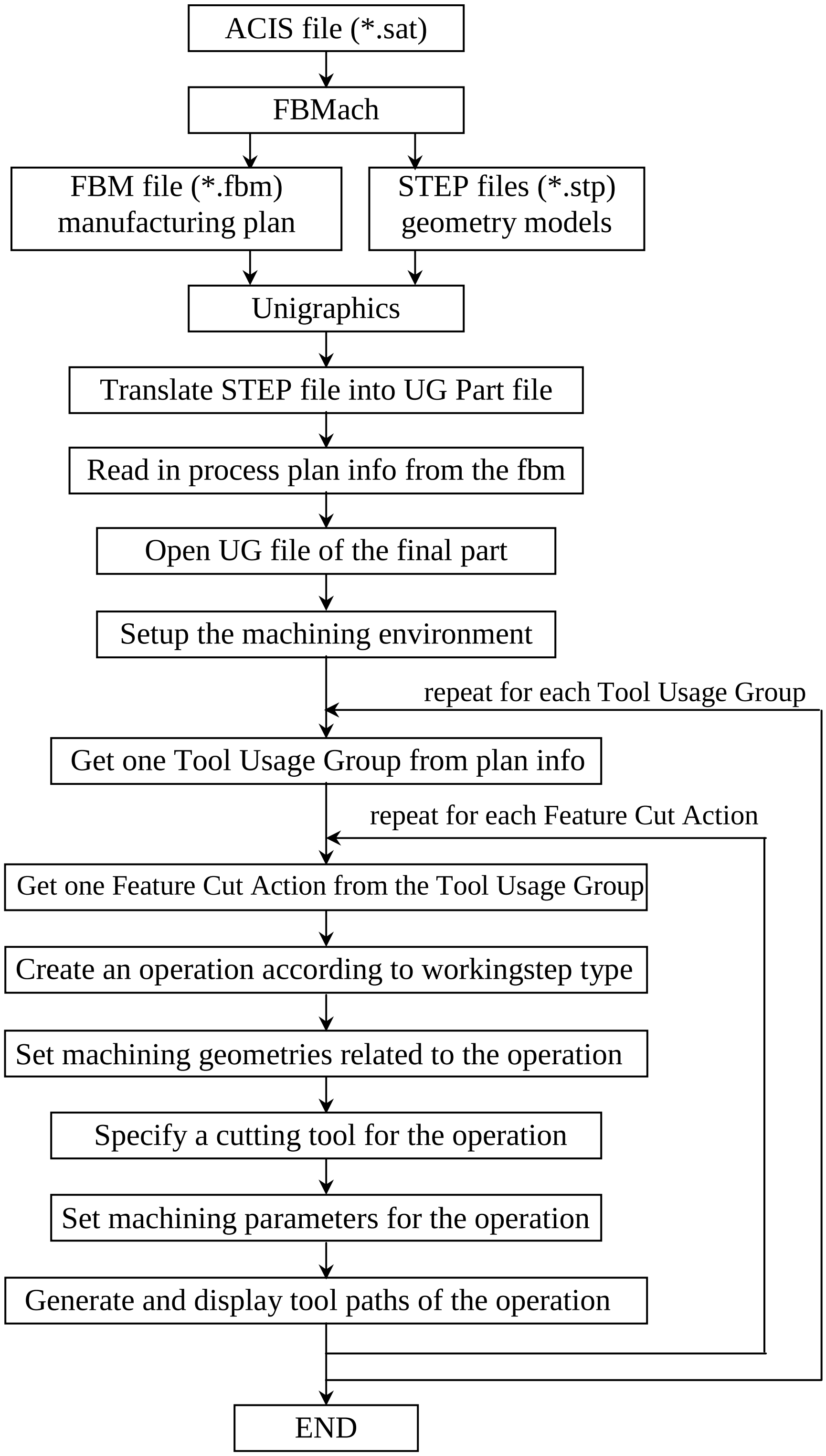


Figure 4.1 Flow chart of the integration layer between FBMach and UG

structure and permits all possibilities of classical software development. UG/Open programs can be developed to run in two different environments: external and internal, [56] depending on how the program is linked. External programs are stand alone programs that can run from the operating system independently from the UG executable. It is not necessary to start UG to run external programs. No graphical interaction is possible unless it is especially programmed in because UG is not running. On the other hand, internal programs can only be run from inside of UG. Internal programs are compiled as dynamically linked libraries and need to be loaded into the UG process space. The results of internal programs are visible in the graphics window of a UG session. For the integration of FBMach and UG, we need the graphical interface to show the solid models of the part and generated tool paths graphically. Therefore, the integration layer is developed as an internal program. Most functions needed for tool path generation can be achieved using UG/Open, but a few functions can only be realized by UG GRIP. UG GRIP is an interpretive programming language using English-like words. UG/Open provides functions that allow two-way communication between UG/Open and UG GRIP. A GRIP program can be called in UG/Open program once it is complied and linked.

4.2 Data models of process plan and CAM objects

To automate the process of tool path generation in UG, we need to determine what information is needed from FBMach and how it should be presented. The data model of process plan information and UG CAM objects are presented in this section.

Entities of the two data models are identified from concepts in CAPP and CAM domains respectively.

4.2.1 Data model of process plan information

When the process planning of a part is completed in FBMach it creates a process plan file for the part. Process plan files from FBMach are simple text files. The high level data model of the process plan information is illustrated in figure 4.2 using EXPRESS-G that has been introduced in the second chapter. To generate tool paths for the part in UG, the part, the stock, the process plan with feature information, and the delta volumes or in-process models for each feature cut action are required from FBMach.

4.2.1.1 The part and stock

The part is the final part that users want to produce and can be created from any CAD system. The stock is the initial work piece that users start with and is defined in FBMach. FBMach reads in the part model in SAT file format to recognize features and generate process plans. UG cannot understand solid models in SAT format, so the models need to be converted into STEP format. When exporting the process plan, the models of the part and stock are saved separately in the neutral file format, STEP AP203. The part model and the stock model are two inputs of the integration layer. The models are going to be translated into UG proprietary format and then imported to the system.

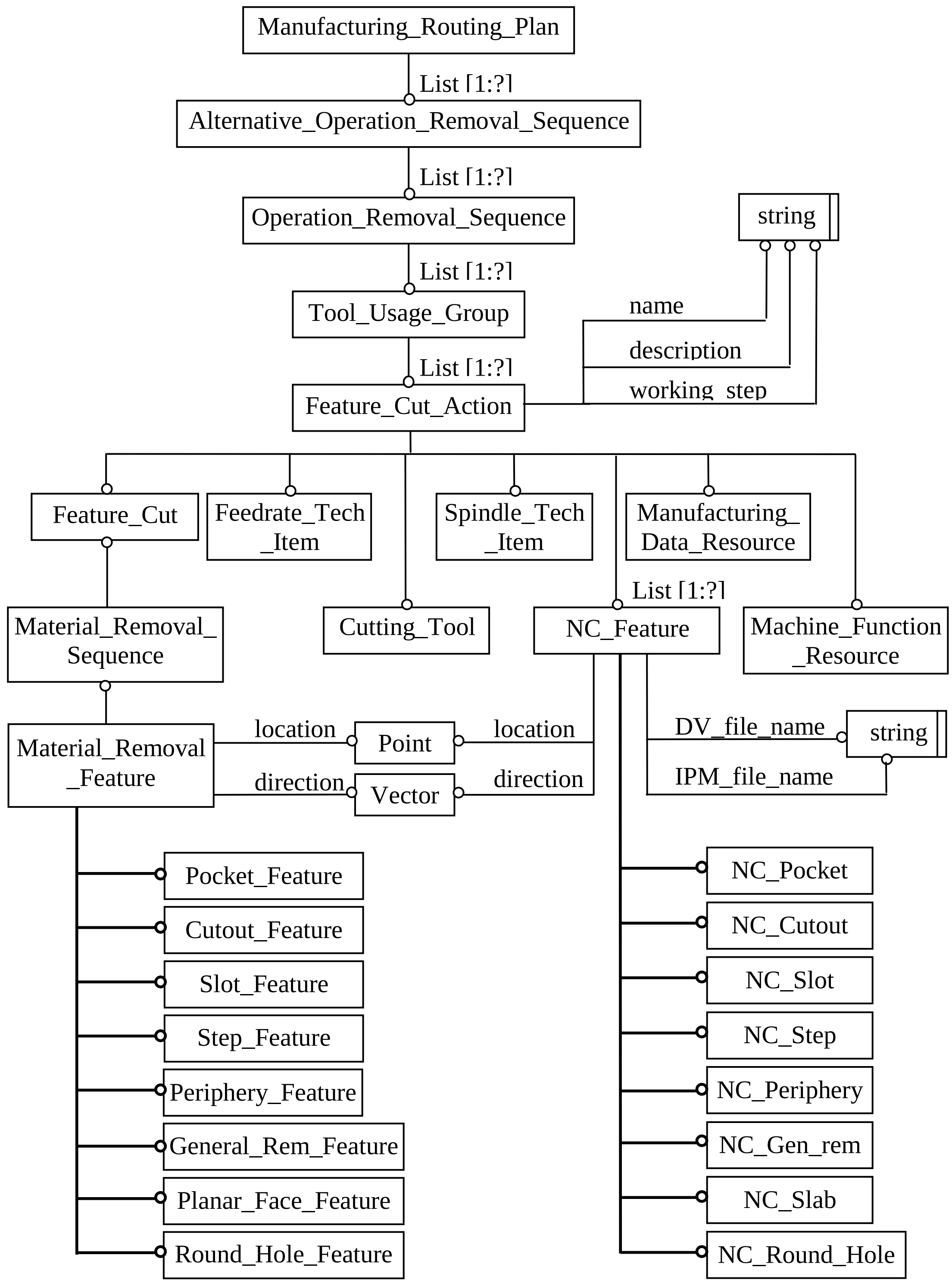


Figure 4.2 High level data model of process plan information

4.2.1.2 Machining features

It has been mentioned that there are two kinds of machining features in FBMach, material removal features and NC features. Material removal features correspond to the geometry on the final part. NC features are in process features that describe the intermediate work piece existing during the machining process. The removal volume for a cut action may not be the same as a feature on the final part. Delta volumes and in-process models are used to determine machining geometries for each operation created by the system. Delta volumes and in-process models are solid models and saved in STEP AP203 format. The references of the STEP files are given in process plan files. The STEP files will be translated and imported into the system when preparing for machining geometries. Different types of features have different attributes besides the geometry of features. The location and orientation of features are common ones available from FBMach. Other key attributes of a feature are also given in process planning files. These attributes are used to determine the kind of feature cut actions and associated parameters.

4.2.1.3 Machining sequence

The process plan contains two parts of information, routing plan and features. The routing plan is a high level process planning node that contains one or many operation alternatives. Only one operation sequence in the operation alternatives is active, meaning it is the sequence to be performed, and is going to be exported to the process plan file. The part will be machined by a sequenced set of feature cut actions where the information of material removal feature and NC feature are accessible. Feature cut actions are grouped into tool usage groups that are contiguously ordered sets of feature cut actions to be performed with a specific cutting tool. Cutting tool information and basic machining parameters including cut feed rate, spindle speed, coolant status, cut depth, are specified in the process plan as well.

4.2.2 Data model of UG CAM objects

In UG to interactively generate tool paths for a part, the user has to first create an operation and then specify a cutting tool, machining geometries and appropriate cutting parameters for the operation. When these steps are finished, the operation is ready to generate tool paths. In the current UG system all these steps have to be done by the user manually, even the macro-level plan has been generated in a CAPP system.

