Sung H. Ahn

V. Sundararajan

Charles Smith

Balaji Kannan

Roshan D'Souza

Ganping Sun

Ashish Mohole

Paul K. Wright

Mechanical Engineering Dept.,
University of California at Berkeley,
Berkeley, CA 94720

JaeHo Kim

Sara McMains

Jordan Smith

Carlo H. Séquin

Computer Science Dept.,
University of California at Berkeley,
Berkeley, CA 94720

CyberCut: An Internet-based CAD/CAM System

''CyberCut™'' is a testbed for an Internet-based CAD/CAM system. It was specifically designed to be a networked, automated system, with a seamless communication flow from a client-side designer to a server-side machining service. The creation of CyberCut required several new software modules. These include: a) a Web-based design tool in which Design-for-Manufacturing information and machining rules constrain the designer to manufacturable parts; b) a geometric representation called SIF-DSG, for unambiguous communication between the client-side designer and the server-side process planner; c) an automated process planning system with several sub-modules that convert an incoming design to a set of tool-paths for execution on a 3-axis CNC milling machine. Using this software-pipeline, a CyberCut service, modeled on the MOSIS service for VLSI chips, has been now been launched for limited student-use at a group of cooperating universities.
[DOI: 10.1115/1.1351811]

1 Introduction

In 21st century manufacturing, the 20th century concept of a monolithic organization clinging to one centralized corporate may fade. The new culture may well be smaller, more agile corporations that spring up for specific purposes, exist while the market sustains the new product, and then gracefully disband as the market changes [1]. In other words, the roles in different stages of a product development cycle, usually played by specific departments in a large corporation, now may be played by smaller networked companies in various locations. The rapidly expanding Internet provides the information infrastructure for such new manufacturing enterprises. However, it also creates more challenges in the traditional communications between design and manufacturing, that are often colloquially referred to as ''over the wall manufacturing.'' While the Internet has the potential to integrate many sub-contractors, this ''wall'' can often be even higher for the following reasons:

1 Designers and process planners belong to different companies. Even an experienced designer may not exactly know the process capacity in another company, and, within that company, it may be also difficult for a process planner to guess the original designer's intention.

2 A part sent for manufacturing may be designed by a less experienced designer who does not have much process knowledge. Today's commercial CAD tools allow designers to design geometrically sophisticated parts, but seldom concern themselves with the manufacturability issues. The parts designed with these CAD tools must later be examined and often modified by experienced manufacturing engineers to ensure problem-free manufacturing. Such design-redesign iterations can require a large amount of time and effort, and result in a longer product development cycle.

3 The commercial CAD/CAM systems used by designers and process planners may be quite different. **Internet**-based manufacturing needs to cope with this heterogeneous design and manufacturing environment. This is much more demanding than an **Intranet** environment where one standard might be enforced within one company (i.e. high-level management insists that all designers must use one chosen CAD package and all process planners must use one companywide CAM package).

Another significant time sink in product development is the process planning work on the manufacturing side. Most commercial CAM software packages provide generative machining tools [2,3]. Using such systems, manufacturing engineers specify the machine type, setups, fixtures, operations, surfaces to be machined, machining coordinates, cutting tools, and cutting parameters (Fig. 1*a*). The main advantage of this generative machining procedure is that an engineer who is close to the actual equipment being used can control many details of the machining process to obtain specific, possibly tight, tolerances. On the other hand, this intensely manual/craftsperson approach can take a long time, several hours to days, to complete the process planning. It takes even longer if the design is difficult to manufacture and some changes to the original design are needed.

In summary, there is a growing awareness that manufacturing information should be presented to the designer during the early phases of part creation. While this has always been the goal of Concurrent Engineering, the use of the Internet and globally distributed supply chains makes even more demands for ''process-

Contributed by the Computer Aided Product Development Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Sept. 2000; revised manuscript received Jan. 2001. Associate Editor D. Dutta.

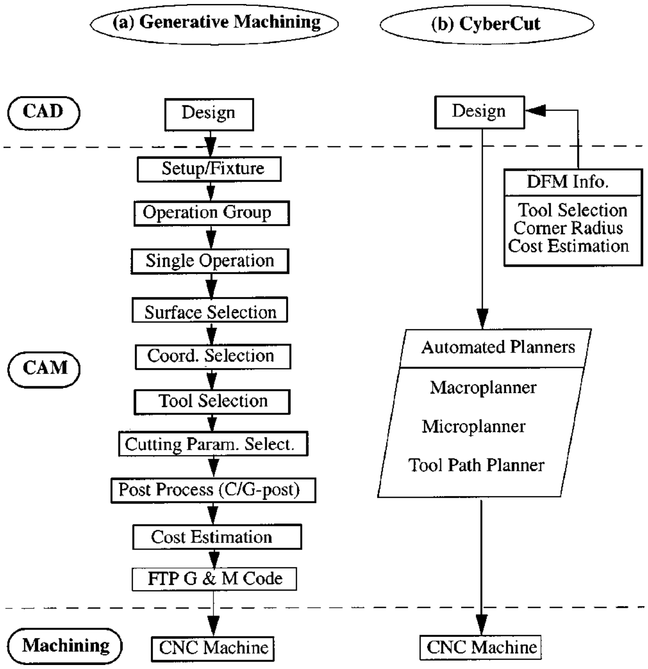


Fig. 1 Comparison between Generative Machining of (*a*) typical CAD/CAM tools and (*b*) CyberCut. Cybercut's automated process planning requires less human interaction, resulting a great saving in process planning time

aware CAD/CAM.'' For example, manufacturability checks ideally need to be incorporated into the ''design-side'' CAD system, that detect problems in the design and prevent an unmanufacturable part from being sent to a fabrication facility. Automation of process planning on the ''manufacturing side'' has also received considerable attention in industry and in the academic environment.

