Feature-Based Process Planning in the AMRF

Mark B. Unger
Steven R. Ray
National Bureau of Standards

ABSTRACT

The growing use of "features" in the design and manufacturing communities is due to the fact that features better capture the functionality or intent associated with a part design. The process planning team at the National Bureau of Standards is using features and their associated functionality to derive process plans from a part model in a semi-automatic fashion. The process planning team believes that features are the key to having a fully automated planning system.

This paper will discuss the feature-based process planning work currently in progress at the AMRF. Before the current work is described, background information will be given on the AMRF, process planning in the AMRF, some existing definitions of a feature, how the process planning team defines a feature, and current research in the field of features.

I. INTRODUCTION

The use of "features" is a recent approach to the old problem of attempting to link CAD with CAM. The features approach constrains the designer/process planner to working with a set of features which have significance for either design, analysis or manufacturing. Instead of using a model consisting of graphics primitives (e.g., lines, circles, points), the designer/planner is asked to use a set of features (e.g., holes, pockets, slots) from which manufacturing operations can be derived. When both the designer and process planner have finished the design and process plan, more information has been entered into the model than if the part had been created on a traditional CAD system. However, the designer and planner have entered the information not at the geometric level but at a higher level.

The process planning team of the Automated Manufacturing Research Facility (AMRF) has proposed and is currently building a process planning system which is based on the use of features. The new system will capture more manufacturing knowledge from both the designer and process planner. The ultimate goal of the process planning team is to completely automate the planning for manufacture of a part. The current goal of the project is to produce a process plan from a design representation containing geometry, topology, datum, tolerance, and features information. The process plan will contain references to: manufacturing features, setup information, tool selections, and sequencing of steps for the production of the part.

This paper will discuss the feature-based process planning work currently in progress at the AMRF. Before the current work can be described, background information will be given on the AMRF, process planning in the AMRF, some existing definitions of a feature, how the process planning team defines a feature, and current research in the field of features.

II. AUTOMATED MANUFACTURING RESEARCH FACILITY

The AMRF [15] was established in 1981 to serve as a testbed facility to support research in measurement techniques and interface standards required for automated machining of parts in small lot sizes. One of the primary thrusts of the project was to establish clear interface specifications and modular structures to allow plug-compatibility between systems. Plug-compatibility provides the potential for a flexible manufacturing environment and offers the capability of incremental automation in existing facilities. Results of this work are already contributing to the formulation of standards for a generic factory model, low level robot interfaces, process plan file structures, N/C machine tool interfaces, communication standards, graphics standards in the form of IGES (Initial Graphical Exchange Specification), and product model standards in the form of PDES (Product Data Exchange Specification). One of the concepts being developed in the generic factory model and central to the design of the AMRF architecture is the concept of hierarchical control [10,15].

AMRF Architecture.

The AMRF is built around the concept of hierarchical control, where high level commands are decomposed into sequences of simpler commands at the next lower level in the hierarchy. These simple commands are in turn decomposed to yet lower level commands (Figure 1). Well-defined protocols have been established to allow command and status information to flow up and down the control hierarchy. By adopting a hierarchical control approach, the complexity of a task is reduced to a manageable amount for every level in the hierarchy. The majority of data transfer (such as process plans and part models) does not flow within the control hierarchy. This information is passed from a separate data administration system directly to any level of control. The separation of data and control avoids many potential information bottlenecks. Process planning is responsible for producing much of the data passed within the AMRF.

III. PROCESS PLANNING IN THE AMRF

A major goal of the process planning team was to address the interface and representation issues of planning systems. This included both a neutral exchange specification for a process plan at any level in the control hierarchy, and a general internal representation of a plan. The exchange format had to be simple enough to be easily parsed by the least capable computers in the facility, yet flexible enough to convey complex process plans containing parallel (i.e., simultaneous) activities. This format has been defined and is being used by computer controllers at the bottom three levels in the AMRF hierarchy[10].

