Numerical Control Tool Path Generation In A Solid Modeler: The Framework For Generative Numerical Control.

Citation

Aber, David Michael. *Numerical Control Tool Path Generation In A Solid Modeler: The Framework For Generative Numerical Control.* 1995, https://preserve.lehigh.edu/lehigh-scholarship/graduate-publications-theses-dissertations/theses-dissertations/numerical-48.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road. Ann Arbor. MI 48106-1346 USA
313/761-4700 800/521-0600

Numerical Control Tool Path Generation in a Solid Modeler - The Framework for Generative Numerical Control

by

David Michael Aber

Presented to the Graduate and Research Committee

of Lehigh University

In Candidacy for the Degree of

Doctor of Philosophy

in

Industrial Engineering

Lehigh University
1995

UMI Number: 9611111

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

11-7-95

(Date)

Emory W. Zimmers

Dissertation Director

12-7-95

(Accepted Date)

Committee Members:

Emory W. Zimmers

Dr. Emory W. Zimmers, Jr.
Professor of Industrial Engineering

Mikell P. Groover

Dr. Mikell P. Groover
Professor of Industrial Engineering

Dr. G. Sathyanarayanan
Professor of Industrial Engineering

Tulga Ozsoy

Dr. Tulga Ozsoy
Associate Professor of
Mechanical Engineering

Acknowledgements

I would like to express my gratitude nearing the end of a path that became my way of life for an extended period. The opportunities given and experiences gained have opened many horizons at both the technical and personal level. At this time, I would like to acknowledge several more significant contributions. The perseverance of my family, which first hand have experienced the true cost of this research and the opportunity for its development in industry. My advisor, Dr. Emory Zimmers, whose dedication to the engineering environment at Lehigh promotes the synergy that can be realized through a close coupling of university and industry in both the lecture hall and laboratory. My doctoral committee, Dr. Mikell Groover, Dr. G. Sathyanarayanan, and Dr. Tulga Ozsoy, whose involvement with this dissertation have provided a broad base of engineering expertise and maintained high scholarship. Their involvement required their time investment and perseverance to close on the recording of this contribution to numerical control programming. The confidence of my colleague Francis J. Hildenbrand, whose project management competence enabled the opportunity for industrial development of this research. The completion of this dissertation is made through the continued support and encouragement of this friend. The faith, desire, and effort of my industry colleagues. These folks accepted my lead in the development of a world class part programming system. This accomplishment was made during the darkest years in our industry's history. The colleagues making outstanding implementation contributions were Ken Reester (manufacturing engineering), Michael Mercincavage (manufacturing engineering and change control), David Starke (knowledge-based programming), Jim Thompson (engineering), David Knapp (programming methods and standards), David Klimchak (programming), Robert Pistey (documentation and software librarian), A. Ismail (programming), Pam Kott (engineering and documentation), Stefani Brill (programming), Sharon Tocker-Lefloch (engineering and documentation), Eunjin Han (programming), and Jim Brown (management).

Contents

Figures

Tables

Abstract

This study examines the problem of numerical control part programming for industrial production machining. Part program preparation is undertaken using the environment provided by a production level solid modeler with associated CAD/CAM services. The approach taken in this research is to begin with the practical knowledge of numerical control machining with concentration in the part programming process. This knowledge is extended to include information on machines, cutting tools, setup requirements, operation parameter selection, and operation planning. Theory is introduced to the study through the literature. The practical knowledge and theory of numerical control provided by other research in the field is then furthered. The objective is to develop new concepts and the programming framework for next generation industrial production numerical control machining. The business environment of the 1990's demands this industrial production and pragmatic link for research in this field.

The results of the study are (1) A conceptual definition of the system components for generative numerical control part programming. (2) An architecture that reduces-to-practice the goals of automation in this field. This architecture provides an evolutionary path for continued functional growth and enhancement. It also incorporates solution scaling to the varied industry numerical control part programming problems and machine shop configurations. (3) Specification of a set of numerical control functions that defines the breadth and depth of this generative architecture.

In summary, a framework for next generation numerical control part programming is proposed by this research.

1.0 Introduction

The creation of computer numerical control toolpaths is a fundamental step in the use of manufacturing numerical control machining applications. These applications generally conform to a basic numerical control (NC) process.

The NC process is shown in Figure 1 on page 3. This structure has been extracted from the literature generalized by the author, and adapted to current practice through the numerical control procedure as defined by Groover and Zimmers [22]. This figure displays tasks and sequence in the transformation of an engineer's design specification to the finished part. This process is a set of steps required to manufacture a part using a numerical control machine tool.

Koloc [1] contends that the application of the NC process is dependent on "the availability of consistently good programs required by production." He also notes that "any NC machine tool operates only as faultlessly and efficiently as the pertinent control program directs it to do." Koloc summarizes his concerns with respect to toolpath creation by noting that "...the quality of the (numerical) control program is one of the key factors determining the profitability of the numerical control machine tool operation."

The need for effective toolpath creation was defined very early in a treatment of numerical control by Wilson [2]. He states "for maximum efficiency, it is imperative that the control tape (program) be correct as it is received on the production floor."

---

[1] Koloc, J., "The Influence of the Programming Language on the Productivity and Reliability of Part Programming," Proceedings of the 13th Annual Meeting of Numerical Control Society, 1976.

[2] Wilson, F., Editor, Numerical Control in Manufacturing, New York: ASTME, McGraw-Hill, 1963.

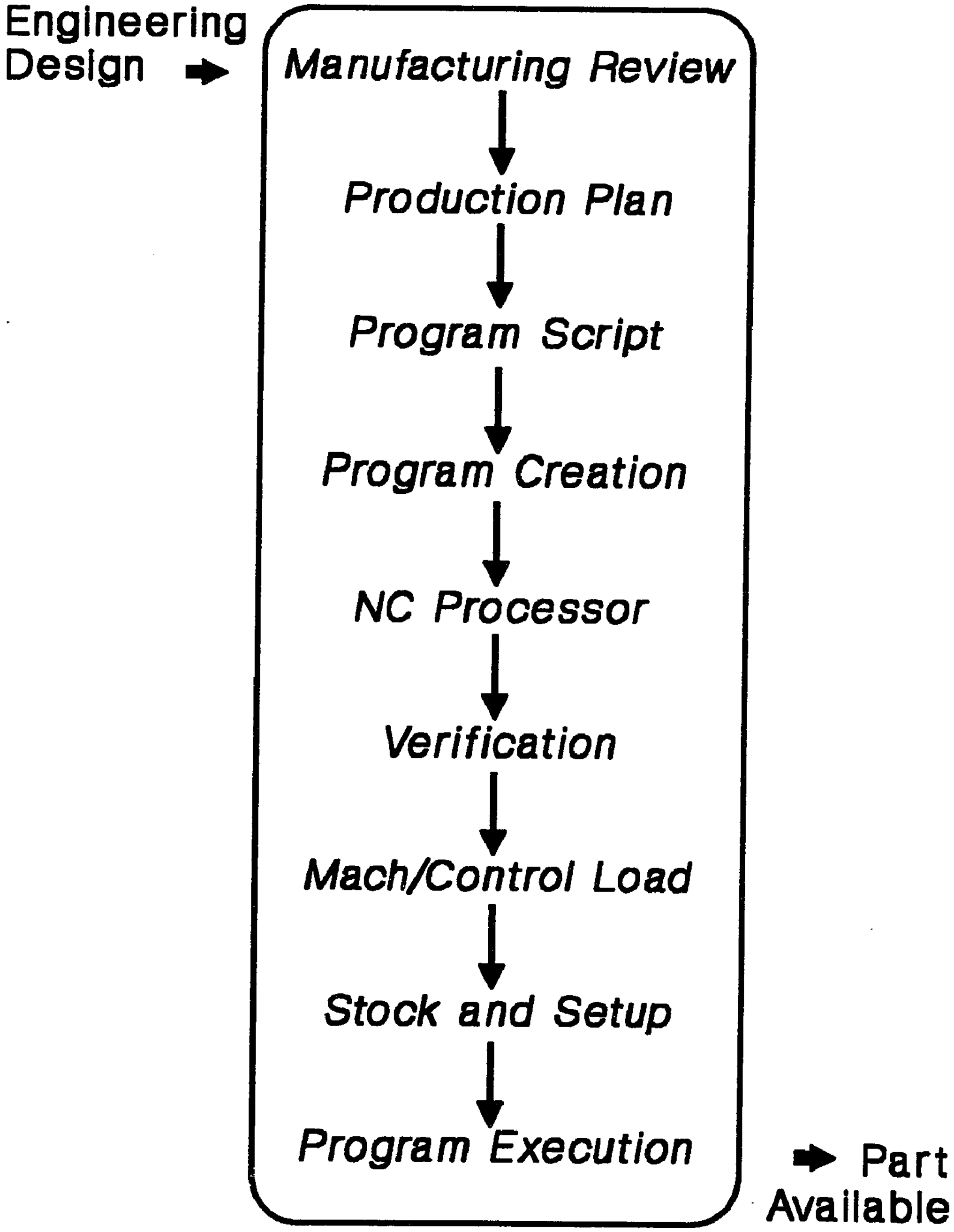

Figure 1. The numerical control process.