2 Literature Review

2.1 Design for Manufacture (DFM) and Manufacturability Evaluation. Hisao [4] implemented a prototype manufacturability evaluation system for CAD/CAM which determined the manufacturability on a feature by feature basis in two phases. In the first phase a qualitative evaluation consisting of checks for good practice rule violation was done and results were fed back to the designer. In the second phase, a quantitative evaluation was done by searching for the cheapest feasible sequence of processes using branch and bound technique. The results of the second evaluation include suggestions to alter the design with the corresponding cost benefit. Cutkosky and Tenenbaum [5] developed the First-Cut and Next-Cut interactive frame works for design of products on the basis of process planning. Processes planning information was embedded at the design phase by using manufacturing features for design. Designers could create a design by subtracting volumetric machining features corresponding to machining operations from a piece of stock material. As features were subtracted, the system used its knowledge base to analyze its manufacturability. If any constraints were violated, the designer was warned of the violating features. Gupta et al. [6] described a methodology for early manufacturability evaluation of prismatic parts. All machining operations which could be used to create the part were identified. Using these operations, different operation plans were generated. For each new operation plan, the system examined whether the plan could produce the desired shape and tolerances. If the plan was capable of doing so, the manufacturability rating based on the estimated machining time for the part was calculated. Das et al. [7] developed a methodology based on the above mentioned work for suggesting improvements to a given design to reduce the number of setups to machine a part. Their work involved different machining operations to satisfy the geometric constraints put on the part by the designer. These constraints reflected the functionality of the part. Feng and Kusiak [8] proposed an object oriented scheme for machining constraints. Design by machining features of the type restricted by machining constraints were considered. Although the constraints that were considered were not original, their approach of applying these constraints at an early design stage was novel. Machining constraints were classified as geometric, machining resource, machining condition and machining precedence constraints. In their prototype system features were represented as objects. Constraints were represented as production rules that were used in evaluating the manufacturability of the features.

2.2 Feature Data Exchange. Feature data exchange is motivated by the need to have different feature based systems to interact with each other. Dunn [9] developed Part-48, a STEP draft that provided a general purpose form feature data model. Form features were classified into 3 basic classes namely volume feature, transition feature and feature pattern. Volume features could either be additive volumes or subtractive volumes. Transition features could be edge transition features or corner transition features. Feature patterns were further classified as circular, array and other. The focus of this work was to classify all possible features and feature patterns into one of the predefined classes. Subsequently, Part-224 was developed by Slovensky [10] for form features specific to computer-aided process planning. In Part-48 no unique representation of a feature is provided. Also, it deals with nominal shape only. On the other hand, Part-224 parametrizes features in a unique way. It also includes tolerance, material and surface properties, and administrative and production control data.

The feature models discussed previously do not have a unified concept of a feature. This implies methods have to developed to analyze each of these classes. It is difficult to incorporate any feature that does not conform to the predefined shapes in the feature data model. The issue of associating process planning information like tools and cutting parameters with feature is also not addressed. Some progress has been made in the present work to address these ambiguities.

2.3 Process Planning. Many process planning systems use a *feature-based approach*, because machining features often provide a convenient mapping between part design and machining processes. For example, MCOES (Manufacturing Cell Operator's Expert System) described in [11] was one of the earliest feature-based design and planning systems for short batch production that used feature-based part family models. A variant feature approach was adopted so that varying levels of detail and granularity could be used in the feature model for the part families. The system consisted of a design data interface, a generative process plan preparation system and an operative process planning system. The design interface supported feature-based modeling of part families. The generative planner allowed manufacturing processes to be described and related to part-family models. The operative planner generated process plans and NC code based on part family descriptions.

PART (Planning of Activities, Resources and Technology) [12] was another early integrated feature-based process planning system, which consisted of a fixture planning system, FIXES [13]. This was a generative fixture planning system for prismatic parts. FIXES was responsible for both setup selection as well as fixture design for each setup. Setups were automatically selected by comparing tolerances between different features of the part as well as their orientation. A setup consisted of features with the smallest tolerances and acceptable machining directions with respect to the axis configuration of the selected machine tool.

As a third example, in a system developed at the University of

Maryland [7] feature-based models (FBMs) were generated from the solid model of the part by feature recognition. Every FBM represented each feature in a single direction. The use of FBMs simultaneously with the solid model of the part allowed their system to consider multiple representations of various features for both machining operation planning and automated fixture planning. In many cases a large number of these FBMs had to be evaluated to arrive at a good and feasible process plan. This was a sequential approach to process planning, in that feature recognition takes place prior to and independent of process planning.

MAPP (Matrix Architecture for Process Planning) [14] was a system that captured a sequence of planning phases that cut across functional boundaries while still allowing organization of data into important functions. There were various problem solving phases in MAPP, such as identifying goals, planning between setups and planning within setups. Each of these phases reads or writes data on any of the blackboards corresponding to the data functions like fixturing, tools or setups. In order for information to flow through this system throughout all the planning phases, the CAD and tool databases were made accessible to each of the problem-solving phases. In particular, the CAD database was used by all phases while the tool database was most heavily used by phase that plans within setups. The problem-solving phases were divided between two modules called MEDIATOR and COORDINATOR. MEDIATOR [15] performed feature recognition and made some early process planning decisions such as possible tool and fixture combinations for each feature. COORDINATOR performed the detailed process planning functions like tool selection. This system thus simultaneously identified features and generated one or more manufacturing methods to machine each feature.

A different approach to process planning was taken by Shirur et al. [16], and Hirode and Shah [17]. A model of the volumes that could be machined was first created by defining abstractly the shapes that a given combination of cutting tool and machine could achieve. This abstraction was represented by means of an algebraic expression that specified a closed profile and a sweep operator along the access direction. An inverse operator was used to map the machining volume back to the machining process by performing a degree of freedom analysis on the machining volume. The authors referred to this as a process-based approach rather than a feature-based approach to emphasize the fact that inverse mapping from process capability to machining volumes was done instead of pre-defining features.

Sakurai [18] implemented a system that automatically planned the setup sequence and generates fixture configurations for each setup from the toleranced solid model of the finished component. Fixturing schemes were synthesized with a major emphasis on accurate location and stability, with kinematic analysis of stability being done using screw theory. Heuristics were used to determine the setup directions and sequence of setups.