AMRF process plans are broken into a hierarchy, just as the AMRF control is. Currently, process plans are used at the cell, workstation and equipment levels. Each process plan contains a number of work elements for a controller to execute. A work element is defined as a procedure or step in a process plan. The work elements are arranged in a precedence graph. An example of an equipment level work element for a milling machine might be endmilling. An example of a cell level work element might be machine-lot. The use of geometry, topology, and features is particularly important in equipment level process plans.

Currently, all of the geometry, topology, and feature information describing the desired part is stored in a part model file. All of the procedural information to produce the desired part is stored in a process planning file. The link between the part model and the process plan is features. Before the discussion of the use of features and their importance can continue, a feature must be defined, and similar research efforts examined.

IV. FEATURES

Each researcher working in the features arena has his own definition of a feature, and the definitions differ. The definitions of a feature as given by some of the major researchers are presented here. The content and focus of each researcher’s work will be discussed later.

Dixon[3]:

"any geometric form or entity whose presence or dimensions in a domain are germane to manufacturing evaluation or planning, or to automation of functional analyses."

Henderson[4]:

"features may be defined as a part ’characteristic,’ including material type, functionality and other pieces of descriptive information."

Hirschtick[7]

"A feature of a geometric model in a given context is a descriptor of that model whose presence and/or size is relevant within the given context."

Hummel[8]:

"features are regions of the part that have some degree of manufacturing significance. Put another way, features form reoccurring geometric and technologic patterns for which the process engineer has acquired years of manufacturing experience."

Luby[12]:

"a geometric form or entity, whose presence or dimensions are required to perform at least one CIM function (i.e. graphics, analysis, process planning), and whose availability as a primitive permits the design process to occur."

Vaghul[16]:

"A feature is a geometric entity: 1) whose presence, location, or dimensions are germane to the functionality or manufacturability of the part; or 2) whose availability as a primitive facilitates design in the domain."

All of the definitions of features share the idea of a "geometric entity." All of the definitions also assume that features are combined to form parts or other objects of importance. All of the definitions also imply that features provide a higher level model of the object than does a traditional CAD geometric model, and thus may be easier to reason about or use.

The definitions of features differ because each researcher considers different ideas significant. Some researchers are interested is *designing* with features, some are interested in *manufacturing* related features. The definition of the feature is dependent upon its use. This is the unfortunate aspect of current feature definitions. A features representation is not unique. A designer and a manufacturer working with a feature model must agree on the same feature definitions and interpretations. If one considers current CAD models, a large amount of the information on the drawings is not geometric and requires consistent interpretations between manufacturing and design. Interpretation is required because some manufacturing attributes cannot be passed geometrically. The AMRF process planning team is studying how features should be defined so that both design and manufacturing can easily work with the model.

The AMRF process planning team proposes the following definition:

> "A feature is a higher level grouping of geometrical, topological and functional primitives into an entity more suitable for use in design, analysis, or manufacture."

The AMRF process planning team's definition varies from the other definitions in the fact that the feature is just a structure which logically groups the manufacturing knowledge associated with the part. Just as topology lies on top of geometry to provide form and organization, features are structures that lie on top of a part representation providing form and organization.

The previous comparison of feature definitions does not constitute a review of research efforts; a more thorough, but hardly complete discussion of each researcher's work follows in the next section.

V. CURRENT RESEARCH

A brief description of current research on features follows. The descriptions are broken down by individual researchers. "Current" is a relative word, thus dates of papers presented have been given to aid in the discussion. The descriptions which follow in this section are of researchers who are working in several areas dealing with features. Dixon and Hummel present information and ideas about features in general. The discussion of Dixon is based on a paper where he describes fundamentally why features are used, and does not discuss the specific systems he has developed. Hummel has demonstrated the powerful advantages of using features in a factory environment. Hummel does not however have a design by features system; features are added manually to an existing CAD model. Henderson and Hirschtick have developed feature extraction systems. The remainder of the researchers discussed in this section are working on feature-based design systems.