The literature has directed our thought to the significant effect that part program quality has on the numerical control process. In this usage, quality is the control or absence of part programming errors. Part programming techniques can be developed to identify and eliminate such errors once the source has been identified.

The data required in an NC part program has two basic forms - geometric and technical. The geometric data described in a part program includes a description of the component to be machined, a representation of the cutting tool, and the related path or trajectory control elements. This data could be extended to include a representation of the original workpiece and fixture. Geometric data is used by the numerical control processor to formulate a toolpath's center line data. The technical data included in a part program controls the machining operation. This data is exemplified by such parameters as spindle speed, feed rate, power requirements, tool changes, tool axis changes, and auxiliary functions required by a material removal process. Geometric and technical data form the substance of the part program data flow that instructs the machine tool controller to process a workpiece.

The author proposes that the quality of the geometric and technical data is the determinant of the overall NC part program quality. The indices of part program quality are graphically presented in Figure 2 on page 5. This figure renders each quality index about the geometric and technical data core. The nucleus and peripheral indices are formulated on a platform of "resource consumption." Resource consumption is the ever present cost consideration of how manufacturing process quality is achieved. Case [3] argues that " ... product cost cannot be simply measured in terms of material and machining costs." The resources expended in the numerical control procedure are quantified to include task preparation, program creation, program verification, methods analysis, and debug.

---

[3] Case, K., "Using a Design by Features CAD System for Process Capability Modelling," Computer Integrated Manufacturing Systems (UK), 1994, p 39.

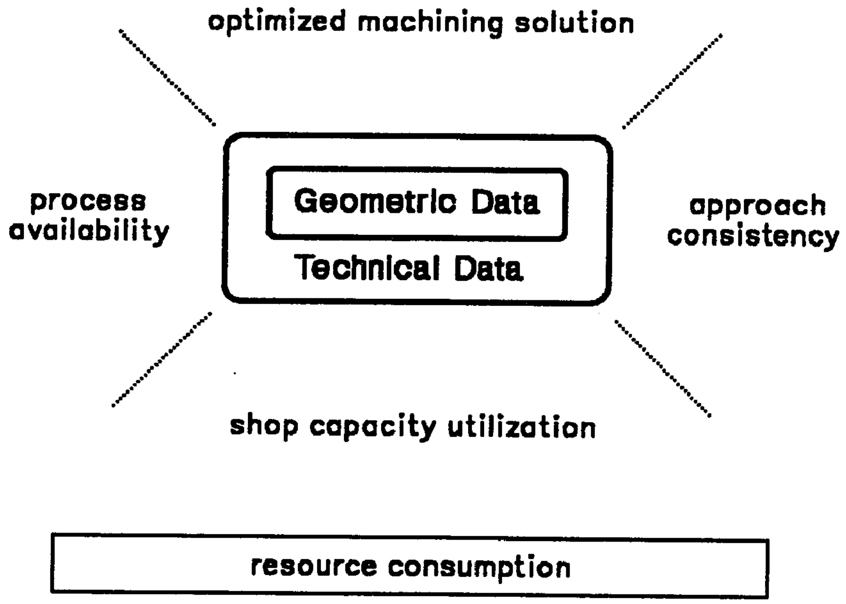

Figure 2. Numerical control programming overall quality.

The sources of error that are inherent in numerical control part programming data have many origins. Pressman and Williams [4] state that errors originate from "drawing (geometry) misinterpretations, mathematical errors, typing and format mistakes, and communication equipment malfunctions." Koloc [40] asserts that these errors can be syntactical, logical, or capacity related. The detection of these errors has been addressed by many part program checking schemes. Numerical control verification techniques are applied at the end of the process. This timing provides identification of part programming errors. However, this timing does not efficiently support a programming process that starts with basic data and builds on it to determine the processing for a part in an industry production machining environment. A more efficient approach places quality efforts in part program generation.

---

[4] Pressman, R.S., Williams, J.E., <u>Numerical Control and Computer Aided Manufacturing</u>, New York: John Wiley, 1977.

The literature illustrates the importance of the quality of numerical control code before it reaches program execution. However, the magnitude of this problem can only be realized when we place the numerical control process within the context of the general machining environment.

Pressman and Williams [60] use the criterion of number of parts produced and part complexity to portray the most efficient numerical control machining environment. Their perspective is presented graphically in Figure 3.

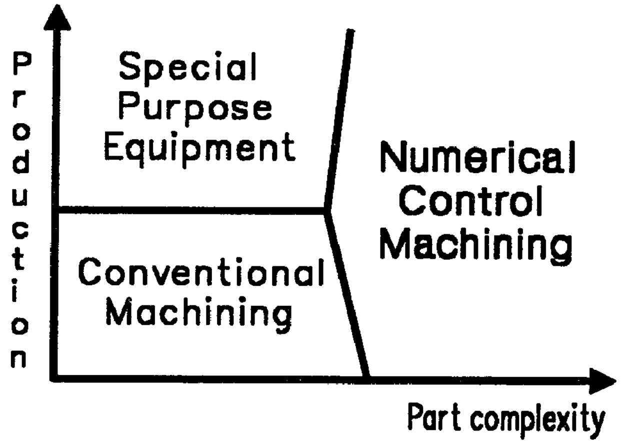

Figure 3. Numerical control machining environment.

This figure describes the numerical control environment consisting of a full range of production levels. Typical applications will continue to be found in the tool or model shop, job shop, production, and high volume machining environments. The part complexity most suitable for numerical control is relatively high. Together these criteria create a numerical control environment where the need for quality programs is increasing and the ability to manually or interactively provide adequate part programs is decreasing. Simultaneously, the opportunity for geometric and technical data errors is increasing.

1.1 Research Objectives and Approach

The objective for this research is to propose a new technique that can be applied to numerical control part programming for industrial machining. The new technique for industrial NC part programming will require an approach similar to that defined by Halevi [5]. He states that the problem can be approached by "technological improvements and effective utilization of available technologies." This author believes that the key to acceptance of this emerging technology can be found. The author proposes that the key is developing a comprehensive framework of automatic NC part programming within the environment provided by state-of-the-art solid modeling software.

The approach chosen for this research is to establish a foundation on the practical knowledge of numerical control machining with concentration in part program preparation. This knowledge is extended to include information on machines, cutting tools, setup requirements, operation parameter selection, and operation planning. Houten [6] states "an NC program can be considered the most detailed form of a process and operations plan." Theory is introduced to the study through the literature on manufacturing operations planning. Effort is taken here to understand and remain cognizant of the relationship with the process planning problem [25,34,52,69]. However, this problem is not the focus of this research. The practical knowledge and theory is then furthered in this study. This research effort will drive an implementation team for proof-of-concept to support a significant development of industry production NC part programming.

---

[5] Halevi, G., The Role of Computers in Manufacturing Processes, New York: John Wiley and Sons, 1980, p 278.

[6] Houten, F.J.A.M. van, PART: A Computer Aided Process Planning System, FEBODRUK Enschede, Netherlands, 1991.

2.0 Overview of Numerical Control Part Programming

Numerical control is traced to the late 1940s [21] with the punch card providing the mechanism to instruct machine movement. Since that time, the techniques for creating the instruction sets that define machine movement have made many stepwise evolutions. New techniques are introduced that have the potential of addressing the creation of these machine instructions commonly termed numerical control part programming. Groover and Zimmers [7] state "Numerical control part programming is the procedure by which the processing steps to be performed on an NC machine are planned and documented."