A class library of UG CAM objects is built to encapsulate basic CAM objects for easy access and manipulation of those objects in the program. The basic objects for UG manufacturing application are operations, machining geometries, cutting tools and machining parameters. A high level data model of UG CAM objects is illustrated in Express-G language in figure 4.3. The EXPRESS-G diagram for the complete data model of UG CAM objects is shown in Appendix B as an entity-level model. The object model can be expanded to include more types of operations and related objects, such as turning operations and turning tools. When later we make the system

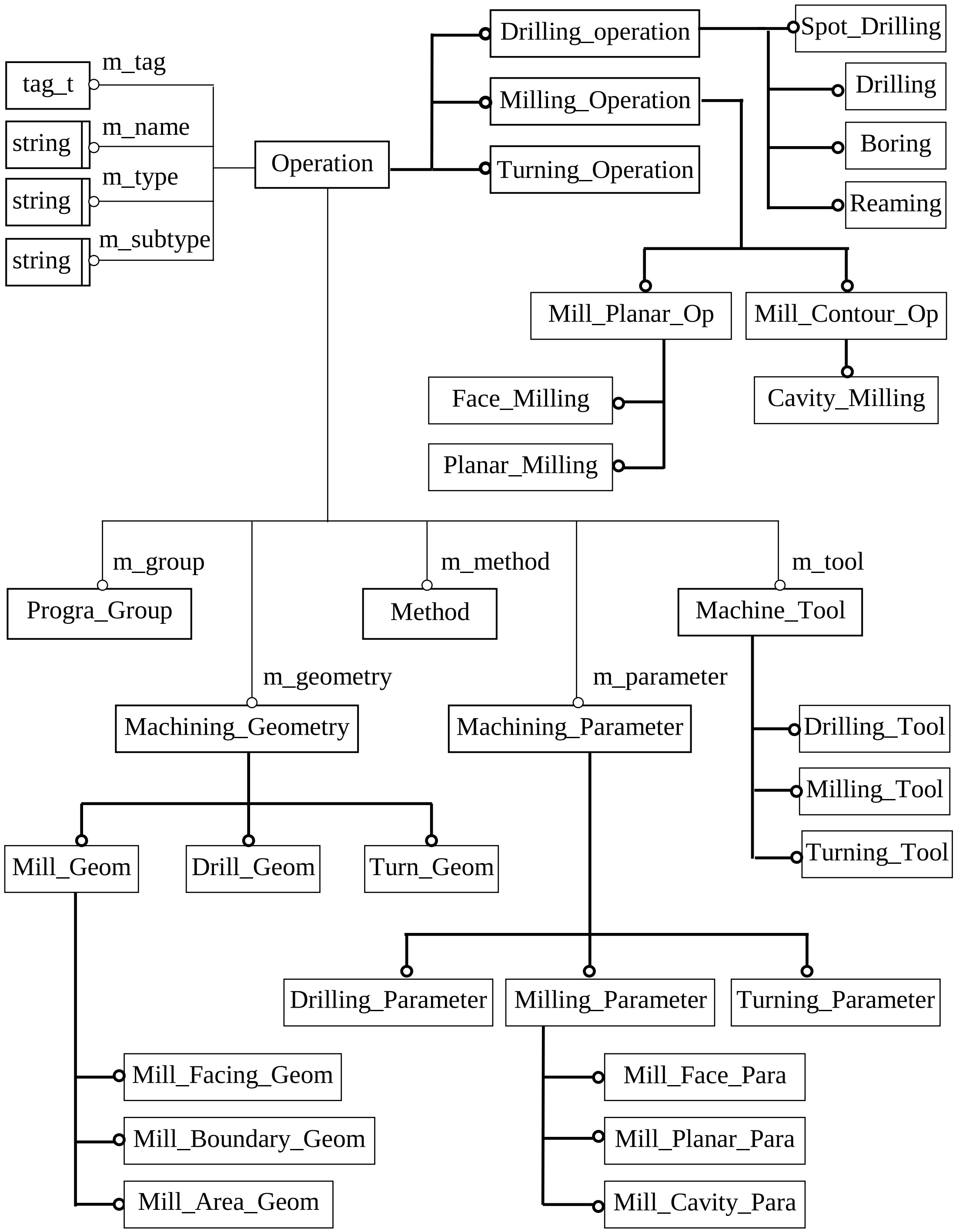


Figure 4.3 High level data model of the UG CAM object library

compatible with process plans in other formats, i.e. STEP-NC files, the object library is reusable.

4.2.2.1 Operation

An operation defines a material removal action, containing all the information needed to generate tool paths for the operation. An operation consists of the name and type of the operation, coordinate system information, a reference to the model geometry, cutting tool information, and manufacturing parameters, etc. The hierarchy relationship of the machining operations can be found in figure 4.3. Drilling, milling and turning operations are subtypes of operations. In this thesis, only prismatic features and associated 2.5-axis operations are explored. Turning operations and related objects shown in figure 4.3 are for expanding the CAM object library later. Face milling, planar milling, and cavity milling operations are implemented for milling features. Standard drilling, spot drilling, boring and reaming operations are implemented for round hole features.

4.2.2.2 Machining Geometry

Machining geometry describes the faces or areas to be machined for a specific operation. The hierarchy relationship of the machining geometry can be found in figure 4.3. Drill geometry is used for drilling operations. The holes of drill geometry are mandatory while the part surface or bottom surface is optional depending on how the cycle parameter is specified for the drilling operation. Mill facing geometry is used for face milling operations. The faces of mill facing geometry must be specified to define the area to be faced off. Mill area geometry is used for cavity milling operations. The part and blank geometry of mill area geometry can be solid models. Cavity milling operation calculates the differentiation between the stock and desired final shape, and generates the tool paths for the differentiations.

The mill boundary geometry is used for planar milling, which may include the floor, part geometry, blank geometry, and check geometry. Part geometry specifies geometry that represents the finished part. Blank geometry specifies geometry that represents the raw material to be removed. Check boundaries are used to in addition to the specified part geometry to define areas that the cutting tool should be kept away from to avoid tool interference with unexpected cut areas of the part or clamps. The floor defining the lowest cut level is mandatory. The combination of part, blank and check geometry defines the area of the part to be machined.

4.2.2.3 Cutting tool

A cutting tool is defined for the operation before tool paths can be generated. Cutting tools are categorized into drilling, milling and turning tools as shown in figure 4.3. Spot drill, twist drill, reamer, and boring tools are implemented for machining hole features. Face mill and end mill tools are implemented for machining milling features.

4.2.2.4 Machining Parameters

Machining parameters are the options that relate to the cutting tool and its interaction with the part material while cutting. Drilling parameter, milling parameter and turning parameter are subtypes of machining parameters as shown in figure 4.3. Cycle parameters of drilling are unique parameters for hole features. Milling parameters are classified into face mill parameter, planar mill parameter and cavity mill parameter, used for face mill, planar mill, and cavity mill operations respectively. Besides the basic parameters corresponding to those provided by FBMach, some other parameters are considered, such as cut method and avoidance geometry. All the necessary parameters are either provided from FBMach, or determined in the system.

4.3 Map the process plan information to CAM objects

The integration layer between FBMach and Unigraphics is developed to connect FBMach and Unigraphics and pass the information between the two systems. FBMach utilizes automatic feature recognition technology to create process plans automatically from CAD solid models. The inputs of the integration layer are the part, the stock, feature information, in process status of the part, and the machining sequence. After the interface reads in the process plan it maps the plan information into CAM objects, so it can create machining operations with associated parameters and generates tool paths for the operations automatically. All the information coming from FBMach is understandable to UG through the interface, which greatly reduces the user interaction required to generate tool paths.

4.3.1 Feature Cut Actions to Operations

A feature cut action in FBMach is mapped into an appropriate UG operation according to the type of working step and the geometry of the material removal feature. An operation is created from a feature cut action, and requires geometry, a tool, and parameters to generate a tool path. Table 4.1 reveals the mapping relationship between feature cut actions and UG operations.

Table 4.1 Mapping of FBMach Feature Cut Actions to UG operations

FBMach			**Unigraphics**
Material Removal Feature	**NC Feature**	**working_step of Feature Cut Action**	**CAM Operation**
Planar Face	NC Slab	planar_face_workingstep	Face Milling
General Material Removal	NC General Removal	pocket_workingstep	Planar Milling
Step	NC Step	step_workingstep	Planar Milling
Slot	NC Slot	slot_workingstep	Planar Milling
Periphery	NC Profile	profile_workingstep	Planar Milling
Cutout	NC Cutout	general_cutout_workingstep	Cavity Milling
Pocket	NC Pocket	pocket_workingstep	Cavity Milling
Round Hole	NC Round Hole	drilling_workingstep	Spot Drilling Drilling Boring Reaming

The feature cut actions for round hole features are mapped into drilling operations, i.e. spot drilling, standard drilling, boring or reaming operations. For hole features, usually feature cut actions with the same attributes are organized into one tool usage group. If the MCS, the cutting tool and the machining parameters are same for all feature cut actions it is possible to create only one operation with geometry properly specified. To improve machining efficiency, each tool usage group for hole features with the same attributes is correspondingly mapped into one UG drilling operation.

The feature cut actions for planar face features are mapped into face milling operations that are designed specifically for the planar faces of a part. For pocket or cutout features, the areas to be machined are always closed. The feature cut actions for these features are mapped into cavity milling operations. The feature cut actions for other milling features whose geometry has arbitrary shape and various boundary conditions are mapped into planar milling operations.

4.3.2 Machining Feature to Machining Geometry

The relationship between material removal features and solid model of the part is lost because of the data translation. When a solid model is saved as a STEP file the identifiers used to relate a material removal feature to entities of the model are not preserved in the solid model, so the association between features and the geometry of the final part is lost when solid models are exported from FBMach. The associations can be reconstructed with the geometry model and the available feature information. Even though the associations can be reconstructed, very often a machining operation needs the geometry from intermediate processes that is not available on the final part. The machining geometries for the operations are obtained from the delta volumes and in-process models. Due to different characteristics of machining features, searching for machining geometry is not the same for different types of features. Table 4.2 shows the required machining geometry combination that must be specified for different features.

Table 4.2 The machining geometry specification

	Machining Geometry					
	Floor	Part	Blank	Check	Faces	Holes
Planar Face		X			X	
General Removal	X	X	X			
Step	X	X	X			
Periphery	X	X	X			
Slot	X	X				
Cutout		X	X			
Pocket		X	X			
Round Hole						X

4.3.2.1 Drilling operations

The top circular edge of each hole feature is searched from the current in-process model and is specified as hole geometry. When a drilling operation is created from one tool usage group, the hole from each feature cut action is appended to the hole geometry. There is one blind hole and one through hole on the part shown in figure 4.4. The diameters of the two holes are the same so they can use the same drilling tool for machining. They are to be machined in one drilling operation, and both holes are specified as hole geometry for the operation.

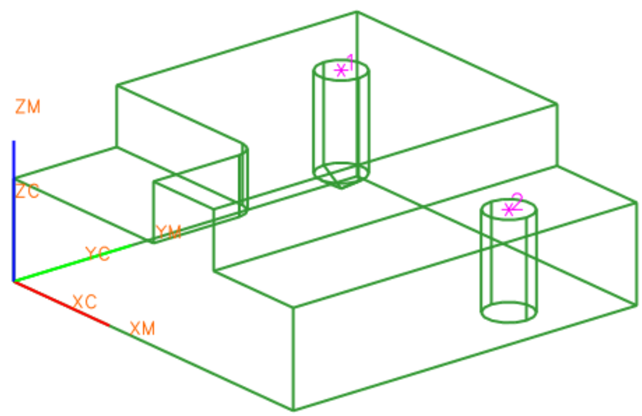


Figure 4.4 Machining geometry specification for hole features

4.3.2.2 Face Milling

The planar faces to be machined are specified as the face geometry for a face milling operation that machines a large open area. All the faces of the in-process model are traversed and the origin and direction of each face are compared with the location and direction of the NC feature to find the planar faces to be machined on the in-process model. Figure 4.5 shows the face geometry for the face milling operation of a planar face. The initial stock is a simple block with the height slightly larger than the height of the boundary box of the part as shown in figure 4.5. The top face of the in-process model is set as the face geometry for the operation and the in-process model that represents the actual shape after the face milling operation is set as the part geometry.

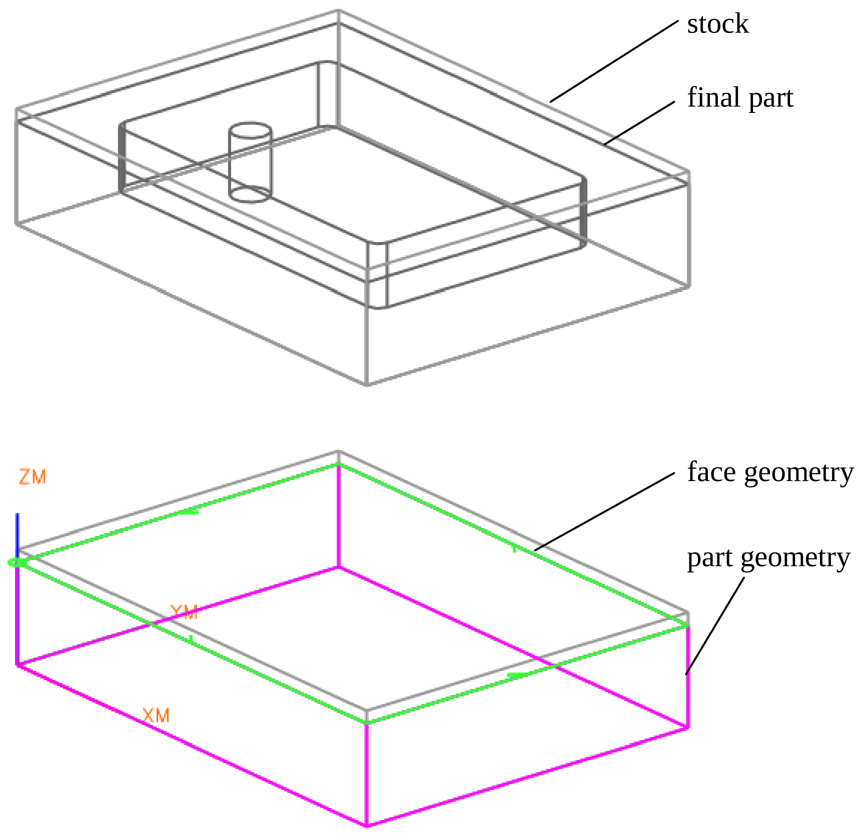


Figure 4.5 Machining geometry specification for a face feature

4.3.2.3 Planar Milling

Delta volumes representing NC features and in-process models representing in process status of the part are solid models. For planar milling operations machining geometry cannot be satisfactorily defined with only solid bodies. Faces or a set of curves or edges are required as machining geometry for planar milling in UG. The location and orientation of the NC feature are used to find the bottom face of the delta volume by the ray tracing method. The bottom face will be the floor geometry to define the lowest cut level for the operation. For slot features the bottom face of the delta volume is specified as part geometry and its material side is set to outside so the area enclosed by the boundary is to be machined. The tool position for slot ends is set to tangent to (Tanto) or On depending on the slot end condition. There is a slot with one open end and one closed end as shown in figure 4.6. The tool position for the open end edges is set to On and the tool position for the closed end edges is set to Tanto.