Dixon[2] 1986

John Dixon (University of Massachusetts) has worked on several feature-based design systems. The work of two of his students is also presented in this paper, (Luby[12] and Vaghul[16]). Dixon's work has been in the area of applying artificial intelligence (AI) to mechanical engineering (ME) design. Dixon notes that many successful electrical engineering (EE) design systems applying AI techniques exist, but that very few AI systems have been successfully applied in the mechanical domain. Dixon feels that there are fundamental differences between EE design and ME design that account for this discrepancy. The fundamental differences he lists are "materials, manufacturing, and geometry." Dixon views features as a way to represent 3-D design and manufacturing geometries. He also feels that features are a good approach for representing design knowledge in AI systems. He has worked on several feature-based expert systems in the areas of extrusions, injection moldings, and castings, and he hopes to reduce the discrepancies between ME and EE design system's capabilities.

Hummel[8] 1986

Keith Hummel (Bendix, Kansas City) has worked in the area of feature representation for process planning use. He has developed an expert system, XCUT, which develops a process plan for the manufacture of parts. XCUT uses a feature-based model of the part to develop the process plan. The XCUT system relies on the user to identify features manually from an existing CAD representation. Once the feature model of the part has been developed, a process plan can be automatically generated using a production rule system. Hummel uses symbolic and object-oriented data structures to represent features. Hummel uses these data structures because they have the capability to represent complex objects, and they have the flexibility to be updated as new techniques and processes are introduced. Hummel feels that "the vast majority of process planning is of a symbolic nature," thus the need for a symbolic data structure. Another distinct advantage of object-oriented techniques which Hummel uses extensively is the concept of inheritance. New types of objects can be created by combining other subtypes of objects. The new objects then inherit the attributes, and properties of the subtypes. Hummel uses inheritance to develop what he calls a "feature taxonomy." The feature taxonomy is an inheritance structure for features. At the top of the feature taxonomy tree are the most general structures, and at the bottom are the most specific structures. The specific structures inherit the properties of the more general structures.

Kramer[11] 1987

Tom Kramer (Catholic University) has developed a design protocol named VWS2 used at the vertical workstation in the Automated Manufacturing Research Facility (AMRF) of the National Bureau of Standards. VWS2 is a feature-based design tool which can be used to design parts for several part families from a set of features produced by a vertical machine tool. VWS2 allows both graphical and textual input of feature information, and then produces several outputs. The primary output is NC code which can be downloaded to a machine tool to manufacture the part. VWS2 has several other useful outputs. The part design can be displayed on a monitor, and a simulation of the NC code run to show the order and path that the cutter will take to produce each feature. VWS2 can also generate a process plan which can be executed in the vertical workstation. Kramer uses the language Franz Lisp, and a property list as the data structure to represent features. A property list is a collection of a number of attribute-value pairs. Each attribute-value pair of a feature must be completely defined to complete the design. Once the design is completed, the attribute-value pairs are then checked using a number of verification rules; the rules check for impossible situations. Kramer has developed a set of features which a vertical milling machine is capable of producing. Some examples of his features are: grooves, holes, pockets, text, and side-contours. There are also a number of subfeatures which can be attached to any of the primary features. Examples of subfeatures are: chamfers, countersinks, and threads.

Luby[12] 1986

Steven Luby (University of Massachusetts) has worked in conjunction with Dixon in the area of a "features-based design aid." The purpose of a design-aid is to provide on-line suggestions during the design process. Luby developed a system named Casper. Casper helps an engineer design castings. Casper is concerned primarily with manufacturability considerations, such as metal flow and feeding, and with minimizing the stresses generated during cooling of a casting. Casper will provide suggestions during design to make the casting design easier to manufacture. The types of features which Casper allows are: slabs, corners, L-brackets, bosses, holes, and ribs. After each feature is added to the part, an evaluation is made of the part to ensure geometric consistency. Luby has also chosen Lisp as the implementation language for Casper, and has used an object oriented programming approach. He has done this by implementing an "object hierarchy," ranging from most general to most specific.