The industrial part programming problem has many facets which differentiates it from the NC laboratory. Not all programming techniques formulated in research are accepted and survive a reduction to industry production practice. These techniques are accepted by NC engineers, planners, part programmers, or toolmakers as they are able to solve current part programming problems and become production systems.

The two most notable techniques within this stepwise evolution have been the development of the Automatically Programmed Tool (APT) language and interactive graphics programming.

2.1 Automatically Programmed Tool

The APT language and programming method consists of five basic sections as described by Milner [8]. These sections provide a logical grouping of sets of instructions for the purpose as indicated by their names.

---

[7] Groover, M.P., Zimmers, E.W., CAD/CAM: Computer-Aided Design and Manufacturing, 1984, p 153.

[8] Milner, D., Vasiliou, V., Computer-Aided Engineering for Manufacture, 1987, pp 56-58.

The structure of an APT language numerical control part program is defined through:

1. Program Identification,
2. Geometry Statements,
3. Machine and Post Processor Statements,
4. Tool Motion Statements, and
5. Program End.

This language based part programming method is perhaps the most comprehensive and best known [29]. It has also influenced the development of numerous other part programming languages. A treatment of the APT part programming language exists in most every text on numerical control or computer-aided manufacturing. Many interactive graphics programming techniques use APT, or derivatives, as a guide for providing part programming functions. The terminology provided by the APT language has also seemingly become the defacto standard for terminology of NC part programming.

2.2 Interactive Graphics Programming

Interactive graphics programming [9] is the current state-of-the-art. The specification of geometry of the traditional language based part programming technique is replaced by the geometry extracted from a computer-aided design (CAD) model. The programming environment, machine and post processor statements, and tool motion are described interactively by the user on a graphics display terminal. This technique of part programming has obtained great popularity and user acceptance. This is due to the use of CAD based geometry and the visual verification or part program checking available in this programming technique.

---

[9] Milner, D., Vasiliou, V., Computer-Aided Engineering for Manufacture, 1987, p 66.

2.3 NC Programming Trends and Developments

The trends and new developments in the area of part programming for computer numerical control machining are important with respect to managing the contribution of this task to NC machining [10]. These trends and developments are being sponsored by a number of factors in product design and manufacturing. These factors are being created by both business competition and technological evolution. The business factors are stated as:

- **Increased Competition** for machining services. This competition is observed in terms of delivery, quality, and cost.
- **Shortened Design and Manufacturing Cycles** require that manufacturing processing be configured with very short lead times. NC part programming has traditionally been a "bottleneck" in maximizing throughput on numerical control equipment.
- **Design and Manufacturing Methodology Acceptance** is responsible for generating more design models that observe standard formats and are directly available to the NC part programming procedure.
- **Opportunity for Growth in NC Machining Market** is very extensive. Current market estimates place NC equipment accounting for only 10-12 percent of total machining capacity.

---

[10] The authors comments in this section are based on personal observations and available trade press.

Beard, T., "Automatic NC Programming Arrives," Modern Machine Shop, January 1994.

Christman, A., "A Short Course in Numerical Control Software," Tooling and Production, pp 54-62, July 1994.

Christman, A., "Software Learns Lessons from the Experts," Machine Design, pp 59-65, May 1994.

Colding, B.N., "Intelligent Selection of Machining Parameters for Metal Cutting Operations: The Least Expensive Way to Increase Productivity," Robotics and Computer-Integrated Manufacturing, Vol 9 No 4-5, pp 407-412, August-October 1992.

- **Enhanced Cost/Benefit Positioning of NC Machining** is the ability to justify numerical control method expenditures against the "real" costs of manufacturing as provided in the context of process re-engineering analysis.

The technological factors for the current trends in NC part programming are:

- **CAD/CAM System Maturation** is providing better capabilities to transfer and share design models. The expectations are being formulated that part programming techniques will use these varied forms of part specification directly in the NC procedure.
- **Geometry and Technology Integration** places expectations on the NC programming procedure. Toolpath generation is evolving toward the utilization of discrete technology to provide a more consistent exchange with the user.
- **Machine or Machining Center Complexity** is evolving the functions available on NC equipment. This factor places demands on the programming procedure used with this equipment.
- **Part Specification Complexity** is increasing. As such, the programming procedure is realizing a broadened requirement to handle this level of part specification complexity.
- **Hardware and Software Tools Evolution** provide the means for generating evolution in part programming procedures. This evolution is available to NC programmers at their desktop. This can be accomplished without the need to involve configuration management teams advocating concerns of possibly hundreds of CPU machine users.

These business and technology factors have been a catalyst for evolution in part programming methods and procedures. The trends of NC part programming offer evolution in many activities of the programming process.

NC Program Simulation and Editing

The trend in this activity is oriented toward both post-programming and in-process tasks. Post-programming simulation is being evolved to include more geometrical detail. This observation can be concluded from a review of programmed elements such as tool assembly visualization, fixture visualization, and stock visualization. Simulation is also becoming available in-process. This task provides "replay" of discrete toolpaths immediately following that segment of the part programming process. This promotes a better continuity in the application of enhanced editing capabilities. These capabilities promote an associativity between program elements and the part or originating geometry.

Manufacturing Technology Based NC Programming

Manufacturing technology is being introduced to the part programming procedure as an additional guide for the part programmer. The evidence for this trend is the movement away from programming procedures that rely solely on "tool tip guidance" across a geometry. The trend is toward tool trajectory specification that more closely maps to common machining practices. These methods are supported with a higher-level command stream while maintaining basic capability for tool tip and axis control. This trend is supported by the example of a general machining "pocket" trajectory. This trajectory is able to remove all the material in the volume of a pocket with a single specification. The routine also places technological elements in the support of this trajectory. Figure 4 on page 13 represents this trend in NC part programming toward manufacturing technology based programming. This figure presents this trend using a pocket machining trajectory. This figure is formulated from a simplified version of *pocket rectangular straight* by CAM-I [5].

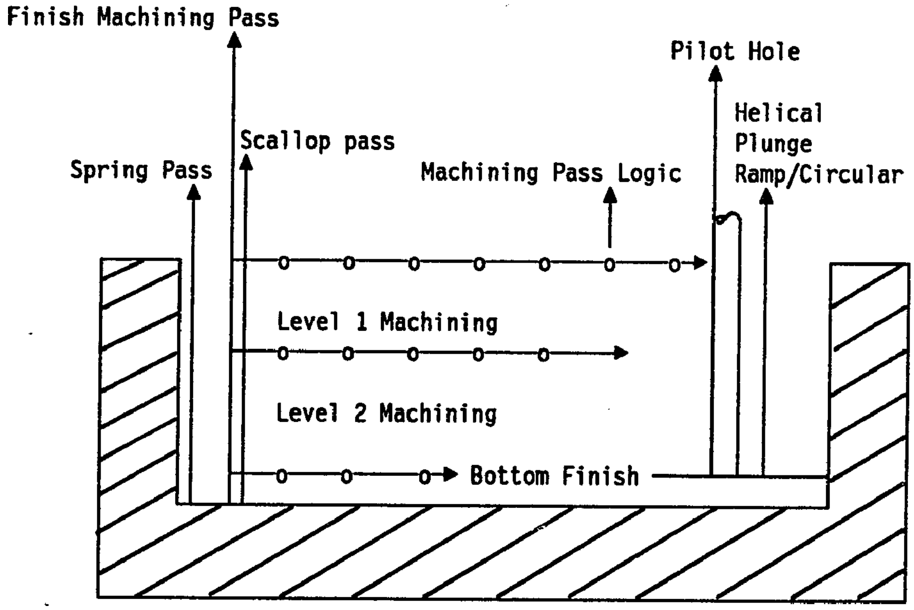

Figure 4. Manufacturing technology based pocket trajectory elements.

Tooling Support and Utilization

The current trend in this activity is to provide to the part programmer with a complete definition of the tool components and tool assembly data for available shop tooling. Software based "catalogs" are being established with methods to support efficient identification of a required tool or tool assembly. This trend assists the part programmer with information on available tooling and associated machinability data.