The Quick Turnaround Cell (QTC) had a feature-based process planner [19] that considered fixture planning as an integral part of process planning. The setup planner was knowledge-based and the fixture-planner was interactive. The fixture planner looked at all the available resources in a workshop and created all details for multiple setup information, including a complete fixture plan, machining operation plans and tool selection. Both vise-related and modular fixturing methods were part of their fixture planner.

Gupta and Nau [20] presented an approach to automatically analyze the manufacturability of machined parts. They identified the various Feature Based Models (FBMs) that could be used to create the part. Then, precedence constraints were generated using parent-child relationships and feature-minimality conditions. By mapping each feature to a machining operation, their system automatically generated operation plans for each FBM. Finally a module that estimated the machining accuracy achievable for each operation was applied to each of these operation plans to determine the operation plan that was capable of producing the desired shape and tolerances with the greatest accuracy. If no such operation plan could be found, the part was non-manufacturable.

3 Internet-Based CAD/CAM

3.1 Previous Work. The Integrated Manufacturing and Design Environment (IMADE) was the first Internet enabled design and manufacturing system for CNC machining [21]. Parts designed and viewed in a 3D wire-frame on a client machine could be sent across the Internet for remote fabrication. However, the client machine needed a variety of non-commercial, development software packages: Noodles (a freeware geometry kernel), Tcl/Tk to run the graphical user interface, a LISP interpreter, and the permission to execute Unix shell script that would FTP the final design file to the machinist. The complexity of the software installation was a significant problem with IMADE, inhibiting outside use. Also, the proprietary feature description language was incompatible with existing G&M codes and only supported rectangular pockets and holes. Decisions on fixturing were delegated to the machinist. Finally, there was minimal optimization of the feature order, resulting in lengthy air cuts.

3.2 Internet-Based CAD/CAM. CyberCut introduces two factors that enable automated handling of the design to manufacturing cycle: (1) process-aware CAD and (2) automated process planning. The advantages and significance of this paradigm in design and manufacturing are described in the following sections. The goal of the CyberCut project was not to create yet another CAD package. Instead, the goal was to tightly couple the entire concept-to-fabrication sequence to 3-axis milling. A proprietary, Web-based, Java CAD tool (WebCAD) [22] was chosen for the following four reasons:

1. **No licensing fees**—in order to encourage collaboration with other organizations, CyberCut is not built on a commercial CAD program. This avoids forcing collaborators to purchase any particular software.
2. **Automation focus**—A custom CAD tool called WebCAD allowed the research project to focus more on the *automation* of the complete pipeline and less on *integration* with legacy software. By contrast, commercial CAD products are often complex and/or deny the programmer access to certain levels of the program's functionality or data structures.
3. **User interface control/design rule integration**—WebCAD enforces manufacturability rules in real time, at a level appropriate to a remote client-designer.
4. **Brokerage model**—CyberCut's ''manufacturing-side'' process planner can now be set up as a brokerage service that would accept a variety of manufacturing requests and distribute process plans to a variety of machine shops. Having a purely web-based CAD tool allows for easy testing of this model.

WebCAD provides an ideal research platform for implementing manufacturability constraints at the user interface level. It also creates designs suitable for downstream process planning and milling. The biggest challenge for WebCAD was to provide a usable software environment capable of creating non-trivial part designs while maintaining a small download size. As a result the current version of WebCAD is not yet appropriate for commercial use. Commercial designers prefer a larger feature set and more flexibility. Geometry definition is much less restrictive with commercial tools, and third party analysis packages (such as finite element stress modeling) are designed around existing CAD programs. In spite of the limitations, the early version of CyberCut/WebCAD is a useful paradigm as a prototype system for future Internet-based CAD/CAM systems with more power.

3.2.1 Features in CyberCut. In CyberCut/WebCAD, a part is described in terms of a stock and 2.5D features. In contrast to all the feature-based CAD/CAM systems cited above, *CyberCut/*

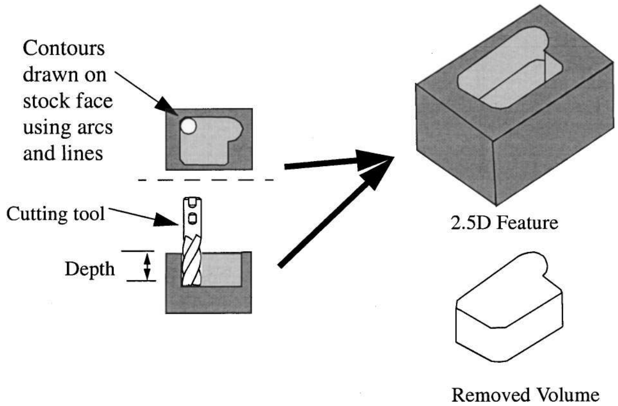


Fig. 2 Definition of features

WebCAD does not make use of a standard feature library. Instead, a 2.5D feature is described in terms of a 2D contour, access direction and depth. Figure 2 illustrates this description of features. Such a general definition of features allows CyberCut to classify features as just milling and drilling features. All milling features, no matter what the shape of their 2D contour is, are referred to as pockets. Drilling features are referred to as holes. Thus pockets in CyberCut encompass all the standard 2.5D milling features such as square/rectangular pockets, steps, slots and shoulders. In addition, features in CyberCut also contain information about corner radii, open edges and nesting, which are essential for downstream process planning. The concepts of nesting and open edges are illustrated in Fig. 3.

An ''open edge'' in the contour of a feature indicates that the cutting tool must cross this edge to the exterior of the feature to fully remove the volume. Thus, features like slots and steps have open edges.

It is seen from Fig. 3 that features that can be accessed from a single direction form a tree, such that the contour of a ''child feature'' is fully contained within the projection of contour of the ''parent feature'' and the child feature is beneath the parent in the component. This arrangement makes it easy to sequence features during process planning, since parent features must be machined before the children.