Vaghul[16] 1985

Vaghul (University of Massachusetts) has built a system called IMPARD. IMPARD stands for Injection Molding Part Design. IMPARD is also a design aid system which evaluates parts during the design process for manufacturability problems. IMPARD is similar to Casper in that features are added, and design considerations are returned to the user. There are differences besides just the type of manufacturing process considered. IMPARD uses a true solid modeler to represent the molding. Unlike Casper which ran a geometric compatibility check to ensure that it could handle the intersection of the new feature with the existing part, IMPARD can generate the actual geometric model that would be produced. It is unclear however if IMPARD’s decisions will still be valid for complex intersections. Vaghul comments that as the number of features becomes larger, designing and evaluating the design will become harder.

Henderson[5] 1984

Mark Henderson (Arizona State) has taken the production rule approach to the problem of deriving features from a CAD model. Production rule systems examine the part model using rules to determine if a feature has been found. Rules are conditional statements which are applied to the part; if all of the conditions are met, then an appropriate action is taken. Henderson uses another interesting approach to the problem of feature extraction in which he does not look at the part definition itself but looks at "cavity volumes." Cavity volumes are found by subtracting the part definition from the raw stock definition. This operation leaves the volumes of material which must be removed by the manufacturing operation. These cavity volumes are then examined for features. When a feature is found, it is subtracted from the cavity volumes using solid modeling techniques, which results in new, smaller cavity volumes. This process is repeated until all cavity volumes are completely eliminated. The features which have been found are formed into a feature tree, and the part has been redefined in terms of features. Henderson limited his search for features to swept features, which he defines as "2 1/2 D volumes created by sweeping a tool profile through space." He limited his search because these are the most common types found in machining. Examples of some of the types of features he can identify are holes, slots, and pockets. Henderson feels that rules are useful for identifying the "general" feature, (e.g., hole, slot, or pocket), but that other techniques may be more suitable for identifying the "specific" feature, (e.g. straight slot versus curved slot). Henderson used the language Prolog because it is a logic language which is suitable for a production rule system. Henderson found conditions and actions of a rule to be easily represented in Prolog.

Hirschtick[7] 1986

Hirschtick (Massachusetts Institute of Technology) has developed a system called "The Extrusion Advisor." It is a system designed to advise a user in the domain of aluminum extrusions. The system examines a geometric model of a part to be manufactured by an extrusion process, it extracts features, and then identifies features which will cause manufacturing difficulties. Hirschtick uses "characteristic patterns" to find an instance of a feature. The geometric definition of the part is searched for one or more characteristic patterns, several of these patterns are combined to form a characteristic pattern set or feature. Examples of characteristic patterns are parallel lines, triangles, etc.... Hirschtick uses a production rule system to examine the part geometry for characteristic patterns. Examples of the kinds of features Hirschtick can recognize are: walls, hollows, knife-edges and a number of more specific kinds of the last three. The output of the Extrusion Advisor is a list of features which will present manufacturing difficulties, the reason why they cause a problem, and a possible remedy. Hirschtick concludes that feature-based design and feature extraction will both be necessary. Hirschtick calls feature-based design, "feature descriptor modeling." He notes that: "Feature descriptor modeling will provide an excellent interface for solid modeling. The resulting feature information will eliminate the need for some feature extraction. But for many powerful applications, it will be necessary to use a feature extraction technique. This is because features will be needed which are of a higher level than those used to create the part."

The AMRF process planning team agrees with Hirschtick’s conclusion that both feature extraction and design by features approaches should be used to approach the design/planning problem. We also feel that the use of object oriented and symbolic techniques is necessary to model features. A number of researchers have represented features as parametric entities which in and of themselves define the geometry and topology. We feel that a feature representation must not define the geometry and topology, but must point to valid topological and geometrical entities in a part model. For instance, some models would reference a pocket with only height, width, and translational attributes; the process planning model represents a pocket as a collection of faces. The definitions of the faces can be examined to determine the underlying surfaces and edges which form the pocket feature. For this reason, we have found it necessary to use a complete part model.