Feature-Based Machining Methods

New developments in NC part programming are introducing feature-based machining. Machining features are pre-defined manufacturable volumes with associated and parameterized machining methods. This trend modifies the form of part geometry. Likewise, the geometric portion of the part programming task is modified. The machining technology portion of the task is also modified for the part programmer by the coupling of method information to the feature entity. This emulates the family of processes concept by applying a variant approach to individual machining methods retrieved with feature definitions. Figure 5 represents the coupling of CAM-I [5] simplified features descriptions and machining methods in the context of a variant approach to creating a machining process.

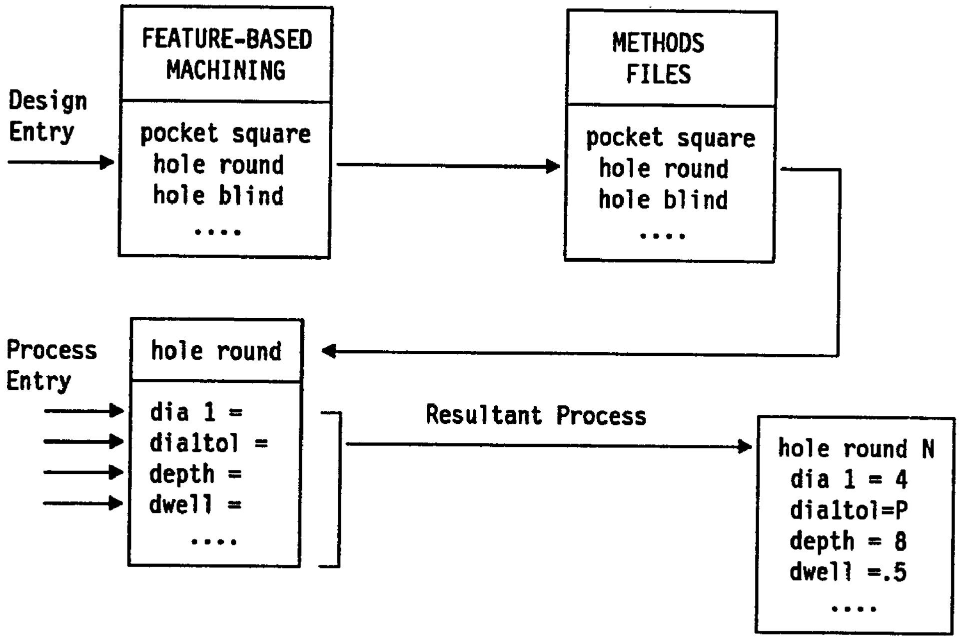

Figure 5. Machine features and methods coupling variant processing approach.

NC Programming Coupled with Machining Data

Machinability Data is being integrated into the data structures of numerical control part programming tools. This provides the part programmer with more discrete control over the cutting conditions that specify machining operations. Tool approach feedrate, retract feedrate, transition movement feedrate, slowdown feedrate, and finishing feedrate are emerging and being placed with the machining feedrate in the machining operation specification. This is further influenced by the availability of parametric control for both axial and radial depth-of-cuts within a series of path trajectories. The source of this machinability data is emerging as imbedded data tables and empirical formula computations.

NC Geometry Processor Evolution

Geometrical processing support for the NC part programming task is undergoing substantial evolution. The solid model, created in design phase, is being used as a direct source of machining geometry. This trend will re-kindle the notion of design and manufacturing working from a single model or data source. Tool offsets are being established with consistent handling from NC math routines. This relieves the part programmer from the task of geometric construction and promotes a simplified task when program change is necessary. Drive and check surface extraction from the basic geometry are assisting the part programmer through the utilization of the bounded geometrical definition of the part model. Part design geometry is being managed by the NC application to define machining across gaps, discontinuities, and overlaps in the surface modeling specification. NC part programming is being simplified by evolution in the geometry processor for each of these specified tasks.

2.4 Movement Toward Automatic Part Programming

The next front of development for numerical control part programming techniques has begun. Research efforts on the automation of aspects of part programming is underway. Early results identify a large variation in the application of the "automatic" terminology. Important research activity continues on the discrete topics and algorithms [15,24,26,47,79] within the scope of automatic part programming. The literature also contains limited treatment of state-of-the-art commercialization production systems based on this research effort [10,65].

The availability and acceptance to date of these techniques has been very slow. Joseph and Davies [11] explain that this rate of acceptance of this new technology is due to the "large amount of knowledge that needs to be gathered and encoded due to the broadness of the process planning domain." Jagdale and Wang [12] also characterize a contribution to acceptance of applications based on these new techniques. They state that "the difficulty lies in capturing and implementing the logic and methodology of the process planning function."

The industrial requirement for a next generation part programming solution is very real. Lee and Chang [13] state "even with computer-aided part programming, this can be very tedious and could slow down the entire production system." They continue "the automation of part programming is desirable."

---

[11] Joseph, A.T., Davies, B.J., "EXCAP - An Expert Process Planning System for Turned Components," First International Conference on Expert Planning Systems, IEE Conference, 1990, p 130.

[12] Jagdale, S.S., Wang, K.K., "An AI Based Generative Process Planning System for Machining Operations," Proceedings: Second International Conference on Industrial and Engineering Application of Artificial Intelligence and Expert Systems, IEA/AIE, 1989, p 528.

[13] Lee, Y., Chang, T., "CASCAM - An Automated System for Sculptured Surface Cavity Machining," Computers in Industry, 1991, p 321.

2.5 Computer Assisted Part Programming

This research examines the problem of NC part programming as required for industry production machining. Figure 6 graphically presents the elements of the computer assisted programming problem and goal as extracted from Milner [14].

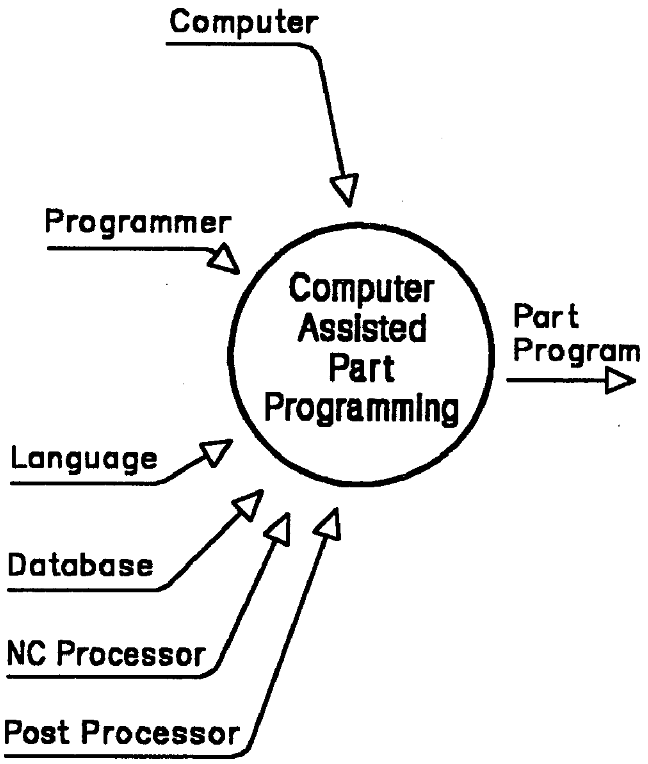

Figure 6. Computer assisted part programming.

---

[14] Milner, D., Vasiliou, V., Computer-Aided Engineering for Manufacture, 1987, pp 51-52.

2.6 Numerical Control Part Programming Environment

The next generation of numerical control part programming techniques can be approached by employing variant or generative based technology. Both of these techniques are introduced here in an effort to clarify the terminology. The broad spectrum of this terminology usage in both theoretical research and industrial early product systems requires this treatment.

2.6.1 Variant Part Program Preparation

The variant approach to numerical control part programming is based on group technology concepts [25]. The basis of this type of application is in the storing and retrieving of standard part programs. The variant approach begins with coding a part machining requirement and searching existing standard programs for a similar code. This standard program is retrieved and then modified as required to meet the specific machining requirement [36].