It should be noted that all information about the features such as geometry, corner radii, nesting and open edges are captured in WebCAD during the design phase itself. *This obviates the need for a feature recognition system*. The stock geometry and the information that the design tool obtains about the features from the designer are the inputs to the process planner.

In extensions being done to CyberCut, freeform surfaces are also being incorporated. Thus, in the extended version, chamfers, corner rounds, fillets as well as pockets with freeform surfaces will be available.

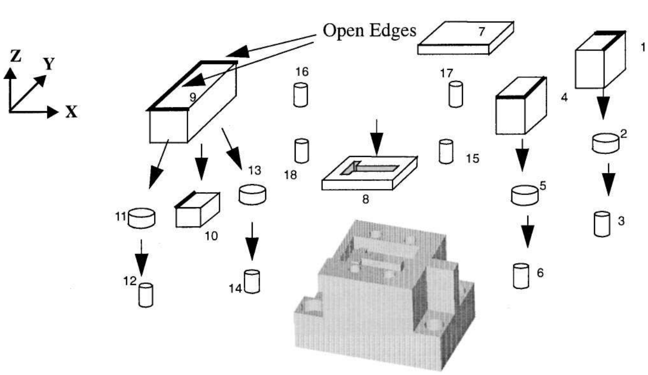


Fig. 3 Nesting of features

Table 1 Manufacturing processes and suitable design methods

Manufacturing Process	Method of Geometry Creation
Milling Machine	**Destructive Solid Geometry** (DSG) - The only operation available to a design is the *removal* of material from the original stock. The mimics the cutting process.
Injection Molding, Sheet Metal Forming	**Shelling** - Automatically generates a part with a thin, uniformly thick wall from a prismatic solid.These manufacturing processes require or benefit from having a uniform wall thickness.
Turning	**Surface of Rotation** - Sweeping a curve around an axis to generate a rotationally symmetric solid.

3.2.2 Process-Aware CAD. The process-aware WebCAD program (CyberCut's ''front-end'') has detailed knowledge of the particular manufacturing process of milling ''built-in.'' Rather than allowing designers to create ''fanciful geometries,'' the software attempts to ensure that the user is guided towards the creation of a machinable design. The result is a part geometry that has already been checked for manufacturability before any attempt is made to create a detailed manufacturing plan. This is one step towards avoiding ''over-the-wall'' manufacturing.

Process aware CAD tools have also been developed for sheet metal forming, and net shape processes as summarized in Table 1 [23,24], but since manufacturability criteria differ widely from process to process, it was deemed beyond the scope of this project to try to incorporate more than milling into CyberCut.

3.2.3 Automated Process Planning. CyberCut process planner is comprised of three levels; the macroplanner, the microplanner and the tool path planner (Fig. 1*b*). The macroplanner takes a global view of the part and considers interactions among features. It is responsible for determining sequences of setups (and operations within setups) while minimizing fabrication time. The microplanner looks at single features and determines the one or more operations needed to cut a certain feature. Finally, each operation plan is comprised of a cutting tool and associated cutting parameters (cutting feed, spindle RPM, width and depth of cut). The sequenced operations are then sent to a tool-path planner to lay out the paths for the cutting tool on the CNC machine. Process planning in CyberCut is performed without much human interaction. Although these planning results might not be the global optimum in terms of production time, the automation of planning saves a great portion of the time needed during prototyping. Full scale production would naturally have different constraints which are beyond the scope of this work.

4 Components of the CyberCut Pipeline

4.1 WebCAD. WebCAD (Fig. 4) is a 3D wireframe design program, which begins with a rectilinear block and lets users remove material from one of the original faces of the stock, mimicking the milling process. Every modification includes the selection of a real and available cutting tool. If the designer chooses to drill a hole, the WebCAD system insists that the user defines a depth and selects a standard drill bit that is capable of drilling to that depth. If the designer wants a hole 2 inch deep, WebCAD *restricts* the drill choices to tools that are 2 inches or longer.

Similarly, for non-drilled features, the designer selects a depth and a milling cutter that is long enough for the depth. The user then draws a contour on any of the surfaces of the original stock (currently, only circular arcs and line segments are available). As illustrated in Fig. 2, the contour is extruded to the requested depth and removed from the stock. To emphasize that the feature is cut with an end mill, sharp internal corners on the finished contour are rounded to show the true shape that will be cut by the cylindrical tool.

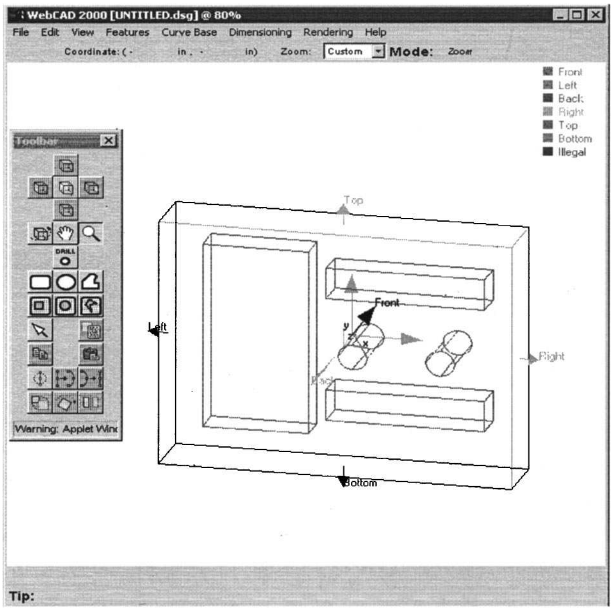


Fig. 4 WebCAD display

If the feature contour is self intersecting, or contains regions too narrow for the selected tool to enter, WebCAD warns the designer to correct the contour geometry. Another check ensures a minimum distance (0.01 inches) between features to prevent the creation of thin walls that might deform or break under machining forces. These design methods and rules constrain design choices, helping a designer compose manufacturable parts. Between the method of drawing the part and the feedback, WebCAD can be classified as feature-based, Constrained Destructive Solid Geometry (CDSG) modeler.