The model currently being used is the AMRF Part Model [6]. The AMRF Part Model is a PDES-like [9] format used to communicate the part geometry, topology and functionality. As the formal definition of PDES is developed, we intend to maintain compatibility. The format specifies what the major sections of the part model are, their order, and how individual entities are defined and appear in the model file. The AMRF part model supports a boundary representation solid model and has the capability to represent geometry, topology, datums, features, tolerances, and header information. The part model also allows tolerances to be associated with features forming a powerful combination; in the current planning system, tolerances influence process selection as significantly as the type of feature. The format is used both by the process planning team and the geometry modeling team and is formally described in an NBS internal report [6]. The process planning team currently supports the following simple features: pockets, slots, holes, counter-bored holes, countersunk holes, threaded holes, cutouts, and profiles. The features supported have not been taken from any of the emerging standard feature dictionaries, but are the basic features that appear in some form in all of the dictionaries and in other systems in the AMRF [11].

VI. IMPLEMENTATION

The goal of the work described in the following sections of this paper is to more tightly couple design and manufacturing by means of feature-based process planning. We feel that features provide the means to communicate the information typically missing from current CAD models. Traditionally, much information is left unspecified at the design stage and must be inferred at the process planning stage. This approach has worked due to the fact that human process planners can infer the intent of the designer. As this process becomes automated, the missing information must be made explicit. Features provide one potential answer to this problem. We are building a prototype system which directly couples the graphical manipulation capabilities of a CAD system to the inferencing capabilities of an automated planning system by the use of features. This work builds on the existing planning system in use within the AMRF [1]. This is the first step toward automatically providing manufacturability feedback to the designer.

This section describes a new interface being developed specifically to create and display feature-based models, and will constitute the graphical half of the design/planning system. This tool is tentatively called DEFEATOR, a DEsign by FEATures editOR. The current system allows display of a part model, the addition of features to the part model by manual feature extraction, or the creation of a new part model strictly by the use of features.

The current system runs on a Silicon Graphics* workstation. The Silicon Graphics workstation is a single user, UNIX based workstation which is especially designed for high speed color graphics applications. The Silicon Graphics comes with a geometry engine which allows rapid display, real time shading, and real time rotations. The workstation allows the user to access low level graphics calls directly from C. The process planning team felt that the flexibility allowed by an interactive, symbolic language was crucial to reaching our goals. Franz Lisp was chosen as the language in which to work. Franz Lisp is interactive, symbolic, and also comes with an object oriented package called Flavors. The C graphics calls were linked into the Lisp environment making for a very powerful interactive graphics environment. Some of the other interfaces developed by the process planning team have been developed in Lisp using Flavors. The Flavor definitions used on the Symbolics Lisp Machine to represent the Part Model were ported to the Silicon Graphics and used in the Franz Lisp graphics environment with few modifications.

DEFEATOR can currently load a part model in from a file, display the part model, and highlight entities of interest such as features, faces, loops, edges, vertices and points. The ability to pick and highlight features has been used by another researcher, George Vanecek, to build a package of routines which allow a user to manually select faces which form features such as holes and pockets. In this case, the user loads a part model generated previously by another solid modeler and then selects and groups the faces to form a feature. The user is also allowed to specify tolerances associated with the feature. The added feature information is then written back out in AMRF part model format. The routines developed by Vanecek are a form of manual feature extraction.

DEFEATOR can also be used as a feature-based design system. In this case, DEFEATOR starts with a clean slate and allows a user to build a part model from scratch. The user is allowed to design new parts from a set of manufacturing features. The user starts with a solid prismatic block feature, or a profile feature which is a solid composed of any number of connected straight line segments and circular arc segments "lifted" into a solid. The user is then given a number of subtractive features such as holes, pockets and slots which can be removed from the solid. The entire geometry, topology, features, and tolerances sections are completed and then a file in AMRF part model format is written.