The variant approach has several inherent weaknesses. Jagdale and Wang [15] characterize the weaknesses in this approach to NC part programming. Machining requirements are typically "addressed at the part level." These parts must also fall into the specific "classes that are supported by the application." The general machining problem will require a large number of programs to be resident and available within the system. This storage requirement for and by "complete part programs" makes it cumbersome to work within a variant technique. At the completed program level, it is difficult to integrate both growth in the manufacturing facility and advances in manufacturing technology.

---

[15] Jagdale, S.S., Wang, K.K., "An AI Based Generative Process Planning System for Machining Operations," Proceedings: Second International Conference on Industrial and Engineering Application of Artificial Intelligence and Expert Systems, IEA/AIE, 1989, pp 528-529.

2.6.2 Generative Part Program Preparation

Generative techniques first emerged as possible solutions to the computer-aided process planning problem in the late seventies [27] These approaches attempted to create the required operations and operation sequences for a component from a manufacturing database without the need for human intervention. Chang [16] described the generative approach as "a system which synthesizes process information in order to create a process plan for a new component automatically." However, this definition is relaxed in the literature. Often, a system that handles a discrete task with built-in decision logic will be categorized as "a generative system." The generative approach to numerical control part programming is based on the availability of *first principle knowledge* encoded in a system. This approach is more versatile than variant techniques when applied to the general machining problem. This is because the system is based on manufacturing reasoning and logic instead of higher-level planning experiences. Using the generative approach, the operational planning and NC part program elements are synthesized from low-level machining information. This approach more readily supports the automatic creation of NC part programs. Joseph and Davies [17] state that this can be accomplished by "analyzing the part geometry and the other factors that influence the manufacturing process." They further state "generative systems should be universal in their applicability, the ones developed so far can only handle a limited range of components." Chang [8] identified the advantages of the generative approach as *rapid solution availability*, *solution consistency*, *ability to handle new components*, and the *potential interfaces to automated systems*.

---

[16] Chang, T.C., "TIPPS - A Totally Integrated Process Planning System," Ph.D. Thesis, Virginia Polytechnic Institute and State University, UMI, 1982, p 247.

[17] Joseph, A.T., Davies, B.J., "EXCAP - An Expert Process Planning System for Turned Components," First International Conference on Expert Planning Systems, IEE Conference, 1990, p 130.

These advantages are precisely the reasoning for the current effort in the subject of NC part program generation. Today, almost thirty years after Niebel [56] presented his initial work on the topic of *mechanized process selection*, there is no truly generative system for industrial production NC part programming. A survey of the current literature will identify the varied approaches and their contribution as applied to this research topic.

This research examines the development of a framework for generative part program preparation that overcomes the traditional limitations of generative systems. These limitations can be categorized as:

- **System Components Selected** for a generative system each possess a set of inherent limitations. This limitation constrains the available solution domain and implementation efficiency.
- **System Architecture** of a generative system provides a core of processing capability based in the *system components* assembled and *system functions* available. Limitations have been introduced historically by the sequence of processing and utilization of intermediate results.
- **System Functions** invoked by the *architecture* have historically limited successful implementation of generative systems. The user intervention required to complete a solution is a significant symptom of this limitation. *System functions* are not available or are incomplete in their handling of the target problem.
- **System Data and Know-how** is the source of information that enables the system to generate a manufacturing plan. This "information available" limitation, although a valid contributor to a successful implementation, is viewed as a temporary condition. Concentration in this research will be much more effectively placed toward results in the other historical limitation categories characterized as being fundamental [18] to a generative system solution.

3.0 Generative Numerical Control Part Programming

Generative numerical control part programming provides knowledge-assisted creation of NC processor commands for the machining of a part which has been designed or specified with features technology. Kochan [18] states "generative numerical control operation sequences are determined automatically by the system."

The generative technology can be developed to increase the productivity of manufacturing engineering and parts programming relative to the manufacture of computer numerical control (CNC) machined parts. Beard [19] states "G/NC ... offers tremendous possibilities to automate one of manufacturing's most time consuming and error-prone tasks." Other benefits include:

- Structured Approach to Part Programming,
- Reduction of Manufacturing Lead Time,
- Labor Skill and Competency Balancing,
- Application of Company Standards,
- Significant Increase in Quality, and
- Scheduling of Resource.

This technology can utilize the part design information provided by the engineer or designer who created the representation of the part. This information, known as the part model, will be the basis for appending manufacturing information to create a manufacturing model. The manufacturing information will include completing the feature sets and defining any additional manufacturing processing related geometry.

---

[18] Kochan, D. CAM Developments in Computer-Integrated Manufacturing. 1987, p 170.

[19] Beard, T. Automatic NC Programming Arrives, Modern Machine Shop, January 1994, p 87.

The goals of advanced numerical control are proposed by Kochan [39].

- Correction Reporting to the User.
- Reliability / Accuracy of the Solution.
- Automatic Processing Across Discontinuities in Part Geometry.
- Generative Processing Based on Low-Level Know-how.

The author can characterize a metric for achievement of these goals for advanced numerical control as manufacturing processing output. The type and quality of this output is critical to the degree of implementation of each goal. The goals are translated into elements of the possible output from generative numerical control part programming. Typical outputs are presented below.

- **Programming Errors** as a record of correction requirements.
- **Decision Data** supporting system direction or logical branches.
- **Workpiece Geometry** as the final sate of the workpiece.
- **In-Process Workpiece** as the dynamic state of the workpiece.
- **Tool List** as a record of tool assemblies used in the program.
- **Part Routing** as ordered operations with tooling.

Let us overlay the metric identified for the Kochan goals of advanced numerical control onto the graphical presentation for Milner's [55] elements of computer assisted part programming. The author proposes that this configuration of inputs and outputs to computer assisted part programming holds promise to achieve the goals as defined for advanced NC. This configuration will also provide a context for the development of generative part programming.

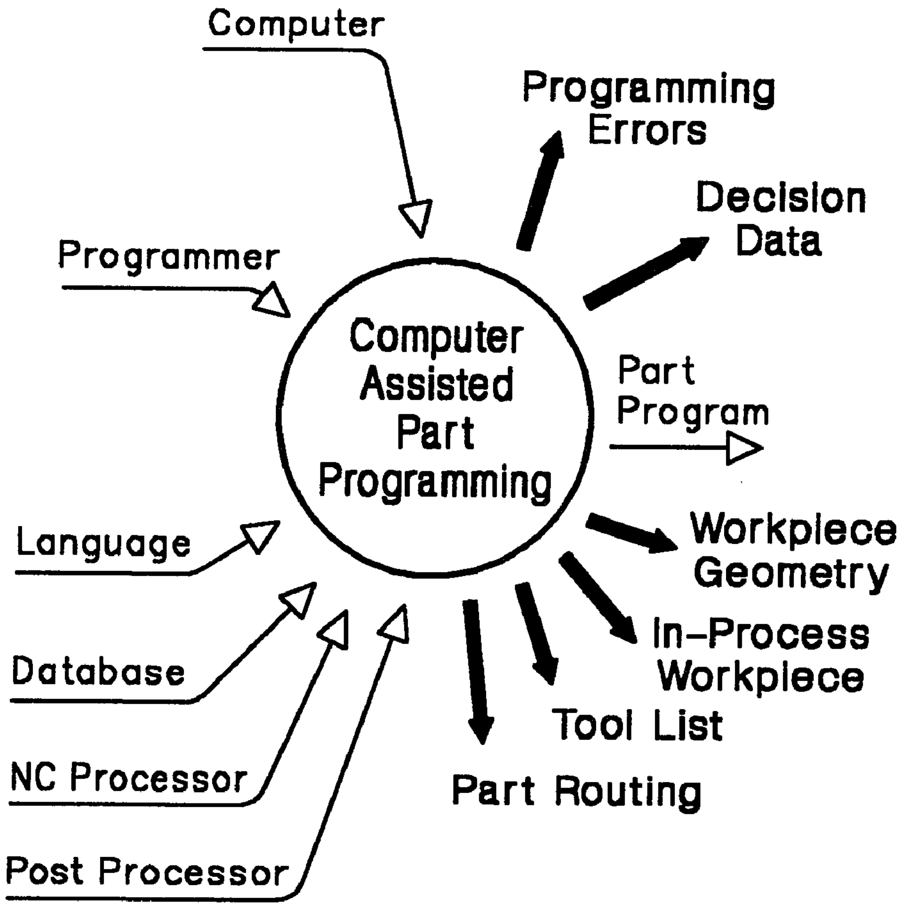

Figure 7. Computer assisted part programming toward advanced numerical control.