WebCAD provides a standard set of editing and graphic functions (pan, zoom, cut, copy, paste, mirror, rotate, and scale), submission to a server-side VRML renderer, and drawing ''helpers'' (grid, snap, simple dimensioning). Other features commonly included in commercial CAD programs (i.e. lofting, extrusion, surface of revolution) are *not* available in WebCAD, as they make it easier for designers to create parts unsuitable for milling.

Finally, a WebCAD part described in pure DSG format can be submitted to the remote process planner that utilizes a BRep-based solid modeler, ACIS. The design information is communicated to the process planner via a new interchange format called SIF-DSG (Solid Interchange Format—Destructive Solid Geometry) [25]. The format captures the essential aspects of the design that is required for process planning. It is specifically designed to be compact to allow rapid transfer over the network. Based on the process plans generated, the user can get a cost estimation. Standard procedures for cost estimation are used as those in Ostwald [26]. The design interface is freely available at http://cybercut.berkeley.edu.

4.2 Macroplanning. The macroplanner performs the higher level tasks of setup sequencing and operation sequencing. The primary goal of this phase is to arrive at a feasible manufacturing plan that results in the minimum number of setup changes and tool changes. The inputs to the macroplanner are features defined in the WebCAD interface.

4.2.1 Precedence Constraints and Feature Interaction. The macroplanner categorizes constraints into 3 different levels, as described in [21]. *Level 0* constraints are those that cannot be violated without compromising the feasibility of machining the part. An example would be the parent-child constraint, where the parent feature provides access to the tool that is used to machine a child feature. *Level 1* constraints are those that affect the quality of the machined part, but are not really critical for manufacturing it. For example, a sequence of operations that would result in smaller burrs would be classified as a *Level 1* constraint. Finally, *Level 2* constraints are those that affect merely the efficiency of operation but not the quality of the machined part. For instance, when a pocket intersects a deeper, narrower pocket, it may be preferable to machine the larger pocket first, so that amount of cutting time is reduced.

It is now possible to organize setups, tool changes and the precedence constraints mentioned above into a hierarchy as shown below:

$$level\ 0 > setups > level\ 1 > level\ 2 > tool\ changes$$

This means that *level 0* constraints are more important than the number of setups, which in turn are more important than *level 1* constraints and so on. In other words, as many setups as needed will be used to machine the part in order that *level 0* constraints are not violated. But, *level 1* and *level 2* constraints could be discarded if the number of setups can be reduced. The rationale behind the last choice is that setup changes take up a large percentage of the time involved in manufacturing the part. Finally, tool changes are the lowest in the hierarchy, because they consume a relatively small portion of the time spent in manufacturing the part.

The process planners of CyberCut provide three fixturing options: conventional vise, toe clamps, and Reference Free Part Encapsulation (RFPE) [27–29]. The process planner analyzes potential interference between fixture and tool path for each fixturing technique, and a fixturing preference is given in the order of vise, toe clamps, and RFPE. Thus if all three fixturing techniques are possible, the vise will be selected as the preferred fixturing system.

4.2.2 Setup Sequencing Algorithms. The previous section illustrated how constraints between features can be obtained and how these constraints can be organized into a hierarchy. The importance of this hierarchy is that it permits an elegant decomposition of the core problem of macroplanning, namely optimization of the number of setups and operations. It is worth noting here that since the macroplanner gets all information about the features from the design tool, the access directions for all features are predefined. In other words, the access direction for each feature is the direction in which the designer designed it from. This enables the macroplanner to preserve the design intent as much as possible. Fixing the access direction for each feature also helps in generation of precedence constraints between various features as described in the previous section.

The macroplanner now identifies a directed graph called the *process graph*, in which the nodes are the features and the arcs are the constraints between the features. Each node contains information about its access direction. Once the process graph is generated, nodes with the same access direction are grouped together to form *hypervertices*. A new directed graph called the *setup graph* is generated, in which these hypervertices are the nodes. There is an arc between any two *hypervertices* if there is a constraint between any two nodes belonging to those *hypervertices*. Further details about these graphs and their properties are mentioned in [21].

There are a few facts about these graphs that are worth mentioning. First, due to the fact that the process graph is directed, the setup graph (the graph that consists of the *hypervertices*) is also directed. Second, if the process graph has any cycles, process planning cannot continue, since each node in the process graph maps onto the machining of a feature. So the part has to be redesigned for it to be manufacturable. Finally, the setup graph may have cycles. These cycles have to be dissolved or split in order to get a linear ordering of setups.

The rules for splitting the cycles are as below [21]:

(1) If the cycle is such that one of the arcs in the cycle is due to a *level 1* or *level 2* constraint, that arc is removed from the cycle. This is a direct consequence of the hierarchy proposed in the previous section.

(2) If all the arcs in the cycle are due to *level 0* constraints, setups are split in the cycle so that the cycle itself is broken.

The above procedure results in the formation of a Directed Acyclic Graph (DAG). A topological sort of this DAG results in reasonably optimal sequence of setups. When the setups have been sequenced, the features are sent one by one, along with an *in-process shape*, to the microplanner for operation planning. The in-process shape is the state of the component when the feature to be planned has not yet been removed. It can be calculated by subtracting the features already machined from the stock. The in-process shape serves as a bounding, temporary-geometry for the cutting tool motions and is used by the microplanner to detect collisions of the tool assembly with the workpiece.

For minimizing the tool changes within a particular setup, a modified version of the approach taken by Yang et al. [30] is adopted. Within each setup, precedence constraints between the various features are noted down and a *process-setup* graph is generated. Each node in this graph represents an operation within this setup. A node in this graph is said to have a *to-constraint* if a another operation is to be done before the operation corresponding to this node can be performed. A node is said to have a *from-constraint* otherwise.