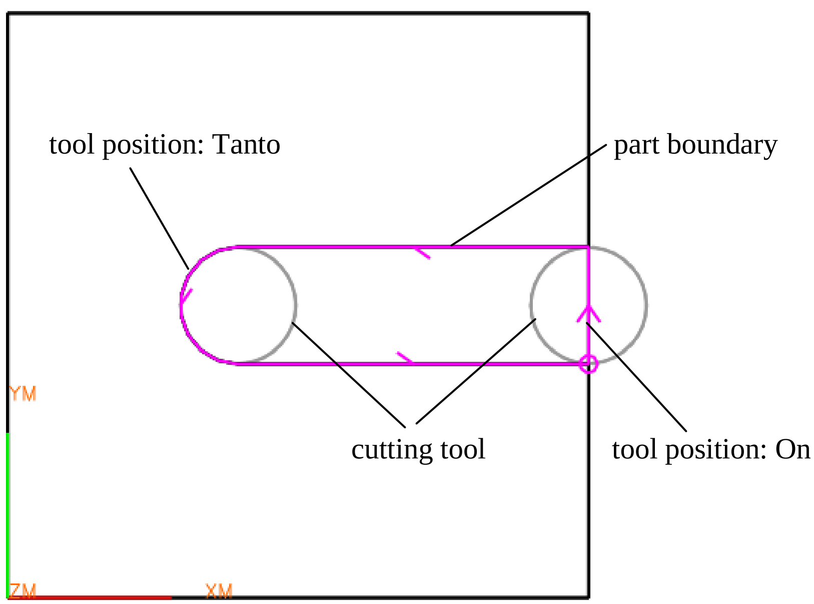


Figure 4.6 Machining geometry specification for a slot feature

For general removal, step, and periphery features, blank and part geometry have to be specified for the operation besides the floor. The blank geometry is searched from the delta volume that represents the material to be machined and the part geometry is searched from the in-process model that represents the to-be shape for a specified operation. The faces on the delta volume that are parallel to and above the floor are set as the blank geometry. The faces on the in-process model that are parallel to and above the floor are set as the part geometry. Figure 4.7 shows the blank and part geometry specified for a general removal feature.

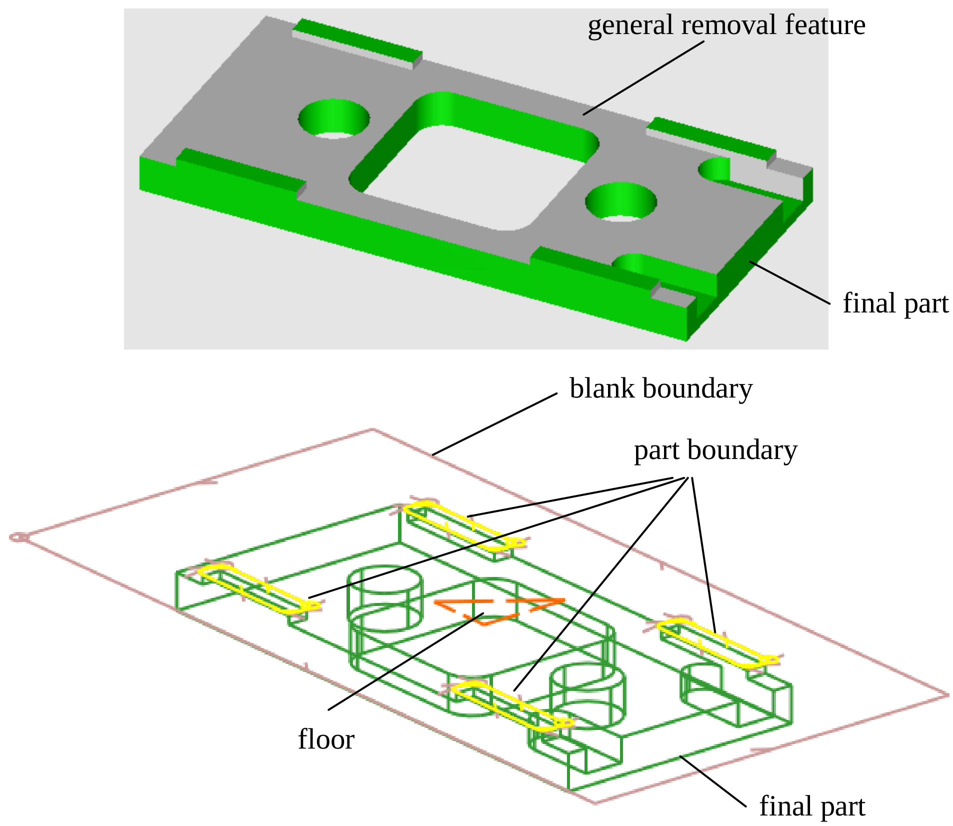


Figure 4.7 Machining geometry specification for a general removal feature

4.3.2.4 Cavity Milling

For a cutout or pocket feature, the cutting tool is restricted inside the cutout or pocket area, and it is inappropriate to travel beyond the outer profile of the cutout or pocket. Pockets can have islands, which is impossible for cutouts. The other difference is that pockets have a bottom face but cutouts do not have one. In some systems a cutout is regarded as a bottomless pocket. [58] The cut area for the operations, however, is always a closed area. The blank and part geometry need to be specified for cavity milling operation. For a specified operation, the in-process model of the corresponding feature cut action is specified as the part geometry, and the in-process model of the previous feature cut action, which is the start shape of this operation, is specified as the blank geometry. Figure 4.8 shows the blank and part geometry of the rough milling operation for a pocket feature.

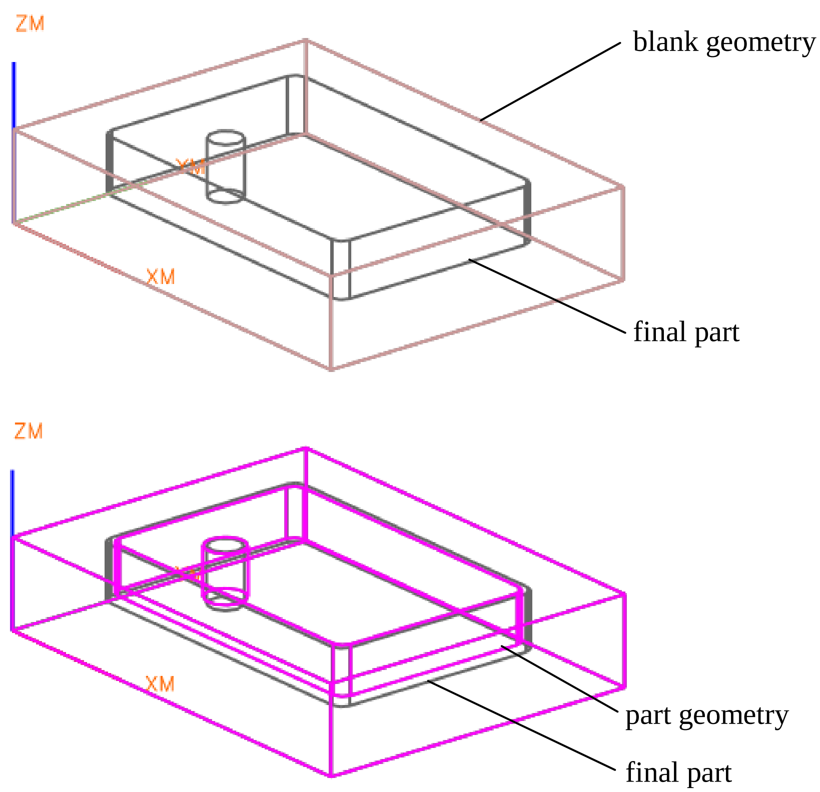


Figure 4.8 Machining geometry specification for a pocket feature

4.3.3 Cutting Tools

Cutting tool mapping is pretty straightforward. A cutting tool from FBMach is to be mapped to an appropriate tool in UG and specified to the related operation. The type of a cutting tool should match the type of the related operation. The cutting tool needs to be able to machine the part and avoid collision with the part. For example, the diameter of the cutting tool for a rough drilling should be slightly less than the hole diameter, the flute length should be sufficient to machine the hole, and the total tool length should be long enough to avoid the tool holder colliding with the part.

For an operation, it is not necessary to define all the parameters of a cutting tool. The important parameters, such as the maximal tool diameter and the flute length, could be given instead of a fully defined tool. It is more flexible not to define a specific cutting tool. In this case, the tool of a previous cut action is used whenever possible to reduce tool changes. Otherwise select a tool from the library or create a new one that satisfies the requirement of the cutting tool. The criteria to choose a cutting tool for rough operations is looser than the ones for finish operations. For example, to machine a simple pocket, the radius of cutting tool for rough milling operations can be larger than the radius of pocket corners, but for finish milling operations the radius of cutting tool has to be equal or less than the smallest corner radius. If the pocket has islands, to avoid gouges with the part, the maximum distance between side walls and islands must be considered to determine the cutting tool radius.

4.3.4 Machining parameters

Some basic machining parameters are specified in FBMach and those parameters are given explicitly or implicitly in the process plan file. The common parameters considered for milling and drilling operations include cut feed rate, spindle speed, coolant status, cut depth, and clearance plane. Other than that there are some unique parameters for different operations, like stepover for milling operations and cycle parameter for drilling operations. Table 4.3 shows the mapping of machining parameters available from FBMach to UG parameters.

Table 4.3 Machining parameters in FBMach and UG

FBMach	UG
DEFAULT_FEEDRATE_TECH_ITEM_KCP	Cut feed rate
DEFAULT_SPINDLE_TECH_ITEM_KCP	Spindle speed
MACHINE_FUNCTION_RESOURCE_KCP	
coolant	Coolant in Machine Control post command
MANUF_DATA_RESOURCE_KCP	
axial_cut_depth	Cut depth/Cycle depth
radial_cut_depth	Stepover
secure_plane	Clearance distance/plane
retract_plane	Return point

Besides those basic parameters given in FBMach, some important parameters such as MCS specification, cut method and engage/retract method, are specified in the system based on the operation type and attributes of machining geometry. The direction of machining features is used to determine the MCS and tool axis for the operation. A MCS can include multiple operations that have the same set-up. The retract and engage method is set to automatic and UG will determine the method based on machining geometry specified to lead the tool into the work piece from the sides rather than plunging it downwards when possible. There are four cut methods for milling operations as mentioned earlier. In the zig cut method, the cutting tool goes in a constant direction while in the zigzag cut method, the cutting direction alternates between two successive cut passes. The zigzag method results in a shorter tool path and less machining time, but the surface finish is poor on the boundary of the features. The follow part method cuts the work piece following the contour of its boundary and hence results in a smooth boundary. Therefore, the zigzag method is used for face milling operations and the follow part method is used for planar and cavity milling operations.

5 Results and Discussions

5.1 Examples and results

The integrated system has been tested with several parts and demonstrated satisfactory results. Four examples are presented in this section to illustrate the developed system. Process planning for the parts are completed in FBMach, and process plan files have been exported from FBMach and ready to use as inputs to the integration layer. FBMach identifies how each feature should be removed and in what order after all machining features are recognized either automatically or manually. Automatic feature recognition in FBMach still has difficulties in recognizing all the features, especially interacting features, so it also supports interactive recognition and manual identification in the cases of interacting features or when users want to change the automatically recognized features. After process planning, users only need to specify three files: the part model, the stock model and the process plan file as the inputs to the prototype. Given these three files, the integration layer retrieves and processes the information from FBMach and automatically generates and displays the tool paths for the specific process plan. Users then can simulate or verify the tool paths in UG before post processing to get NC programs.

5.1.1 Example 1

The first example uses the test part shown in figure 5.1. The dimensions of the test part are 1.36 inches by 0.63 inch by 0.11 inch. The stock defined in the example is a simple block of 2 inches by 1 inch by 0.25 inch, slightly bigger than the bounding box of the test part, and enclosing the test part in the center of the stock. This example demonstrates six types of features, including cutout, general removal feature, slot, periphery, round hole and planar face as shown in figure 5.2. In the process plan there are a total of 9 material removal features and 22 feature cut actions for this part that are shown in figure 5.3. It takes about five minutes for the system to generate and display tool paths for the process plan within UG. The operations and tool paths generated in UG are shown in figure 5.4.