3.1 Research Domain

In this research, the computer assisted numerical control programming problem is approached from two specific focus directions. These directions, *part program quality* and *methods productivity*, provide a formidable challenge and problem approach domain. The problem approach has been prepared in Figure 8. The level of detail chosen is to guide the reader, rather than provide an extensive definition.

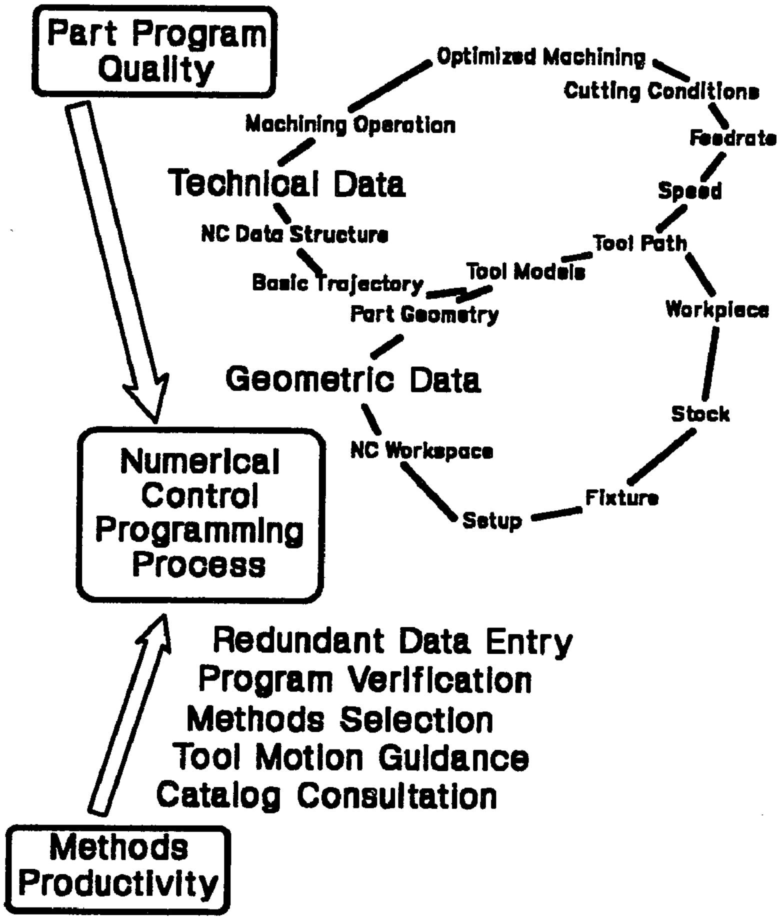

Figure 8. Research problem domain and focus directions.

3.1.1 Part Program Quality Focus

The part program quality focus originates with geometric data as a principal contributor to part program quality. A data chain is formed by the part program elements that are bound by the geometric nature of their contribution to the numerical control process.

- Part Geometry
- Tool Representation
- Tool Path
- Workpiece Geometry
- Stock Geometry
- Fixture Geometry
- Setup Data
- Machining Workspace

Technical data is supported by geometric data in its contribution to part program quality. This aspect of the part program quality casts a very different data flow chain. These items are bound by the machining technology nature of their contribution.

- Machining Operation
- Optimized Machining
- Cutting Conditions
- Machining Feedrate
- Spindle Speed
- Tool Path
- Tool Representation
- Basic Trajectory
- NC Data Structure

3.1.2 Methods Productivity Focus

This focus toward the NC part programming process is composed of a set of lower-level methods. These methods are representative of a user's engagement necessary to create an NC part program. These methods are established as concerns with current computer assisted part programming techniques [42].

- **Redundant Data Entry**

  The entry of part program data into the programming tool multiple times or in modified forms throughout the part programming session.

- **Program Verification**

  The checking procedures to validate part program quality through user visual program review, software replay, and production hardware tape try out.

- **Method Selection**

  The formulation of the detailed process plan that will be used to generate a numerical control part program specific to the production machining environment.

- **Tool Motion Guidance**

  The toolpaths specification task to describe the geometry and technology of each machining method through interaction with geometry.

- **Catalog Consultation**

  The access to and use of reference material that would contain data such as standard procedures, cutting conditions, materials, machines, and tool specifications.

3.2 Research and Computer Integrated Manufacturing

The identification and solution of the concerns that affect the creation of quality NC toolpaths have a strong correlation to the goals of computer integrated manufacturing (CIM). The advantages of a next generation part programming techniques are stated by Kochan [39] and adapted by the author as shown in Figure 9.

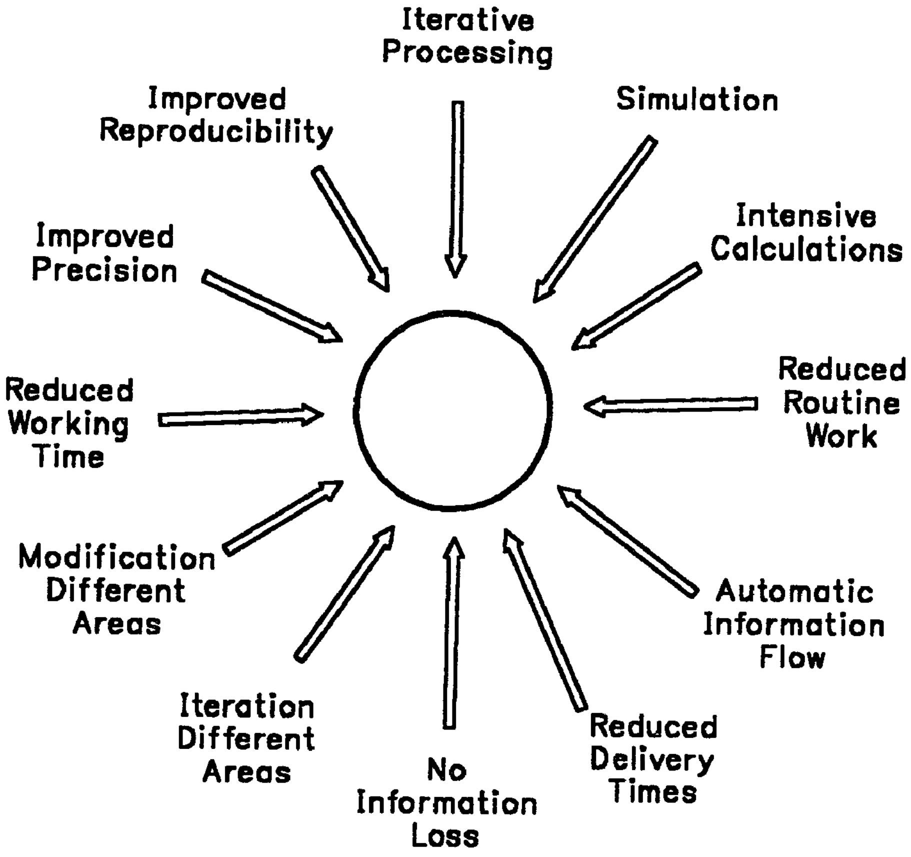

Figure 9. Computer integrated manufacturing goals.

3.3 Research Toward Automatic NC Program Generation

The research effort toward automatic part program generation is available from the literature. A selection of this research is reviewed here to establish the foundation of the current research. The available literature's relationship to the current research can be categorized as to the type of problem approached and the results obtained. The three categories proposed are:

1. Research formulating *algorithms* that provide a solution at the *sub-function level* of the operational planning and NC part programming problem. This literature is reviewed to establish *system level* requirements that may be inherent in the algorithm solution.
2. Research formulating a *system definition* that approaches automatic operational planning and NC part programming. This literature can be found to approach varied manufacturing processing. System level solutions for turning, two-dimensional cutting, and milling, and part inspection are reviewed. The findings are applied to furthering a *system definition for generative numerical control technology* with focus on prismatic parts manufacturing.
3. Research based *early commercialization* for next generation NC part programming techniques. The *system definition* for these efforts is focused to approach industrial production machining.