For each node that has a *from-constraint* in the current *process-setup* graph, a set N, which is the union of sets N_1 and N_2 is determined, where N_1 and N_2 are defined below:

N_1 is the set of nodes in the current state of the *process-setup* graph that have only *from-constraints*.

N_2 is the set of nodes in the *process-setup* graph that have only *from-constraints*, after the current node is removed.

The set N for a node which is such that it contains the maximum number of nodes with the same tool is then determined. The nodes in this set are machined first. The above procedure is then continued recursively till there are no more operations to be done in that particular setup. It has been found that the above algorithms work very well for optimization of process plans.

4.3 Microplanning. Microplanning is concerned with selecting appropriate tools and the associated cutting parameters for each feature generated by the macroplanner. Features can either be holes or pockets. The tool needed for drilling a hole is selected based on the diameter and depth of the hole. For pockets, however, several combinations of tools are possible. The tools for a particular pocket are chosen such that the machining cost/time is a minimum.

Microplanning [31] is accomplished in three phases, (1) finding a feasible set of tools, (2) decomposing the feature into subfeatures to be machined using different tools and (3) selecting an optimal sequence of tools to machine with minimum cost/time. A concise tool data base schema has been developed that incorporates not only tool and tool holder geometry but also cutting parameters classified according to the type of operation (roughing, finishing etc.) and the work piece material.

A tool is feasible if its tool accessibility is not a null set. The smallest feasible tool is the one whose tool accessibility is either the pocket itself or closest to the pocket. Tool accessibility is defined as the region within a pocket that a tool can reach without gouging. Figure 5 illustrates the procedure to calculate tool accessibility. The tool accessibility of a given tool is larger than that of a tool with a larger diameter. Moreover, the tool accessibility of the tool with the larger diameter is a subset of the tool accessibility of the tool with smaller diameter. Therefore, the shape of the pocket after a particular tool has finished machining is independent of all tools that are used before it assuming that the tools are used in the descending order of diameter. Moreover, in a given tool sequence, the region to be machined by a particular tool is dependent only on the tool immediately proceeding it. Figure 6 illustrates the procedure to decompose a pocket into sub-pockets for any two tools. An associated problem with pocket decomposition is the issue of inter sub-pocket boundaries. These boundaries have to be traversed over by at least one tool in the tool pair. If not, slivers of material may be left at the boundaries. The boundaries of the sub-pockets assigned to the smaller tool in the tool pair are extended to cover inter sub-feature boundaries. Figure 6(*c*) illustrates inter sub-feature boundaries. The problem of finding the optimal tool sequence can be reduced to that of finding the shortest path in a single-source single-sink directed acyclic graph. Figure 7 illustrates a graph for 4 tools. The nodes in the graph represent the states of the stock just after the tool named in the stock has been machined. The edge represents the cost of machining the region assigned to the tool named in the end node of the edge. This cost includes the cost of machining, the cost of tool change and the cost of tool wear. The source represents the unmachined stock and the sink is the part after this particular pocket is completely machined. The smallest tool used is named in the sink.

At the end of microplanning, an operation plan consisting of a list of operations is obtained. Each operation is comprised of the decomposed sub-feature, the associated tool and the cutting parameters. The macroplanner puts together all the operation plans within a setup and tries to cluster operations with the same tool, so that tool changes can be minimized.

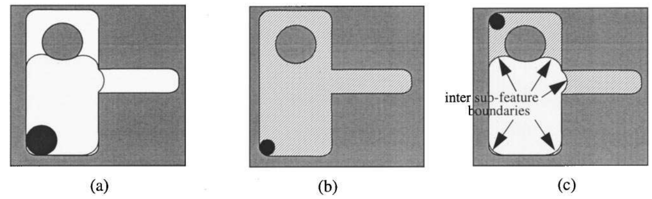


Fig. 6 Decomposing pocket: (*a*) Tool accessibility of larger tool in tool pair (*b*) Tool accessibility of smaller tool in tool pair (*c*) Region to be machined by smaller tool in tool pair is given by the boolean difference of the tool accessibility of smaller tool and tool accessibility of the larger tool

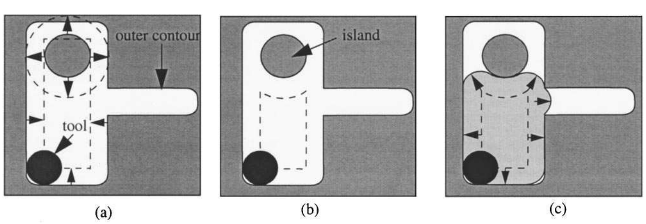


Fig. 5 Finding tool accessibility: (*a*) Shrink outer contour and grow island contour by a distance equal to tool radius (*b*) Subtract grown island contours from shrunk outer contour (*c*) grow the result by radius of the tool

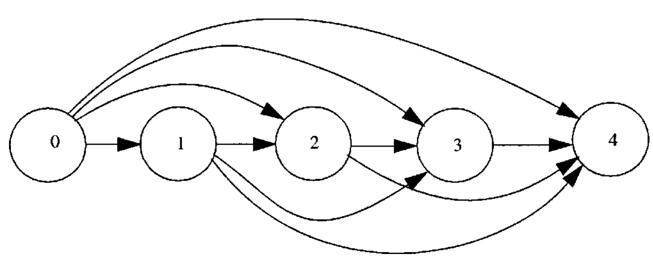


Fig. 7 Directed acyclic graph for 4 tools

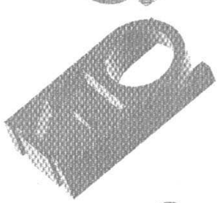
4.4 Tool Path Planning. The individual operations are the input to the tool path planner. Contour-parallel tool paths [32,33] are generated based on specified width of cut and depth of cut for the sub-feature in every operation. The feed rates specified are valid when the tool engagement is equal to the width of cut. However, when the tool plunges initially the engagement is equal to the diameter of the tool The tool path planner automatically identifies section of the tool paths where the tool engagement is equal to the diameter of the tool and adjusts the actual feed accordingly. It also identifies sections of the tool path that do actual machining and those that are used for traversing between disconnected regions. Actual machining times and air times are then calculated. This information is then used to calculate tool wear cost and cost of machine usage. The output of the tool path planner is a G&M code file that can be used to operate a Haas-VF0 3-axis NC machine. There is also a detailed feature by feature cost report that details cost of machine usage and tool wear which is fed back to the designer.