A screen dump showing the DEFEATOR interface with a part model displayed can be seen in Figure 2.0. The DEFEATOR screen shows some of the capabilities of the system. Four subwindows are displayed; the subwindows are: 1) graphics, 2) help, 3) controller bar, and 4) command. The graphics subwindow is used for the graphical display of the part model. A fairly complex part is shown, this model was produced using another solid modeler. The help window displays textual messages to the user during an editing session. The first message displayed to the user is shown in the figure. Below the help window is a control-bar window. The control-bar window allows entering of numerical values using the mouse alone. If DEFEATOR needs a numerical value for an attribute, the attribute is placed on a controller-bar. The controller-bar is mouse sensitive, and each time the mouse is moved over the bar, a numerical value is output. Rarely does the user have to go to the keyboard, although this is an option. Below the controller-bar window is the command window. The command window is where all commands are entered. Each rectangle in the command window is called a "button." Each button is mouse sensitive, and when activated causes a command to be executed. The buttons contain the top-level commands, and will be briefly described.

Selected commands will be described by column. The first column contains the highest-priority system commands. "Features" is the top-level feature-based-design command. The features command displays a menu of the design features which can be added to the model. The second column contains the part-model access commands. The load, parse and write buttons allow the part-model to be loaded or written. The third column contains display commands. Draw displays a part-model which has been loaded. Rotate allows the part to be rotated in real time. Pick allows the user to identify subentities such as faces and edges using the mouse. After being picked, the entity is returned and highlighted. The fourth column contains feature manipulation commands. These commands can be called after a feature has been added. They allow a feature to be copied, modified or deleted from the part model. The fifth column contains the display option commands. If the display is not centered or scaled correctly, pan and zoom will solve the problem. Shade is another display command that is under development. It will be discussed in the planned work section. The last column contains the commands developed by George Vanecek mentioned previously. Label is the command which allows the user to manually identify features from an existing model. This is a useful function when a features representation is desired on a model created by another system.

There are two other command windows which are not shown. One contains commands for fixturing a part. The second contains commands for process planning for a part. Neither the fixturing nor planning commands have been extensively developed, but will be developed in future efforts.

VII. PLANNED WORK

Current plans are to use the system described above to create an intelligent, interactive environment which allows a process engineer to manipulate a part model while planning its manufacturing sequence. Initially, this will be accomplished by coupling the Silicon Graphics system with a Symbolics Lisp machine. Much of the low-level communications work has been accomplished. Using this dual system, the Silicon Graphics workstation can serve as an intelligent graphics console for the Lisp machine. During a typical planning session, a process engineer will bring up a previously created part model for display on the graphics screen. He can use DEFEATOR to identify and update all the relevant feature information. Then, by pointing at each of the features in turn, the engineer can specify the manufacturing sequence. DEFEATOR will automatically communicate the engineer’s actions to the Lisp Machine, which will then build the process plan data structures. At the same time, expert systems running on the Lisp machine, such as the Semi Intelligent Process Selector (SIPS) [13], will provide manufacturing feedback such as the suggested processes to use, their cost, tooling, and machine selection. Individual feature planning will be integrated into a hierarchy of distributed planning modules [2]. Higher levels of planning will incorporate optimization, minimizing tool changes, minimizing setup time, and scheduling.

A major focus of the work underway is the complete integration of data preparation supporting manufacturing. This includes: 1) design and process planning at all levels in a hierarchy, 2) equipment programming, and 3) N/C coding. In this context, feature-based planning cannot be considered in isolation. Rather the needs of each of the data preparation stages must be met in the definition of features, and their uses. Our efforts are primarily to identify or define reasonable interfaces and data structures to allow this integration to take place throughout the research community and ultimately in a production environment. Examples include not only part model and process plan exchange formats, but also more detailed interfaces linking sub-modules within a process planning system.

VIII. CONCLUSIONS

The use of features is a promising approach to the area of design and process planning. It allows information to be encoded symbolically for easy use by planning and analysis systems. The reason a features representation has not been widely used in the past is that existing computers did not have the power to manipulate objects symbolically. New languages and computers have been developed which have the ability to manipulate these symbolic representations. These advances have made the use of features more realizable.