The key elements of each selected research are presented. The concepts brought forward as a foundation to participate in the development of a framework for generative numerical control are *italicized* in this text. A synthesis of these concepts is included for the closure of this section. The conceptual formulation of the framework for generative numerical control immediately follows this treatment of the available literature.

3.3.1 Algorithm Level Research

Vandenbrande and Requicha [72] present their work on feature recognition for mechanical components to support machining operational planning. This work employs a "generate and test" approach supported by rules-based processing. This approach is used to create volumes for machining on a typical 3-axis machining center. These volumes are formed with handling for stringent manufacturing constraints. Feature "targets" are defined, completed, and then verified. The verification of each feature includes checks for *nonintrusion* or gouging of a part surface. *Presence* or the contribution that a feature makes to the creation of a part surface is also checked. Verification continues with *accessibility* of the feature volume to the machine spindle. The final check identified in this work for feature validity is the application of *dimensional rules.* These rules govern both the absolute limits of the feature and limits imposed by machining practice. *Feature interactions* are formulated throughout the processing and represented for manufacturing system utilization. The process of feature recognition decomposes the complete original volume of the subject model. *Destructive Solid Geometry (DSG) techniques* are used to represent progress toward the feature recognition goal. The model is built using set difference operators on each feature volume with respect to the stock definition.

Hinds [26] proposes a technique for machining a complete part based on the assignment of a series of parallel planes. This technique can be applied to parts modeled using constructive solid geometry. The machining model is created from the solid geometry of the part by using tool offsets (+/-) with each primitive. This machining model is intersected with the series of parallel planes to determine a machining contour within each plane. Collectively, these series of parallel planes are assembled from "top" to "bottom" to form a machining pattern. This approach can be applied to both the *roughing and finishing* of the part.

Guyder [24] proposes a technique for machining a cavity. This method begins with the identification of a constructive solid geometry model for the cavity. The stock and fixture models are also defined as this time. These models are interactively used to define *containment regions* (volumes needing to be machined) and *avoidance regions* (volumes with no machining permitted). A parallel planes machining approach is used to extract machining contour information at each Z-plane. Attention is then given within each plane to optimally place toolpaths. Each toolpath observes a specified pre-drill location to permit tool plunge. These single plane toolpaths minimize tool retraction requirements. Rough and finish machining can both be approached with this technique. The technique has been developed to automate the optimal milling of cavities found primarily in mold and die machining.

Delbressine et al [15] present the notion of a "manufacturable design transaction" as a method to create part designs that take into account manufacturing restrictions. The principle component of this method is the "manufacturable object (MO)." The name of the system that implements this concept is Integration of Design and Manufacturing (IDM). Delbressine introduces the concept of *precedence based on reachability*. The factors that influence reachability are proposed as MO shape, position and orientation of the MO, position and orientation of the machine, and the shape with size of tools. This system determines the *manufacturing processes in the design phase*. This is accomplished by a direct coupling of a manufacturing process to the manufacturable object. The main focus of this work is to plan setups that consider reachability and the required accuracy of the geometry. Accuracy of the geometry is a characteristic of the "external tolerances" for an MO. These tolerances are presented as the allowable deviation in position and orientation. Reference is made to an *intermediate design state* that provides information to support the setup planning activity.

Khoshnevis and Tan [38] present a process planning module for generating all possible sequences of operations for a target hole. The approach taken is to use a set of manufacturing rules and a *forward chaining based rules processor* to determine a sequence of operations. The approach begins with a stock definition, in terms of a raw stock state, and works toward the final part state. The procedure for planning a hole is divided into three parts. First, a starting process is selected (spot drill or center drill). Secondly, a selection is made of a basic group of processes. Thirdly, is the breakup of these processes into rough, semi-finish, and finish operations. The Khoshnevis method *assigns a new feature to each intermediate surface* in a multiple operation plan. Each new feature is created as the difference between the goal state and the last intermediate feature. The planner will display the planned sequence for the user. This sequence of machining operations is subtracted from the stock model to verify manufacturing process completeness for the target hole.

Zhang et al [78] examine the machining operation sequencing problem. This work does not address machining operation creation. The approach used in the sequencing solution is to quantify the machining precedence relationships between features which are imposed by a set of domains. The precedence generating domains that are investigated by Zhang are determined as geometric relationships between features, datum and reference faces, feature interaction, fixture requirement, and good machining practice. The machining precedence relationships and approach directions are used to determine the optimal number of machine setups. Then, the feature sequence for each setup is determined with a minimal number of tool changes. Each feature is sequenced in the setup with treatment of the feature's multiple operations as a "machining macro." The concept of *machining feature patterns* is mentioned as having contribution to the efficiency of the sequence planner.

3.3.2 System Level Research

Bala and Chang [2] propose the algorithms to select the best cutter for pocket machining. They continue with their work to generate the toolpaths and NC code for the pocket using any user defined cutter size. Their approach optimizes both cutter size and cutter path generation. The algorithms have been designed for prismatic parts machining and implemented in the CUSPS system. CUSPS is the CUtter Selection and Pocketing System. This system is implemented as an interactive programming tool and is proposed to be an important part of an automatic manufacturing system. The work presented has the limitations that it can not address *pocket machining with multiple cutters* or three-dimensional pocket features. Primarily, algorithms controlling automatic cutter path generation for prismatic pockets are introduced. The system structure representing the Bala and Chang work is presented in Figure 10.

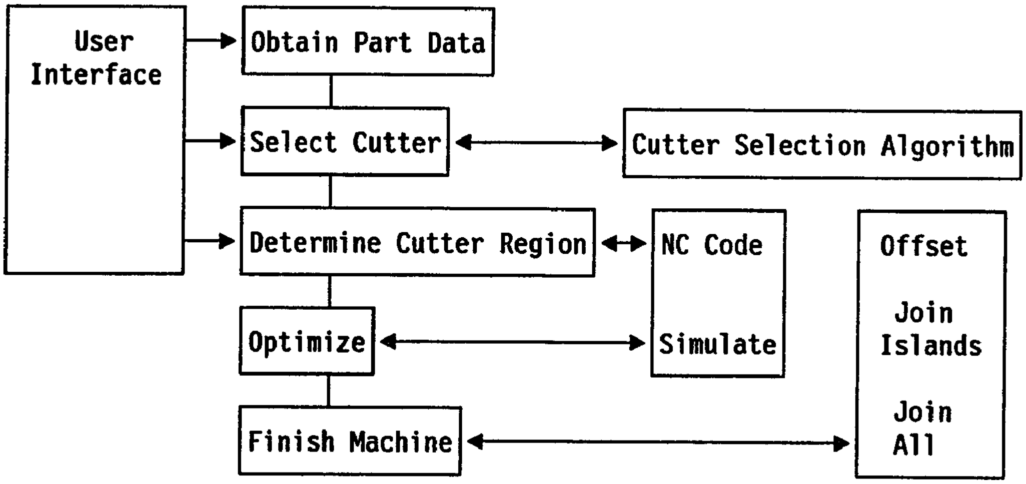

Figure 10. The Bala and Chang system structure.

Ghosh et al [19] examine the process planning problem related to strategy formulation in a two-dimensional cutting cell. The cutting technologies approached by this treatment are laser, plasma-arc, water-jet, punching and nibbling. Ghosh maintains that the diversity of the operational parameters associated with these processes requires a level of automation for effective planning. Process planning creation is more complex in this environment as a part's individual process plan must be nested with similar or dissimilar parts for manufacturing. Ghosh proposes a global space approach applied across the available capabilities of the cell to be a solution for the two-dimensional cutting of sheet metal. The *generative process planning strategy* for the Ghosh work is presented in Figure 11.