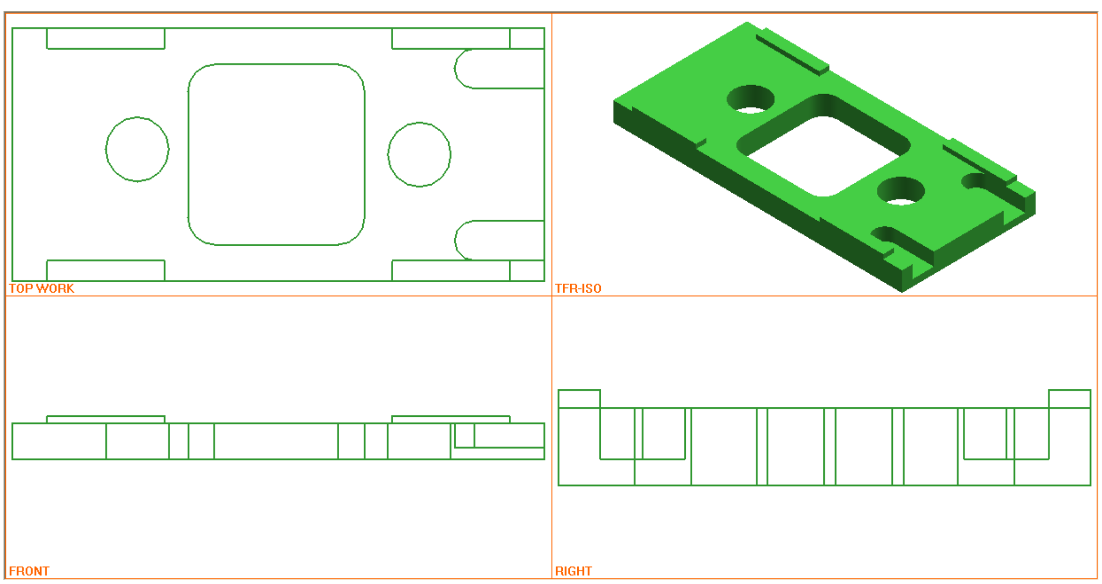


Figure 5.1 The test part of example 1

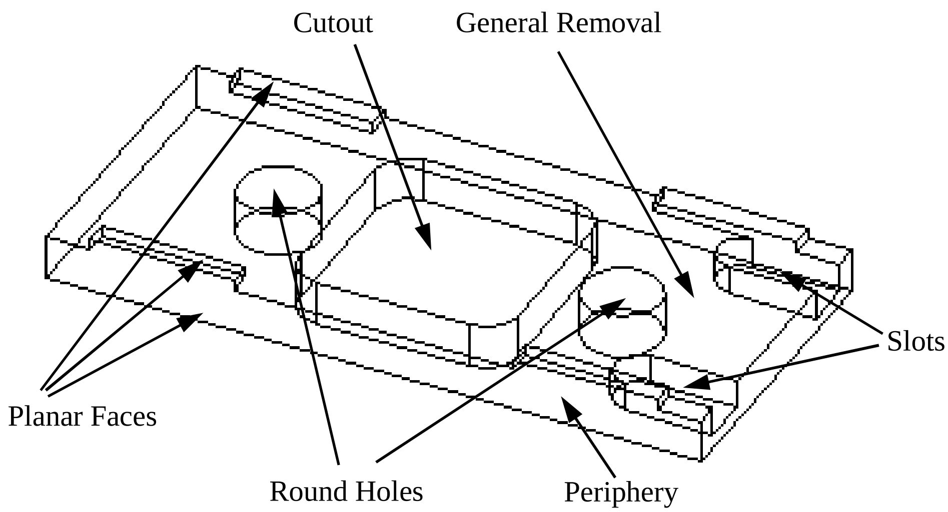


Figure 5.2 The machining features on the test part

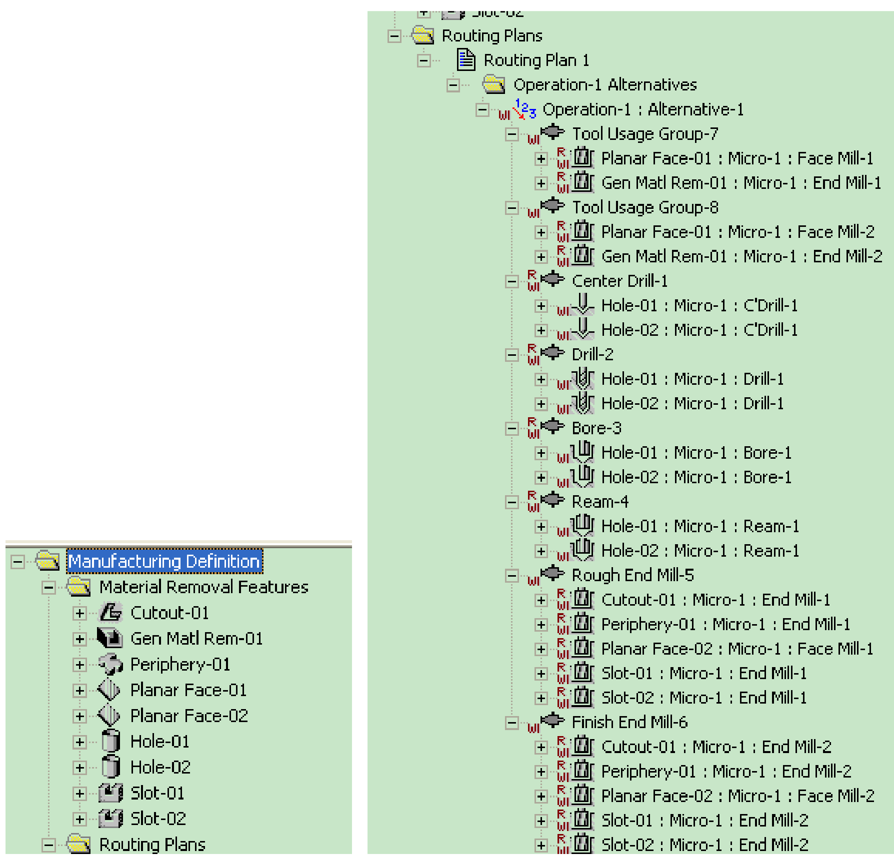


Figure 5.3 Material removal features and operation sequence in FBMach

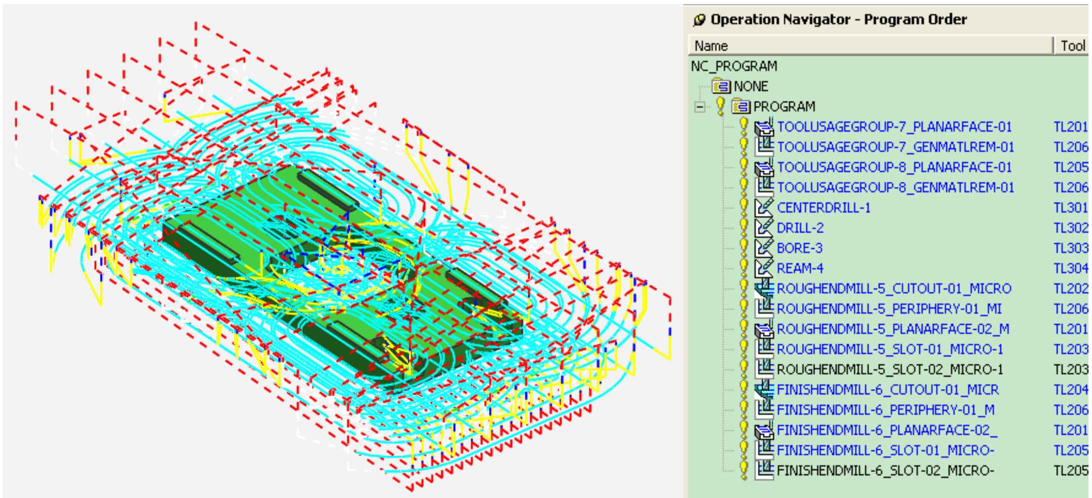


Figure 5.4 The operations and tool paths generated in UG

5.1.2 Example 2

The recognized machining features also depend on the size and shape of the stock. If a different stock is defined, the number and types of material removal features and feature cut actions may be different. The process plan in this example is for the same test part as in the first example, but the stock definition is different. The stock in this example is a simple block that just encloses the part and its dimensions are the same as the boundary box of the part. There are no planar face and periphery features in this example because of the different stock definition and hence no feature cut actions for these features. The material removal features and operation sequence are shown in figure 5.5 and the operations and the tool paths in UG generated from the process plan are shown in figure 5.6.

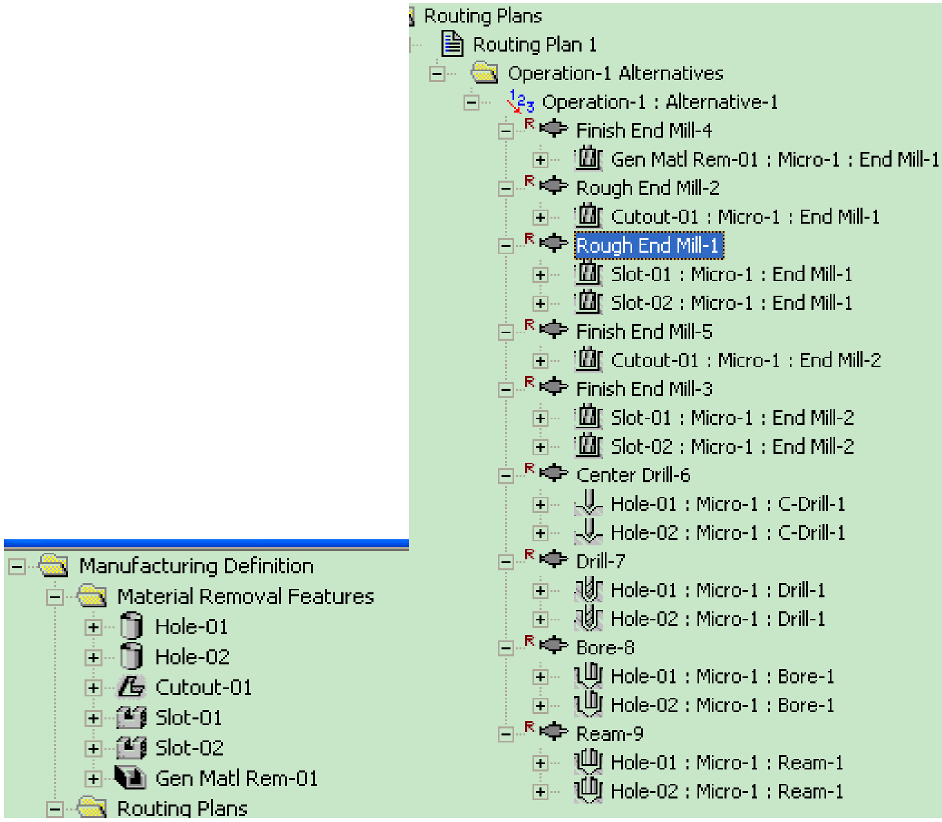


Figure 5.5 Material removal features and operation sequence in FBMach

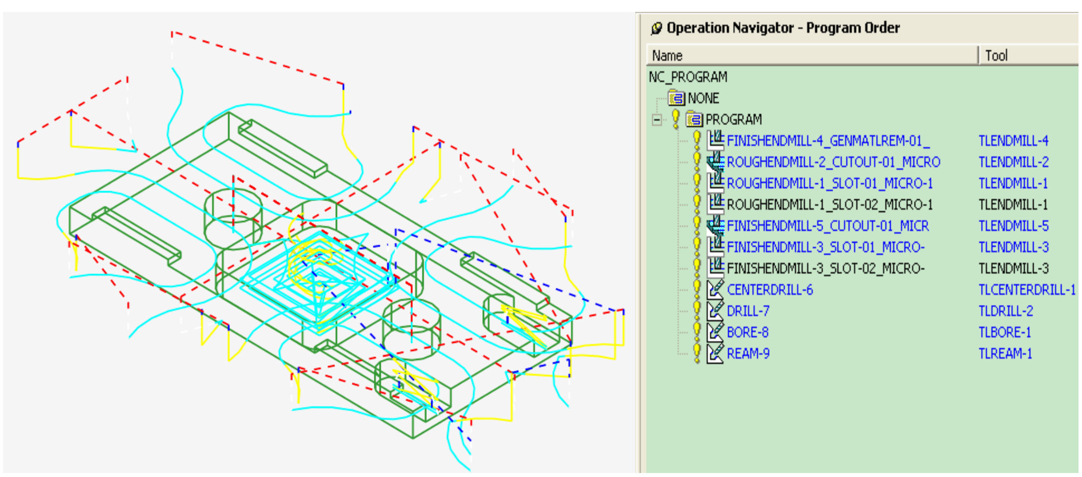
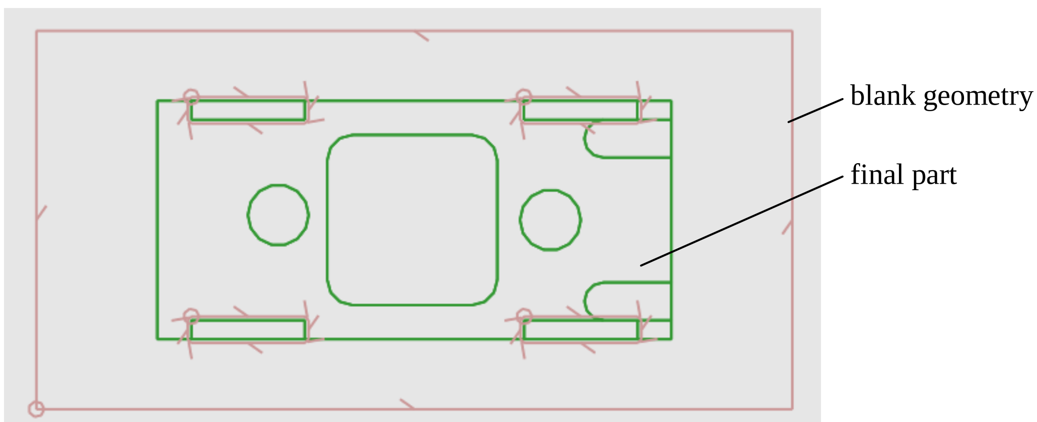


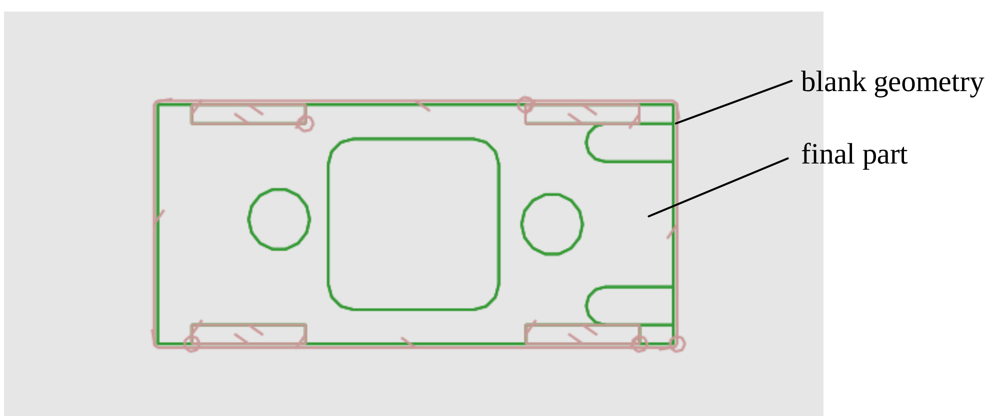
Figure 5.6 Operations and tool paths generated for example 2

5.1.3 Example 3

The feature cut actions in a process plan may be sequenced in different ways to produce the same part. The process plan of third example is for the same test part with the same stock as the first example, but with a different sequence of feature cut actions. In the first example the general removal feature is machined before the periphery feature, and in this example the general removal feature is machined after



a. blank geometry for the general removal feature in example 1



b. blank geometry for the general removal feature in example 3

Figure 5.7 Blank geometry for the general removal feature in different plans the periphery feature. The machining geometry for the same features in the two plans is different due to the different operation sequence. The blank geometry for the general removal feature in the two examples is shown in figure 5.7.