5 Conclusions

The convenience and speed of Internet-based communications create new opportunities for global commerce between product designers using CAD, and traditional manufacturers that provide ''downstream'' Computer Aided Process Planning (CAPP) and Computer Aided Manufacturing (CAM) services. Nevertheless, there are still many ''classical problems'' to be overcome that have always been challenges to the CAD/CAM community. ''Over-the-wall-manufacturing'' has been a convenient label for this problem. In general, it means that designers create part-designs that contain unresolvable ambiguities at the planning and/or manufacturing stages.

No. of Setups	3
No. of Features	9
Planning Time	0.75
Setup Time	55.22
Machining Time	33.80
Material	ABS

Cost of material	$2.69
Cost of tool	$2.25
Cost of planning	$0.36
Cost of setup	$26.69
Cost of machining	$16.33
Total Cost	$48.32

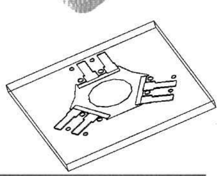


No. of Setups	3
No. of Features	9
Planning Time	0.83
Setup Time	40.60
Machining Time	13.33
Material	ABS

Cost of material	$1.34
Cost of tool	$0.88
Cost of planning	$0.40
Cost of setup	$19.65
Cost of machining	$6.44
Total Cost	$28.71

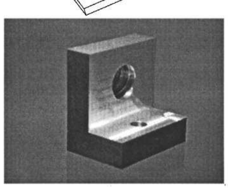
No. of Setups	6
No. of Features	12
Planning Time	0.66
Setup Time	80.0
Machining Time	43.13
Material	ABS

Cost of material	$1.54
Cost of tool	$2.87
Cost of planning	$0.32
Cost of setup	$38.72
Cost of machining	$20.94
Total Cost	$64.39



No. of Setups	1
No. of Features	17
Planning Time	2.85
Setup Time	34.17
Machining Time	20.20
Material	ABS

Cost of material	$11.45
Cost of tool	$1.34
Cost of planning	$1.38
Cost of setup	$15.52
Cost of machining	$9.76
Total Cost	$39.45



No. of Setups	3
No. of Features	17
Planning Time	1.14
Setup Time	61.48
Machining Time	5.02
Material	Al6061

Cost of material	$0.54
Cost of tool	$1.34
Cost of planning	$0.55
Cost of setup	$29.72
Cost of machining	$2.43
Total Cost	$34.48

Fig. 8 Test parts made by CyberCut pipeline. Times are in minutes and a hourly rate of $29.00 was used for cost calculation

The CyberCut project set out to create a CAD/CAM system that would make the manufacturing constraints clear to the designer so that all designs would be manufacturable. In CyberCut, mechanical designers are obliged to work with a Web-based Java CAD interface called WebCAD that explicitly contains the rules for 3-axis CNC machining using a set of prescribed cutting tools (standard mills and drills).

Another major constraint is that the WebCAD designer must use Destructive Solid Geometry (DSG), rather than unconstrained CSG. Consequently, the DSG-paradigm and machining rules vastly simplify process planning and allow for significant amount of *automated process planning*. Features do not have to be ''recognized'' by either humans or complex software because they have been predetermined by using WebCAD. Similarly, the selection of the cutting speed, feed rates, tool paths and operations, in general, can be automated.

CyberCut has been used for the design and manufacture of simple parts, that do not require special purpose fixtures nor have high tolerance or fine surface finish requirements. Figure 8 shows a range of components which were used for testing the ability of the system. These parts were designed in WebCAD, planned with CyberCut's process planners, and fabricated using a 3-axis milling machine. The planning took only minutes, but the physical setup time on the machine was often close to one hour, usually comparable with the machining time. Typical times and costs are shown in the adjoining table in Figure 8.

At the time of this writing, the price of CyberCut's automation is that it is applicable only to relatively simple 2.5D parts. As work proceeds CyberCut is being extended to more complex geometries, higher quality, and broader process capabilities. Nevertheless traditional CAD/CAM practice will continue to dominate applications where full design flexibility is desired and an organization is willing to pay (in both time and cost) for consequently greater cost in manufacturing.

Acknowledgment

Authors thank to National Science Foundation for the CyberCut grant (EIA-9617995) and to Mike Schmittdiel and James Porter for their assistance in fabrication of parts.

References

[1] Wright, P. K., 2001, *21st Century Manufacturing*, Prentice Hall, Englewood Cliffs, NJ.

[2] SDRC, 2000, http://www.SDRC.com/pub/catalog/ideas/index.html#CAM, SDRC.

[3] PTC, 2000, http://www.ptc.com/products/proe/production/index.htm, Parametric Technology Corp.

[4] Hsiao, D., 1990, A Meta Knowledge Base for Manufacturability Evaluation, Ph.D. Dissertation, Dept. of Mech. and Aerospace Eng., Arizona State University.

[5] Cutkosky, M. R., and Tenenbaum, J. M., 1992, ''Toward a Framework for Concurrent Design,'' Int. J. Syst. Autom.: Res. Appl., **1**, No. 3, pp. 239–261.

[6] Gupta, S. K., Regli, W. C., and Nau, D. S., 1994, ''Integrating DFM With CAD Through Design Critiquing,'' Concurrent Eng. Res. Appl., **2**, No. 2, pp. 85–95.

[7] Das, D., Gupta, S. K., and Nau, D. S., 1995, ''Estimation of Setup Time for Machined Parts: Accounting for Work-holding Constraints Using a Vise,'' Computers in Engineering—and Proceedings of the 1995 Database Symposium.