Features better capture the functionality associated with a part by overlaying a data structure which captures this functionality. This extra encoded information allows process plans to be produced, allows other systems to review the design, and allows manufacturing operations to be produced automatically from the design. The work underway at the National Bureau of Standards is attempting to develop the concepts of feature-based planning in this wider context of a fully integrated manufacturing facility.

The NBS Automated Manufacturing Research Facility is partially supported by the Navy Manufacturing Technology Program.

** Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.*

REFERENCES

[1] Brown, P.F., McLean, C.R., "Interactive Process Planning in the AMRF", Bound Volume of the *Proceedings of the Winter Annual Meeting of the ASME*, Anaheim, CA, December 1986.

[2] Brown, P.F., Ray, S.R., "Research Issues in Process Planning at the National Bureau of Standards", *Proceedings of the 19th CIRP International Seminar on Manufacturing Systems*, Pennsylvania State University, June 1987, pp. 111-119.

[3] Dixon, J.R., "Artificial Intelligence and Design: A Mechanical Engineering View", *Proceedings of AAAI-86 Fifth National Conference on Artificial Intelligence*, Vol. 2, Philadelphia, PA, August 1986, pp. 872-877.

[4] Henderson, M.R., "Automated Group Technology Part Coding from a Three-Dimensional CAD Database", *Proceedings of the Winter Annual Meeting of the ASME,* Anaheim, CA, December 1986, pp. 195-204.

[5] Henderson, M.R., "Extraction of Feature Information from Three Dimensional CAD Data", *Ph.D. Thesis*, Purdue University, May 1984.

[6] Hopp, T., "AMRF Database Report Format: Part Model", *NBS Internal Report NBSIR 87-3672*, (1987).

[7] Hirschtick, J. K., Gossard, D. C.,’’Geometric Reasoning for Design Advisory Systems’’, *Proceedings of the 1986 Computers in Engineering Conference,* Chicago, IL., July, 1986, pp. 263-270.

[8] Hummel, K. and Brooks, S., "Symbolic Representation of Manufacturing Features for an Automated Process Planning System", *Proceedings of Winter Annual Meeting of the ASME*, Anaheim, CA, December 1986, pp. 233-243.

[9] Jud, J.L., "Designing with Features - The New Generation of Product Modelers", *AUTOFACT 87*, Detroit, Michigan, Novemeber 1987.

[10] McLean, C., "An Architecture for Intelligent Manufacturing Control", *Proceedings of Summer 1985 ASME Conference*, Boston, Massachusetts, August 1985.

[11] Kramer, T., Jau-Shi J., ’’The Design Protocol, Part Design Editor, and Geometry Library of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards’’, NBSIR 88-3717, 1988.

[12] Luby, S. C., Dixon, J. R., Simmons, M. K.,’’Creating and Using a Features Data Base’’, *Computers in Mechanical Engineering*, Vol. 5, no. 3, November 1986, pp. 25-33.

[13] Nau, D.S. "Hierarchical Abstraction for Process Planning", to appear in *Second Intl. Conference on Applications of Artificial Intelligence in Engineering*, Boston, MA, 1987.

[14] Ray, S.R. and McLean, C., "AMRF Process Plan Format", *NBS Internal Report*, (in preparation), (1988).

[15] Simpson, J.A., Hocken, R.J. and Albus, J.S., "The Automated Manufacturing Research Facility of the National Bureau of Standards", *Journal of Manufacturing Systems*, Vol. 1, No. 1, 1982, pp. 17-32.

[16] Vaghul, M., Dixon, J. R., Sinsmeister, G. E.,’’Expert Systems in a CAD Environment: Injection Molding Part Design as an Example’’, *Proceedings of the 1985 ASME Computers in Engineering Conference*, Boston, MA, Aug, 1985.

Figure 1.0 Example of Command Decomposition using Hierarchical Process Plans.
(not available in electronic version)

Figure 2.0 Screendump of DEFEATOR
(not available in electronic version)