5.1.4 Example 4

This example uses a different test part that is shown in Figure 5.8. This test part has a more complex shape and needs to be machined from different directions. The stock defined in the example is slightly larger than the bounding box of the final part, and encloses the part totally as shown in figure 5.8. If defining a stock of different size or shape, the feature set and cut actions may be different. This example demonstrates planar faces, periphery, slot, step, round hole, and general removal features. Those features are in different directions and need different MCS to machine the part. The operations and tool paths generated according to the process plan are shown in figure 5.9. The step feature shown in figure 5.10 can only be machined from the side. The MCS shown in figure 5.10 is created based on the direction of the step feature and specified for the operation.

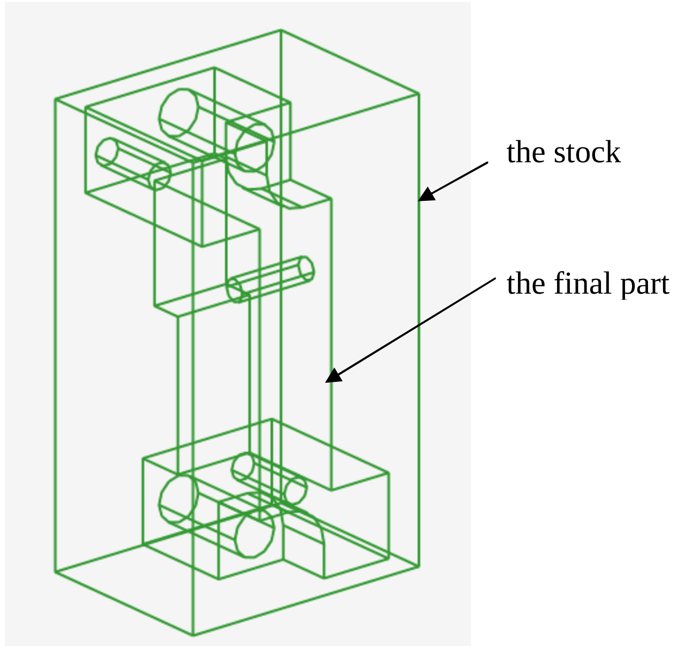


Figure 5.8 The final part and stock for the fourth example

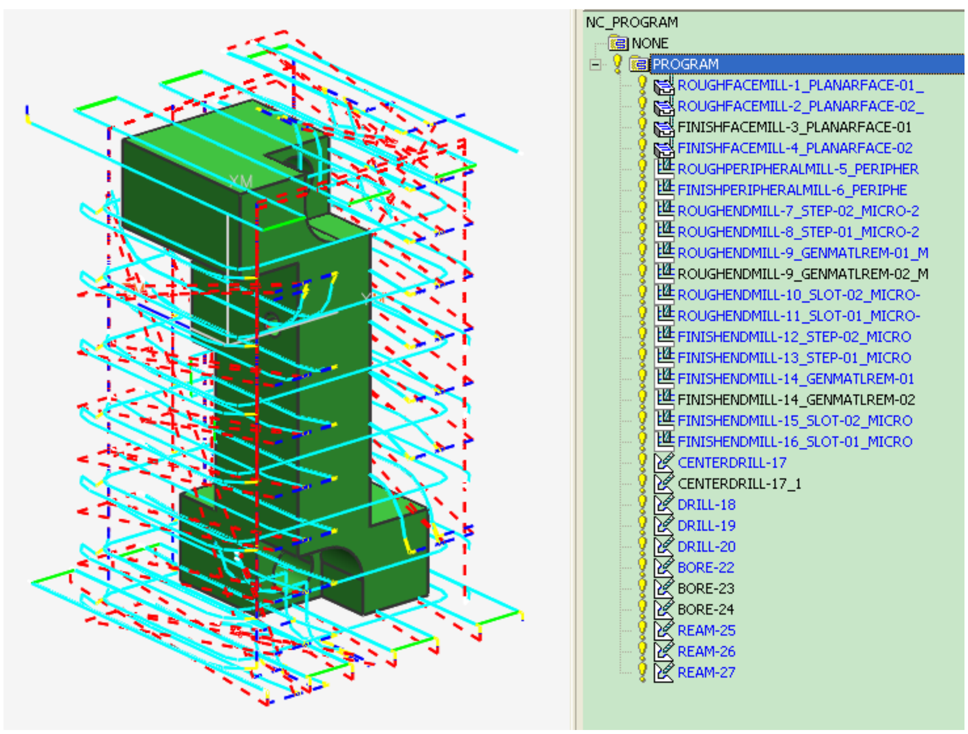


Figure 5.9 The operations and tool paths generated in UG

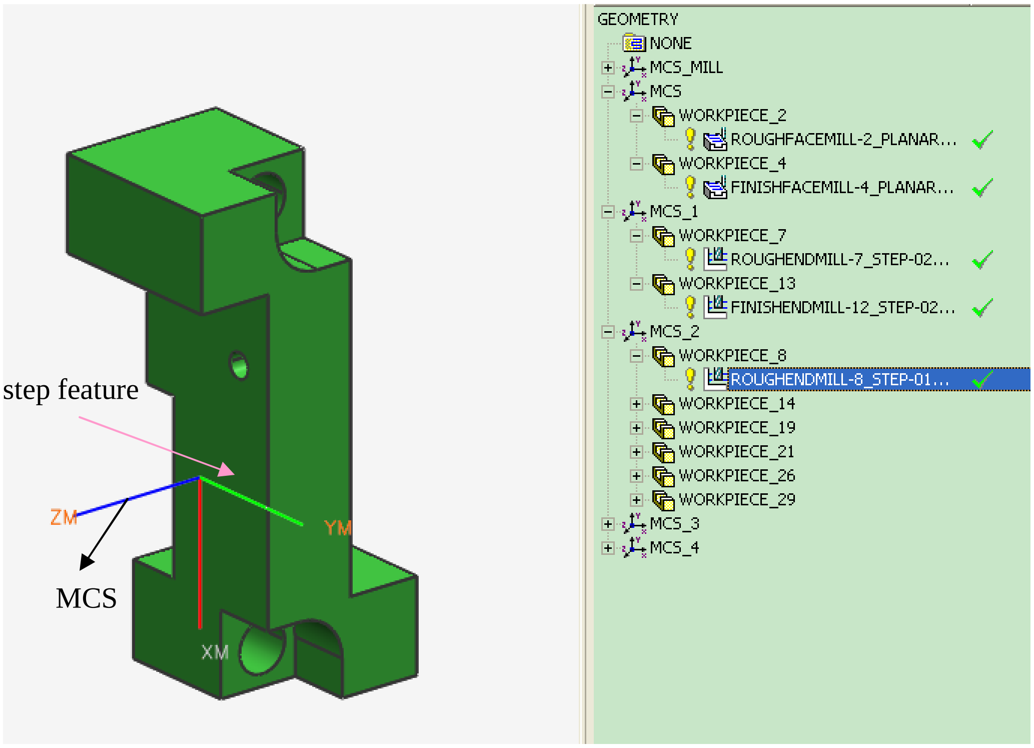


Figure 5.10 The machine coordinate system for the step feature

5.2 Discussions

Current available commercial CAD/CAM packages, including Unigraphics, are limited in terms of integration. Although CAD and CAM systems in the same family share the geometry data of the part from design, the users have to make decisions on what and how to cut the part based upon their expertise, and have to reconstruct all the in-process machining geometries for the operations. Due to the extensive manual interactions the process is time-consuming and error-prone. In the integration layer users only need to specify three files, the part file, the stock file and the process plan file. Given the three files the integration layer can retrieve and process the process plan and feature information from FBMach and automatically generate and display the tool paths for the specific process plan.

The machining geometry for each cut very likely changes when the process plan changes. The geometry and sometimes the machining parameters have to be specified again by the user. With this integrated CAD/CAM system the in-process models and the machining geometries are recalculated by the system when the plan changes. It would normally take hours to redo the work and generate tool paths for a new process plan, but with the integrated system that is reduced to minutes to do it. The system is helpful to evaluate different process plans for a part with different cutting sequences, different cut actions, or stock changes.

The integration layer creates machining operations in UG corresponding to cut actions defined in FBMach. Some operations can be combined to optimize the process plan, providing that the cutting tool and all other cutting conditions are the same. The operations for holes are analyzed and optimized in the current implemented system. Since the two through holes in the test part have the same directions, diameter, depth and cutting parameters, the operations for the two holes are combined into one. As shown in figure 5.4 the DRILL-2 operation is for both through holes. Planar milling operations may have similar situations, like the two slots, but for now the system has not analyzed planar milling operations.

The speed of the process of tool path generation would be linear to the number of machining features and feature cut actions in a process plan. Machining parameters, such as cut depth or stepover, would also influence the speed of tool path generation. The smaller the cut depth or stepover means more tool paths that need to be generated and therefore more time that needs to be spent. Before the operation is ready to generate tool paths, the search and specification of machining geometry takes most of the time.

6 Conclusions

6.1 Conclusions

Machining feature is a key concept to seamlessly integrate CAD, CAPP and CAM systems. The thesis has discussed the integration of CAD/CAM systems based on machining features. A prototype is developed in conjunction with a CAPP system, FBMach, and a commercial CAD/CAM system, UG to demonstrate the CAD/CAM integration for prismatic parts. In the prototype machining operations are created according to the process plan, associated information is specified for the operations and then tool paths are generated in UG. Machining features are utilized to define machining geometries and eliminate the necessity of user interventions in UG. Once the features are recognized and the process plan is generated from the solid model, the information is directly available to the system and tool paths can be automatically generated with solid models and process plans.

The implementation of the prototype includes three parts, reading in and interpreting process plan information, building a UG CAM object library to store manufacturing information, and converting the process plan information into UG CAM objects to get ready for tool path generation. The process plan information is transferred from FBMach to UG through FBMach exported files. FBMach files are in ASCII text format and include the information of features and process plans. A UG CAM object library is built to define UG CAM objects, which are categorized into four kinds of objects: operations, machining geometries, cutting tools and machining parameters. Currently the class library supports face milling, planar milling, cavity milling and drilling operations. It is easy to expand the library to support additional types of operations and related objects. An integration layer between FBMach and UG is implemented to read in the process plan, map the information into CAM objects and generate tool paths according to the process plan. The explicit inputs of the integration layer are the process plan file from FBMach, the geometry model of the part, and the geometry model of the stock. The implicit inputs include the geometry models of delta volumes and in-process models of machining features. The CAM object library is reusable if we migrate to use STEP-NC files as input.

The research focuses on enhancing the connection between process planning and tool path generation. All the information exported from FBMach is comprehensible to UG through the interface, which greatly reduces the user interaction required to generate tool paths. Both geometry information of the product and process plan and machining feature information are transferred from FBMach to UG. Therefore, the system is able to specify all necessary information to prepare the operations for tool path generation without user interactions. The process of tool path generation is automated in the integrated system.

6.2 Future research

The integrated system is implemented for prismatic features and related operations and does not support turning operations, or multi-axis machining operations. For future research, turning operations as well as complex multi-axis milling operations can be included in the system. Robust automatic feature recognition for turning features and complex milling features is needed for the integration. The data models of process plan and CAM object library will need to be expanded to include different operations and their associated objects.

STEP-NC is promising on CAD/CAM integration although its development is still in the initial stage. Currently user defined format is used in FBMach exported files for transferring the process plan and feature information. The system can migrate to STEP-NC format for transferring information between CAPP and CAM systems. STEP-NC allows two-way communication between CAD, CAPP and CAM systems. The process changes made in the CAM system can be sent back to the CAPP system through STEP-NC files.

In a process plan if the cutting tool and all other cutting conditions are the same, the automatically created operations may be combined to optimize the process plan. The analysis and combination of drilling operations is implemented in the system. Milling operations also need to be analyzed and combined for optimization. A product may be machined by different operation sequences. To evaluate different process plans of a part, the functions to calculate the cost and time for different operation sequences can be developed.