[8] Feng, C.-X., and Kusiak, A., 1995, ''Constraint Based Design of Parts,'' Comput.-Aided Des., **27**, No. 5, pp. 343–353.

[9] Dunn, M., ed., 1992, *Industrial Automation Systems and Integration-Product Data Representation and Exchange-Part 48: Integrated Generic Resources: Form Features*, second edition, ISO/WD 10303-48.

[10] Slovensky, L., ed., 1994, *Industrial Automation Systems and Integration-Product Data Representation and Exchange-Part 224: Application Protocols: Mechanical Product Definition for Process Planning Using Form Features*, working draft, ISO TC184/SC4/WG3_N333.

[11] Shah, J. J., and Mäntylä, M., 1995, *Parametric and Feature-Based CAD/CAM-Concepts, Techniques and Applications*, A Wiley-Interscience Publication, Wiley, New York.

[12] van Houten, F. J. A. M., and van 't Erve, A. H., 1992, ''Feature Based Manufacturing With PART. Proceedings of the International Conference on Manufacturing Automation,'' Proceedings of the International Conference on Manu-

facturing Automation, Hong Kong, 10–12 Aug., Hong Kong: Univ. Hong Kong, pp. 464–475.
[13] Boerma, J. R., and Kals, H. J. J., 1988, ''FIXES, a System for Automatic Selection of Setups and Design of Fixtures,'' Ann. CIRP, **37**, No. 1, pp. 443–446.
[14] Hayes, C. C., Gaines, D. M., Faheem, W., and Castano, J. F., 1997, ''MAPP: A Matrix Architecture for Process Planning,'' IEEE International Symposium on Assembly and Task Planning (ISATP), August.
[15] Gaines, D. M., Castano, F., and Hayes, C. C., 1999, ''MEDIATOR: A Resource Adaptive Feature Recognizer that Intertwines Feature Extraction and Manufacturing Analysis,'' ASME J. Mech. Des., **121**, pp. 145–158.
[16] Shirur, A., Shah, J. J., and Hirode, K., 1998, ''Machining Algebra for Mapping Volumes to Machining Operations for Developing Extensible Generative CAPP,'' Journal of Manufacturing Systems, Society of Manufacturing Engineers, **17**, No. 3, pp. 167–182.
[17] Hirode, K., and Shah, J. J., 1999, ''Metrics for Evaluating Machining Process Plans,'' Proceedings of the 1999 Design Engineering Technical Conference, September 12–15, Las Vegas, Nevada.
[18] Sakurai, H., 1992, ''Automatic Setup Planning and Fixture Design for Machining,'' J. Manuf. Syst., **11**, No. 1, pp. 30–37.
[19] Joneja, A., and Tien-Chien Chang, 1999, ''Setup and Fixture Planning in Automated Process Planning Systems,'' IIE Transactions, **31**, No. 7, Kluwer Academic, Durdrecht, July, pp. 653–665.
[20] Gupta, S. K., and Nau, D. S., 1995, ''A Systematic Approach for Analyzing the Manufacturability of Machined Parts,'' Comput.-Aided Des., **27**, No. 5, pp. 323–342.
[21] Sarma, S. E., 1995, *A Methodology for Integrating CAD and CAM in Milling*, Ph.D. thesis in Mechanical Engineering, University of California, Berkeley.
[22] Kim, J. H., Wang, F. C., Sequin, C. H., and Wright, P. K., 1999, ''Design for Machining Over Internet,'' Design Engineering Technical Conference (DETC) on Computer Integrated Engineering Paper Number DETC'99/CIE-9082, Las Vegas, NV, September.
[23] Mukherjee, A., and Liu, C. R., 1997, ''Conceptual Design, Manufacturability Evaluation and Preliminary Process Planning Using Function-Form Relationships in Stamped Metal Parts,'' Robot. Comput. Manufact., **13**, No. 3, pp. 253–270.
[24] Chen, Y., and Wei, C., 1997, ''Computer-Aided Feature-Based Design for Net Shape Manufacturing,'' Rob. Comput.-Integr. Manufact., **10**, No. 2, pp. 147–164.
[25] McMains, S., Sequin, C. H., and Smith, J., 1997, ''Dealing with Data Imprecision, Design Tolerances, and Manufacturing Limitations in SIF,'' Measurements and Standards Issues in Rapid Prototyping, October 16–17, National Institute of Standards and Technology, Gaithersburg, MD.
[26] Ostwald, P. F., 1992, *Engineering Cost Estimating* (3rd ed.), Prentice Hall, Englewood Cliffs, NJ.
[27] Sarma, S. E., and Wright, P. K., 1997, ''Reference Free Part Encapsulation: A New Universal Fixturing Concept,'' J. Manuf. Syst., **16**, No. 1, pp 35–47.
[28] Ahn, S. H., Plancarte, J., and Wright, P. K., 2000, ''The Impact of Reference Free Part Encapsulation (RFPE) on Design for Manufacturability with CNC Machining,'' ASME International Design Technical Engineering Conference, Baltimore, Maryland, September 10–13.
[29] Ahn, S. H., and Wright, P. K., 2000, ''Reference Free Part Encapsulation (RFPE): An Investigation of Material Properties and the Role of RFPE in a Taxonomy of Fixturing Systems,'' submitted to J. Manuf. Syst.
[30] Yang, Q., Nau, D. S., and Hendler, J., 1992, ''Merging Separately Generated Plans with Restricted Interactions,'' Comput. Intelligence, **8**, No. 4, pp. 648–676.
[31] D'Souza, R., Wright, P. K., and Sequin, C., 2000, ''Microplanning for 2.5 D Pocket Machining,'' submitted to J. Manuf. Syst.
[32] Held, M., 1991, *On the Computational Geometry of Pocket Machining*, Lecture Notes in Computer Science 500, Springer-Verlag, Berlin.
[33] Held, M., 1994, ''Pocket Machining based on Contour-parallel Toolpaths Generated by means of Proximity Maps,'' Comput.-Aided Des., **26**, No. 2, pp. 189–203.