References

[1] Colin B. Besant. *Computer-Aided Design and Manufacture*. 3rd ed. Ellis Horwood Ltd. 1986

[2] Willian F. Gibbs. CAD/CAM integration – what's the right choice for you? *Modern Machine Shop*, July 1998. http://www.mmsonline.com/articles/mtg9807.html

[3] Steven L. Brooks. *Representing Manufacturing Features to Support Design and Process Changes*. Ph.D. thesis, The University of Kansas, Lawrence, KS. 1997

[4] The official STEP-NC page, http://www.step-nc.org/

[5] FB-Machining Advisor Tutorial. Honeywell International, FM&T. 2002

[6] Unigraphics NX documentation. Unigraphics Solutions Inc. 2002

[7] Christoph M. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann Publishers, San Francisco, CA, 1989

[8] Michael E. Mortenson. *Geometric Modeling*. 2nd. John Wiley & Sons, Inc. 1997

[9] K. Lee. *Principles of CAD/CAM/CAE systems*. Addison-Wesley. 1999

[10] S. A. Brown, C. E. Drayton, and B. Mittman. A description of the APT language. *Communications of the ACM*, Nov. 1963, v6, n11, 649-658

[11] Tien-Chien Chang, Richard A. Wysk and Hus-Pin Wang. *Computer-Aided Manufacturing*. 2nd ed. Prentice Hall. 1998

[12] Rober, Stephen J., and Shin, Yung C. Modeling and control of CNC machines using a PC-based open architecture controller. *Mechatronics*, June 1995, v5, n4, 401-420.

[13] M. Liang, S. Ahamed and B. van den Berg. A STEP based tool path generation system for rough machining of planar surfaces. *Computers in Industry*, December 1996, v 32, n 2, 219-231

[14] Jung Hyun Han, Inho Han and Juneho Yi. Manufacturing feature recognition toward integration with process planning. *IEEE Transactions on Systems, Man and Cybernetics – part B: Cybernetics*, June 2001, v 31, n 3, 373-380

[15] H. B. Marri, A. Gunasekaran and R. J. Grieve. Computer-aided process

planning: a state of art. *The International Journal of Advanced Manufacturing Technology*, 1998, vol. 14, 261-268

[16] S. L. Brooks, K. E. Hummel, and M. L. Wolf. XCUT: a rule-based expert system for the automated process planning of machined parts. *Proceedings of ASME Winter Annual Meeting*, 1997, 181-194

[17] Khoshnevis B., Sormaz D. N., and Park J. Y. An integrated process planning system using feature reasoning and space search-based optimization. *IIE Transactions*, 1999, vol. 31, 597-616

[18] Huikang K. Miao, Nandakumar Sridharan and Jimi J. Shah. CAD-CAM integration using machining features. *Int. J. Computer Integrated Manufacturing*, 2002, v 15, n 4, 296-318

[19] Shah, J. J. and Mantyla, M. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. John Wiley and Sons, New York, 1995

[20] Otto. W. Salomons. *Computer Support in the Design of Mechanical Products: Constraint specification and satisfaction in feature based design for manufacturing*. Ph.D. thesis, University of Twente, Netherlands, 1995

[21] Chen, X. and Hoffmann, C. M. On editability of feature-based design. *Computer-Aided Design*, Dec. 1995, v. 27, n. 12, 905-914

[22] Han, J. H., Pratt, M. and Regli, W. C. Manufacturing feature recognition from solid models: a status report. *IEEE Transactions on Robotics and Automation*, 2000, v. 16, n. 6, 782-796

[23] J. J. Shah. Assessment of features technology. *Computer-Aided Design*, 1991, v. 23, n.5, 331-343

[24] ISO 10303-224:1999. Application Protocol: Mechanical Product Definition for Process Planning Using Machining Features, International Standards Organization, 1999.

[25] Mangesh P. Bhandarkar and Rakesh Nagi, STEP-based feature extraction from STEP geometry for agile manufacturing. *Computers in Industry*, 2000, vol. 41, 3-24

[26] Zhang, S. G., Ajmal, A, Wootton, J, and Chisholm, A. Feature-based inspection process planning system for co-ordinate measuring machine. *Journal of Materials Processing Technology*, Nov. 2000, v 107, n 1-3, 111-118

[27] S. K. Gupta and D. S. Nau. Systematic approach for analysing the

manufacturability of machined parts. *Computer-Aided Design*, 1995, v. 27, n. 5, 323-342

[28] Eric Wang and Yong Se Kim. Feature-based assembly mating relationship. *Journal of Manufacturing systems*, 1999, v. 18, n. 3, 187-202

[29] JungHyun Han, Mujin Kang and Hoogon Choi. STEP-based feature recognition for manufacturing cost optimization. *Computer-Aided Design*, 2001. v. 33, n. 9, 671-686

[30] JungHyun Han and David Rosen. Special panel session for feature recognition at the 1997 ASME Computers in Engineering Conference. *Computer-Aided Design*, 1998, vol. 30, n.13, 979-982

[31] Brooks, S.L. and Greenway Jr., R.B. Using STEP to integrate design features with manufacturing features. *ASME Computers in Engineering Conference*, Boston, MA, September 1995, 579–586

[32] Somashekar Subrahmanyam and Michael Wozny. An overview of automatic feature recognition techniques for computer-aided process planning. *Computers in Industry*, 1995, v. 26, n.1, 1-21

[33] S. Joshi and T. C. Chang. Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-Aided Design*, March 1988, v20, n2, 58–66

[34] S. H. F. Chuang and M. R. Henderson. Using subgraph isomorphisms to recognize and decompose boundary representation features. *Journal of Mechanical Design, Transactions of the ASME*, Sept. 1994, v 116, n 3, 793-800

[35] Woo, T., Feature Extraction by Volume Decomposition, Proc. *Conf. CAD/CAM Technology in Mechanical Engineering, Cambridge*, MA, USA, March 1982, 76-94

[36] D. Waco and Y. S. Kim. Geometric Reasoning for Machining Features using Convex Decomposition. 2nd ACM Solid Modeling Symposium, 1993, 323-331

[37] V. Sundararajan and P. K. Wright. Volumetric feature recognition for machining components with freeform surfaces. *Computer-Aided Design*, 2004, v. 36, n. 1, 11-25

[38] J. H. Vandenbrande and A. A. G Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec 1993, vol. 15, n. 12, 1269-1285

[39] J. Han and A. Requicha. Integration of feature based design and feature recognition. *Computer-Aided Design,* May 1997, vol. 29, no. 5, 393–403

[40] X.G. Ye, J.Y.H. Fuh, K.S. Lee. A hybrid method for recognition of undercut features from moulded parts. *Computer-Aided Design,* 2001, vol. 33, no. 14, 1023-1034

[41] Wang, Eric. Feature recognition using combined convex and maximal volume decompositions. *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005*, Sep 2005, vol. 3A, 469-475

[42] Gupta, Satyandra K., Regli, William C. and Nau, Dana S. Manufacturing feature instances: Which ones to recognize? *Proceedings of the Symposium on Solid Modeling and Applications*, 1995, 141-152

[43] Douglas A. Schenck and Peter Wilson. *Information Modeling: The EXPRESS Way*. Oxford University Press, USA, 1994.

[44] ISO 10303-42. Industrial automation systems and integration -- Product data representation and exchange -- Part 42: Integrated generic resource: Geometric and topological representation. 1994

[45] Denise Janowski. Transferring CAD/CAM files. *Modern Machine Shop*, July 1999, v. 72, n. 2, 70-

[46] Brian C. Kuttner. Why IGES doesn't always work. *Modern Machine Shop*, 1995. http://www.mmsonline.com/articles/cnc9501.html

[47] Martin Hardwick. *STEP into Automatic Machining*. STEP Tools, Inc. 2001

[48] Brunnermeier, S. and Martin, S. Interoperability Cost Analysis of the U.S. Automotive Supply Chain. Research Triangle Institute. March 1999. http://www.nist.gov/director/prog-ofc/report99-1.pdf

[49] Xu, X. W. and Q. He. Striving for a total integration of CAD, CAPP, CAM and CNC. *Robotics and Computer-Integrated Manufacturing*, 2004, v.20, n.2, 101-109

[50] Weck, M., Wolf, J. and Kiritsis, D. STEP-NC—The STEP compliant NC programming interface evaluation and improvement of the modern interface. *Proceedings of the ISM Project Forum*, Genf, Switzerland, October 2001

[51] Suk-hwan Suh, Jung-Hoon Cho and Hee-Dong Hong. On the architecture of intelligent STEP-complaint CNC. *International Journal of Computer Integrated*

Manufacturing, 2002, v.15, n.2, 168-177

[52] S. T. Newman, R. D. Allen and R. S. U. Rosso, Jr. CAD/CAM solution for STEP-compliant CNC manufacturing. *International Journal of Computer Integrated Manufacturing*, 2003, v.16, n.7-8, 590-597

[53] Hardwick, M. On STEP-NC and the complexities of product data integration. *Journal of Computing and Information Science in Engineering*, Mar 2004, v.4, n.1, 60-67

[54] X. W. Xun, H. Wang, J. Mao, S. T. Newman, T. R. Kamer, F. M. Proctor and J. L. Michaloski. STEP-Compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration. *International Journal of Production Research*, Sep 2005, v 43, n 17, 3703-3743

[55] Allsion B Feeny, Tom Kramer, Fred Proctor, Martin Hardwick, and David Loffredo. STEP-NC implementation – ARM or AIM? ISO T24 STEP-Manufacturing Meeting, San Diego, USA, Mar 2003

[56] UG Open API Reference Guide. Unigraphics Solutions Inc. 2002

[57] Jan H. Vandenbrande and Aristides A. G Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Dec. 1993, v. 15, n. 12, 1269-1285

[58] William C. Regli and Dana S. Nau. Building a general approach to feature recognition of material removal shape element volumes (MRSEVs). *Proceedings of the 2nd Symposium on Solid Modeling and Applications,* Montreal, Canada, May 1993, 293-302

Appendix A: A process plan file exported from FBMach

The example file is just to give an idea how the process plan file is organized. Only one hole feature, the cut actions for the hole feature, and other information related to the hole feature are included in the process plan.

```
START_INSTANCE
MATL_REM_FEATURE_KCP "Hole-01"
type = ROUND_HOLE_MATL_REM_KCP
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
bottom_condition = "through"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.09
max_diameter = 0.161
taper_angle = 0
END_INSTANCE

START_INSTANCE
MATL_REM_SEQUENCE_KCP "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
FEATURE_CUT_KCP = "Hole-01:Micro-1:C'Drill-1"
FEATURE_CUT_KCP = "Hole-01:Micro-1:Drill-1"
FEATURE_CUT_KCP = "Hole-01:Micro-1:Bore-1"
FEATURE_CUT_KCP = "Hole-01:Micro-1:Ream-1"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_KCP "Hole-01:Micro-1:C'Drill-1"
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_KCP "Hole-01:Micro-1:Drill-1"
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
END_INSTANCE
```

START_INSTANCE
FEATURE_CUT_KCP "Hole-01:Micro-1:Bore-1"
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_KCP "Hole-01:Micro-1:Ream-1"
MATL_REM_SEQUENCE_KCP = "Hole-01:Micro-1"
MATL_REM_FEATURE_KCP = "Hole-01"
END_INSTANCE

START_INSTANCE
MANF_ROUTING_PLAN_KCP "Routing Plan 1"
description = ""
child_node = "Operation-1"
END_INSTANCE

START_INSTANCE
ALTERNATE_OPERATION_REMOVAL_SEQUENCE_KCP "Operation-1"
description = ""
child_node = "Operation-1 : Alternative-1"
END_INSTANCE

START_INSTANCE
OPERATION_REMOVAL_SEQUENCE_KCP "Operation-1 : Alternative-1"
description = "Work Instructions for Operation-1 : Alternative-1"
child_node = "Tool Usage Group-7"
child_node = "Tool Usage Group-8"
child_node = "Center Drill-1"
child_node = "Drill-2"
child_node = "Bore-3"
child_node = "Ream-4"
child_node = "Rough End Mill-5"
child_node = "Finish End Mill-6"
END_INSTANCE

START_INSTANCE
TOOL_USAGE_GROUP_KCP "Center Drill-1"
description = "TOOL : Center Drill-1"
child_node = "Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
child_node = "Center Drill-1:Hole-02:Micro-1:C'Drill-1:Cut-2:1"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_ACTION_KCP "Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
description = "C-Drill 0.161 in. Dia. Hole-01 to 0.1127 in. Dia."

FEATURE_CUT_KCP = "Hole-01:Micro-1:C'Drill-1"
working_step_offset_type =drilling_workingstep
MANUF_TOOL_RESOURCE_KCP "tl301"
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
MANUF_DATA_RESOURCE_KCP = "manuf_data:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
NC_FEATURE_KCP = "Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1:NC Hole-1"
END_INSTANCE

START_INSTANCE
NC_FEATURE_KCP = "Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1:NC Hole-1"
type = NC_ROUND_HOLE_KCP
FEATURE_CUT_ACTION_KCP = "Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
description = "flat"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.05635
max_diameter = 0.1127
taper_angle = 1.5708
delta_volume = DV1_Hole-01C'Drill-1.stp
in_process_model = IP1_Hole-01C'Drill-1.stp
END_INSTANCE

START_INSTANCE
TOOL_USAGE_GROUP_KCP "Drill-2"
description = "TOOL : Drill-2"
child_node = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
child_node = "Drill-2:Hole-02:Micro-1:Drill-1:Cut-2:1"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_ACTION_KCP "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
description = "Drill 0.161 in. Dia. Hole-01 to 0.141 in. Dia. and a Depth of 0.340 in."
FEATURE_CUT_KCP = "Hole-01:Micro-1:Drill-1"
working_step_offset_type =drilling_workingstep
MANUF_TOOL_RESOURCE_KCP "tl302"
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Drill-2:Hole-01:Micro-1:Drill-

1:Cut-1:1"
MANUF_DATA_RESOURCE_KCP = "manuf_data:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
NC_FEATURE_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1:NC Hole-3"
END_INSTANCE

START_INSTANCE
NC_FEATURE_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1:NC Hole-3"
type = NC_ROUND_HOLE_KCP
FEATURE_CUT_ACTION_KCP = "Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
description = "flat"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.16
max_diameter = 0.141
taper_angle = 0
delta_volume = DV1_Hole-01Drill-1.stp
in_process_model = IP1_Hole-01Drill-1.stp
END_INSTANCE

START_INSTANCE
TOOL_USAGE_GROUP_KCP "Bore-3"
description = "TOOL : Bore-3"
child_node = "Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
child_node = "Bore-3:Hole-02:Micro-1:Bore-1:Cut-2:1"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_ACTION_KCP "Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
description = "Bore 0.161 in. Dia. Hole-01 to 0.151 in. Dia. and a Depth of 0.340 in."
FEATURE_CUT_KCP = "Hole-01:Micro-1:Bore-1"
working_step_offset_type =drilling_workingstep
MANUF_TOOL_RESOURCE_KCP "tl303"
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
MANUF_DATA_RESOURCE_KCP = "manuf_data:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
NC_FEATURE_KCP = "Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1:NC Hole-5"

END_INSTANCE

START_INSTANCE
NC_FEATURE_KCP = "Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1:NC Hole-5"
type = NC_ROUND_HOLE_KCP
FEATURE_CUT_ACTION_KCP = "Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
description = "flat"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.16
max_diameter = 0.151
taper_angle = 0
delta_volume = DV1_Hole-01Bore-1.stp
in_process_model = IP1_Hole-01Bore-1.stp
END_INSTANCE

START_INSTANCE
TOOL_USAGE_GROUP_KCP "Ream-4"
description = "TOOL : Ream-4"
child_node = "Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
child_node = "Ream-4:Hole-02:Micro-1:Ream-1:Cut-2:1"
END_INSTANCE

START_INSTANCE
FEATURE_CUT_ACTION_KCP "Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
description = "Ream 0.161 in. Dia. Hole-01 to Depth of 0.340 in."
FEATURE_CUT_KCP = "Hole-01:Micro-1:Ream-1"
working_step_offset_type =drilling_workingstep
MANUF_TOOL_RESOURCE_KCP "TL304"
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
MANUF_DATA_RESOURCE_KCP = "manuf_data:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
NC_FEATURE_KCP = "Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1:NC Hole-7"
END_INSTANCE

START_INSTANCE
NC_FEATURE_KCP = "Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1:NC Hole-7"
type = NC_ROUND_HOLE_KCP
FEATURE_CUT_ACTION_KCP = "Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"

description = "flat"
top_center_location.x = 4.24453
top_center_location.y = -9.80979
top_center_location.z = 5.82072
direction.i = 0
direction.j = 0
direction.k = 1
max_side_depth = 0.16
max_diameter = 0.161
taper_angle = 0
delta_volume = DV1_Hole-01Ream-1.stp
in_process_model = IP1_Hole-01Ream-1.stp
END_INSTANCE

START_INSTANCE
MANUF_TOOL_RESOURCE_KCP "tl301"
description = "0.2 DIA CENTER DRILL"
tool_type = twist_drill
units = INCHES
diameter = 0.2
assembly_length = 2
hand = right
number_of_teeth = 2
included_angle = 90
flute_length = 1
END_INSTANCE

START_INSTANCE
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
feed_value = 0.00211667
value_unit = METER_KCP
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
speed_value = 0.0833333
value_unit = REVOLUTIONS
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
MANUF_DATA_RESOURCE_KCP "manuf_data:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
type = center_drilling
axial_cut_depth = 0.05635

retract_plane:location.x = 4.24453
retract_plane:location.y = -9.80979
retract_plane:location.z = 5.83072
retract_plane:z_vector.i = 0
retract_plane:z_vector.j = 0
retract_plane:z_vector.k = 1
retract_plane:x_vector.i = 1
retract_plane:x_vector.j = 0
retract_plane:x_vector.k = 0
secure_plane:location.x = 4.24453
secure_plane:location.y = -9.80979
secure_plane:location.z = 5.83072
secure_plane:z_vector.i = 0
secure_plane:z_vector.j = 0
secure_plane:z_vector.k = 1
secure_plane:x_vector.i = 1
secure_plane:x_vector.j = 0
secure_plane:x_vector.k = 0
END_INSTANCE

START_INSTANCE
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Center Drill-1:Hole-01:Micro-1:C'Drill-1:Cut-1:1"
coolant = on
END_INSTANCE

START_INSTANCE
MANUF_TOOL_RESOURCE_KCP "tl302"
description = "0.141 DIA DRILL"
tool_type = twist_drill
units = INCHES
diameter = 0.141
assembly_length = 2
hand = right
number_of_teeth = 2
included_angle = 120
flute_length = 1
END_INSTANCE

START_INSTANCE
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
feed_value = 0.00211667
value_unit = METER_KCP
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE

DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
speed_value = 0.0833333
value_unit = REVOLUTIONS
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
MANUF_DATA_RESOURCE_KCP "manuf_data:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
type = drilling
axial_cut_depth = 0.34
retract_plane:location.x = 4.24453
retract_plane:location.y = -9.80979
retract_plane:location.z = 5.83072
retract_plane:z_vector.i = 0
retract_plane:z_vector.j = 0
retract_plane:z_vector.k = 1
retract_plane:x_vector.i = 1
retract_plane:x_vector.j = 0
retract_plane:x_vector.k = 0
secure_plane:location.x = 4.24453
secure_plane:location.y = -9.80979
secure_plane:location.z = 5.83072
secure_plane:z_vector.i = 0
secure_plane:z_vector.j = 0
secure_plane:z_vector.k = 1
secure_plane:x_vector.i = 1
secure_plane:x_vector.j = 0
secure_plane:x_vector.k = 0
END_INSTANCE

START_INSTANCE
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Drill-2:Hole-01:Micro-1:Drill-1:Cut-1:1"
coolant = on
END_INSTANCE

START_INSTANCE
MANUF_TOOL_RESOURCE_KCP "tl303"
description = "0.151 DIA BORE"
tool_type = bore
diameter = 0.151
assembly_length = 2
number_of_teeth = 4
flute_length = 1
hand = right
END_INSTANCE

START_INSTANCE
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
feed_value = 0.00211667
value_unit = METER_KCP
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
speed_value = 0.0833333
value_unit = REVOLUTIONS
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
MANUF_DATA_RESOURCE_KCP "manuf_data:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
END_INSTANCE

START_INSTANCE
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Bore-3:Hole-01:Micro-1:Bore-1:Cut-1:1"
coolant = on
END_INSTANCE

START_INSTANCE
MANUF_TOOL_RESOURCE_KCP "TL304"
description = "0.161 DIA REAMER"
tool_type = ream
diameter = 0.161
assembly_length = 2
number_of_teeth = 4
flute_length = 1
hand = right
END_INSTANCE

START_INSTANCE
DEFAULT_FEEDRATE_TECH_ITEM_KCP "feedrate:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
feed_value = 0.00211667
value_unit = METER_KCP
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
DEFAULT_SPINDLE_TECH_ITEM_KCP "spindle_speed:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"

speed_value = 0.0833333
value_unit = REVOLUTIONS
time_unit = SECOND_KCP
END_INSTANCE

START_INSTANCE
MANUF_DATA_RESOURCE_KCP "manuf_data:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
END_INSTANCE

START_INSTANCE
MACHINE_FUNCTION_RESOURCE_KCP "machine_function:Ream-4:Hole-01:Micro-1:Ream-1:Cut-1:1"
coolant = on
END_INSTANCE

Appendix B: Data model of UG CAM objects

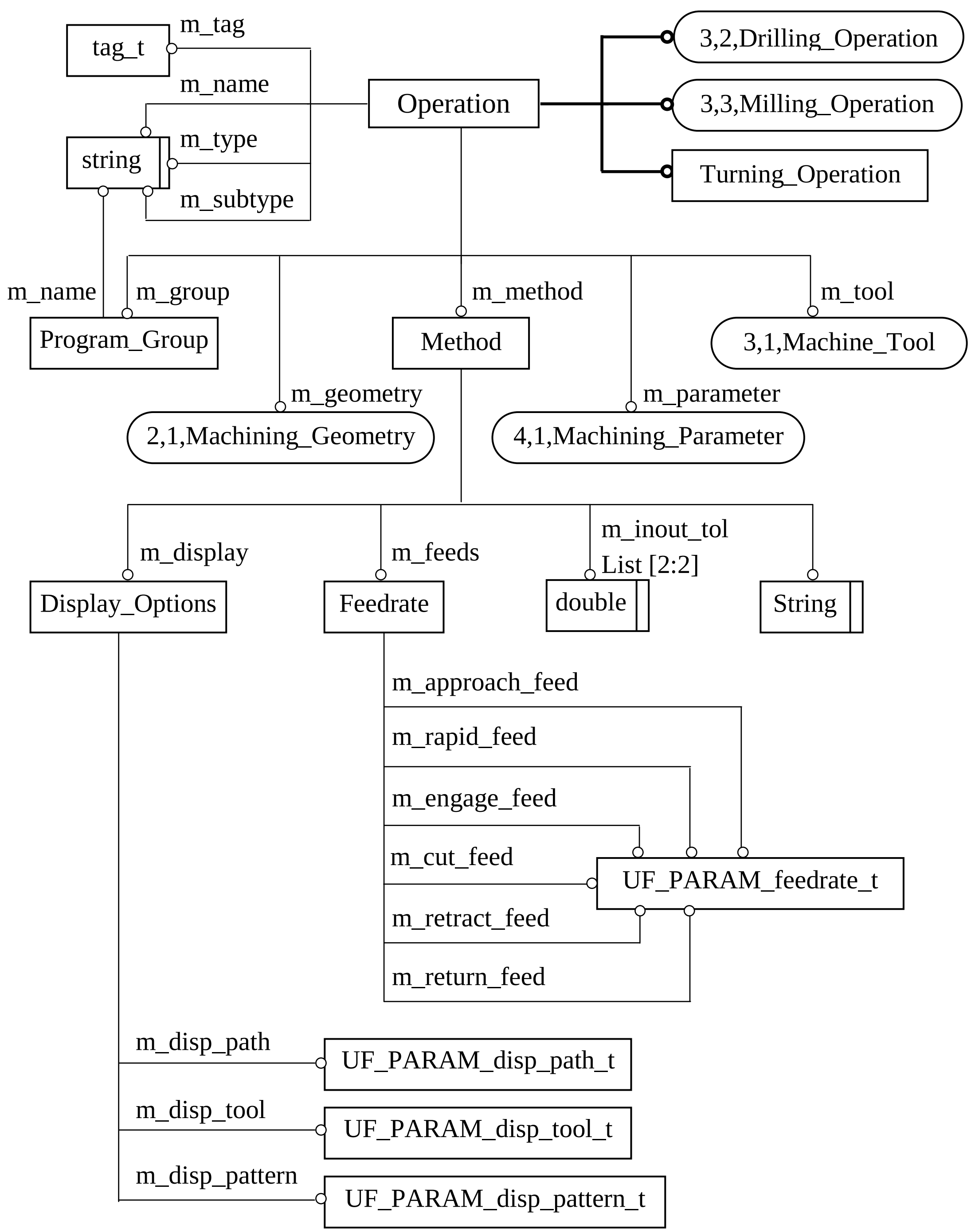


Figure B.1 Page 1 of 5: data model of UG CAM objects

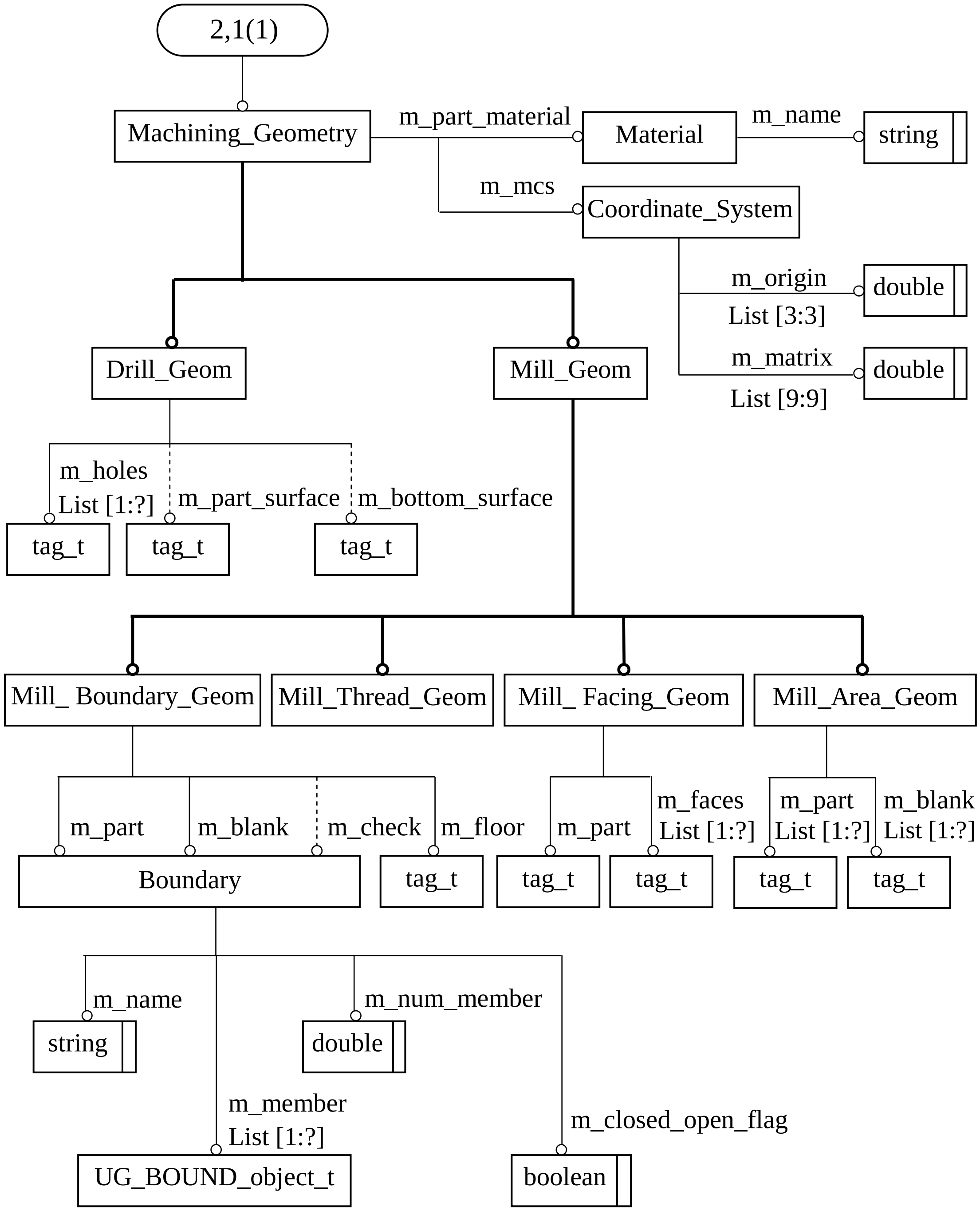


Figure B.2 Page 2 of 5: data model of UG CAM objects

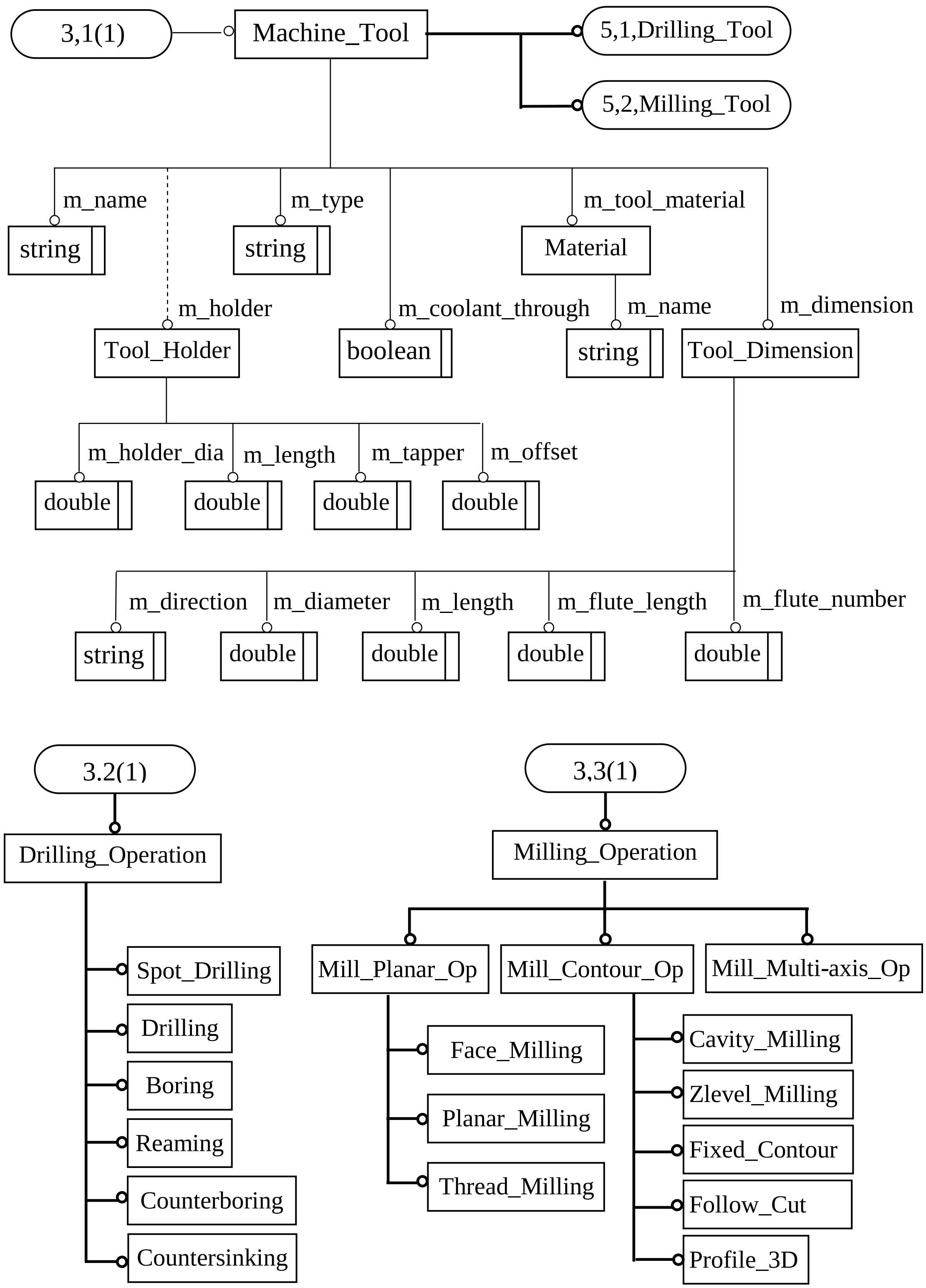


Figure B.3 Page 3 of 5: data model of UG CAM objects

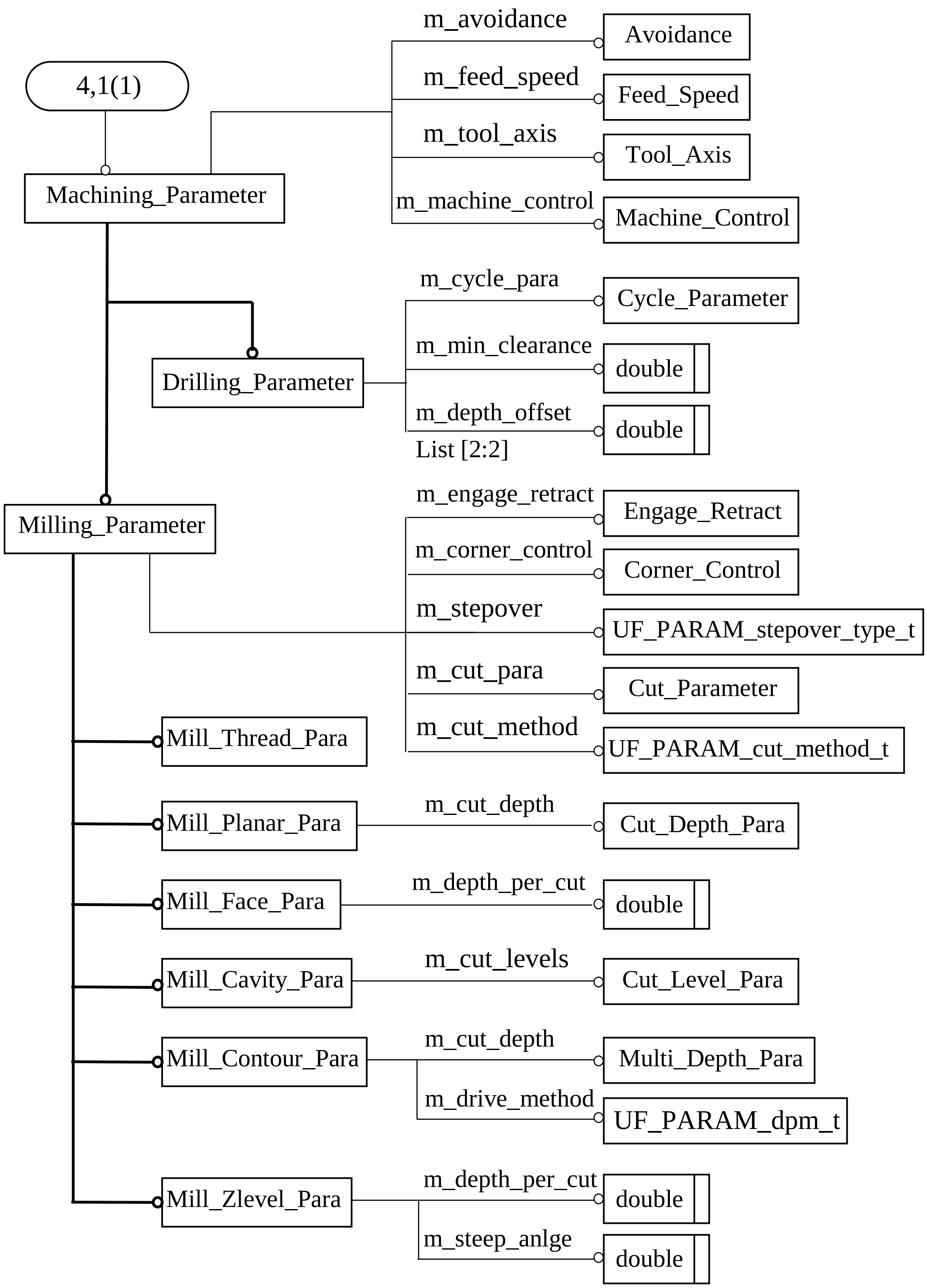


Figure B.4 Page 4 of 5: data model of UG CAM objects

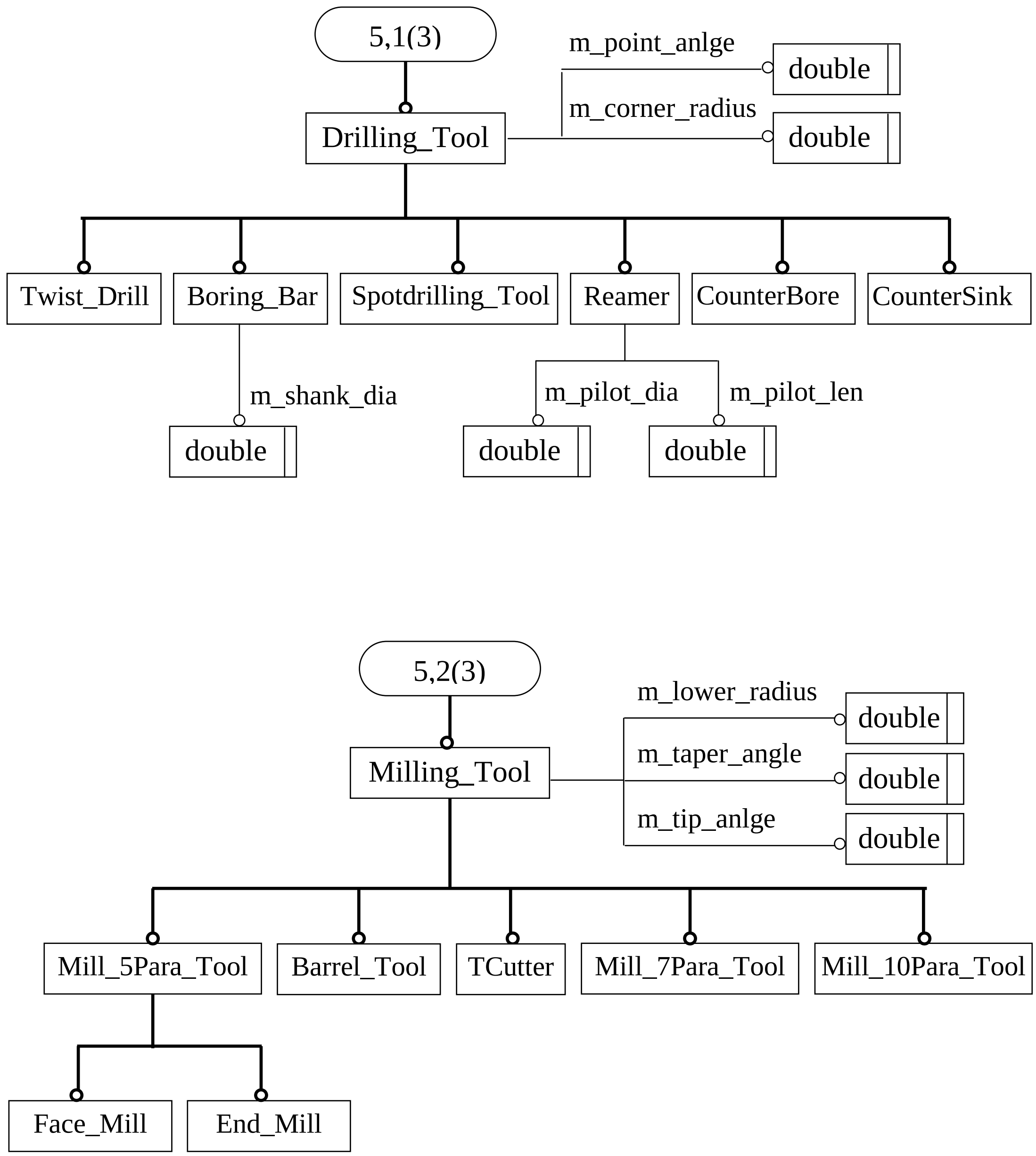


Figure B.5 Page 5 of 5: data model of UG CAM objects