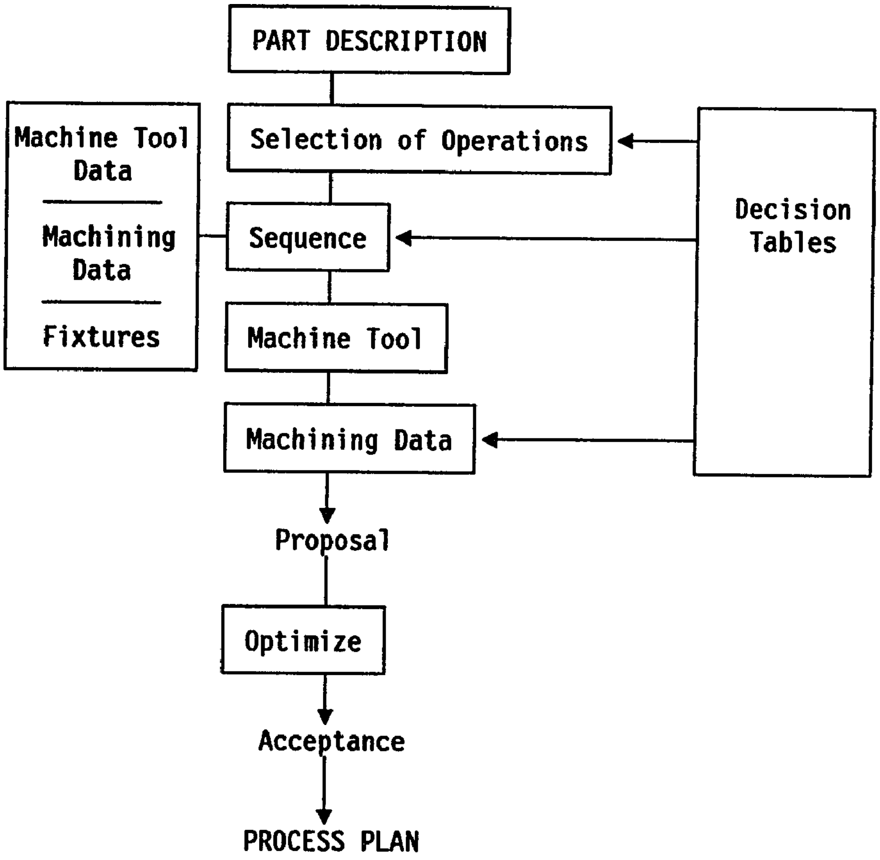

Figure 11. The Ghosh generative process planning strategy.

Marefat and Britanik [50] approach process planning from a case-based perspective. This development is proposed by combining the advantages of both the variant and generative approach to planning. This work is developed to operate on prismatic parts. The underlying concept in the case-based approach is that of "dynamic memory." This concept enables the reasoning system to use old experiences to solve new problems. This concept is supported by mechanisms for abstracting and storing feature sub-plans as shown in Figure 12. These feature sub-plans form the basis from which the planner learns of experiences. These experiences allow the planner to adapt and become more effective. This work makes use of *feature relationships* as it approaches individual feature sub-plan formulation. Feature sub-plans are merged through a hierarchical process plan graph. This hierarchy forms levels that identify features for common fixtures and then common tools within each fixture.

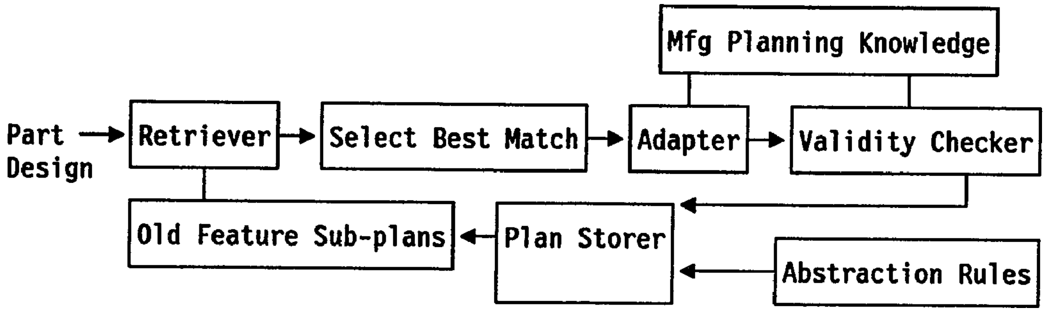

Figure 12. The Marefat and Britanik case-based planning approach.

El-Midany et al [16] present their concept of "AUTO CNC Part Programming." This concept is based on the interactive preparation of numerical control geometry coupled with a *machinability database*. The resultant toolpaths are produced with greater accuracy. The objective used in the work is the selection of proper machining conditions. These machining conditions are available to NC part programming through Machinability Data Handbook [20] data entered to a database. This system is implemented AUTOCAD and dBASEIV.

---

[20] A product of the Institute of Advanced Manufacturing Sciences, Inc.

Arikan and Totuk [1] approach a machining planner with the ultimate goal to produce a CNC program file. Their approach employs DSG techniques to create a part design. This design is created by successive subtractions of features defined with available processes from the stock. This approach ensures the machinability of designs within the context of the implemented manufacturing processes. *Process sequencing* is approached by the level of feature creation complexity. Facing requirements are sequenced first. The milling type features are sequenced within a level of complexity using surface height. Holemaking features are then addressed with conflicts being resolved by smallest diameter. Machinability parameters are added after the machining sequence has been defined. These parameters can be created by empirical calculations, entry from the user, or by searching personal databases. This system is given the name Computer Aided Design, Process Planning, and NC Program Preparation System (CADP-NC). Arikan and Totuk introduce the subject of *personal databases for machining parameters.* These files are organized by cutter types to hold data on tool material, tool diameter, speed, and feed. Endmill files will also record the number-of-teeth. This system is represented in Figure 13.

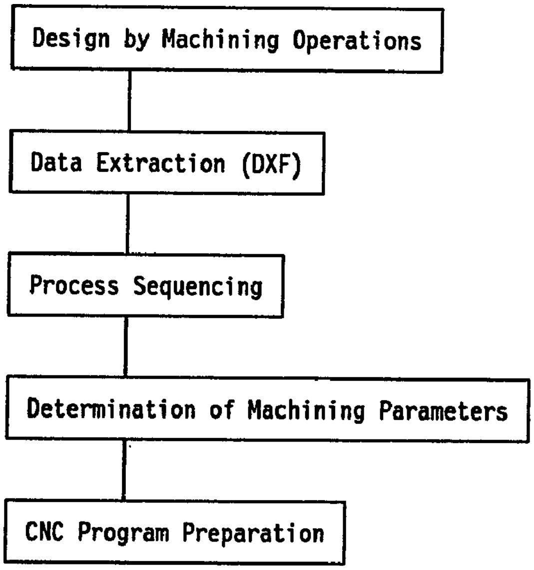

Figure 13. Flowchart for Arikan and Totuk system.

Wang [73] presents developments in applying solid geometric modeling capabilities to the problem of process planning. The focus of this work is the activities of NC programming, verification, tool design, and process optimization. The approach taken in this paper is to provide "useful tools" to significantly contribute to the process planner's productivity. This approach attempts to focus toward the concept of a generative solution. The system proposed has three components. A process technology module based on group technology is the core of the system. This core is supported by a geometric modeling module and a manufacturing database module. The topics that are introduced in this work begin with *identifying the volumes to be machined.* These volumes are manually created by the user as a representation of the machining requirements of a part. *Intermediate part geometry* is also introduced by Wang. This geometric model is built using a *tool swept volume approach* with successive subtractions being made from the stock solid. A dynamic cutting load analysis is proposed through the use of the geometry presented to the tool. This geometry is identified in the context of the intermediate part geometry. The metal removal rate is computed and used with unit horsepower to predict optimal cutting conditions. Wang concludes that solid modeling technology can be useful to manufacturing process planning problems.

Yeh and Fischer [76] approach automatic planning of machining operations for rotational parts. The system they present has modules for IGES design extraction, geometry plotting, CAPP, and G-code creation. Three databases support *tools, machines, and materials.* The *machine's working range* is checked against workpiece dimensions. The cutting speed and feed are computed using empirical equations rather than a tabular data source. The system is named Automated Machining-Operations Process Planning System (AMOPPS). AMOPPS is designed for rotational parts without internal bores. The system supports turning, facing, and grooving operations.

Matsumura et al [53] approaches the problem of operational planning for turning with an "adaptive prediction" model. This model sets machining operation parameters to minimize tool cost by simultaneously considering tool wear and surface roughness. The adaptive prediction approach combines operational planning data with on-machine experience. This approach dynamically adapts the operation planning based on the differences between the predicted and actual process. The experience based machining data is used to update the operation planning database. The contribution made by this work is the real-time optimization of operational planning based on the sampling of the actual machining process. The diagram representing the adaptive prediction procedure is presented in Figure 14.

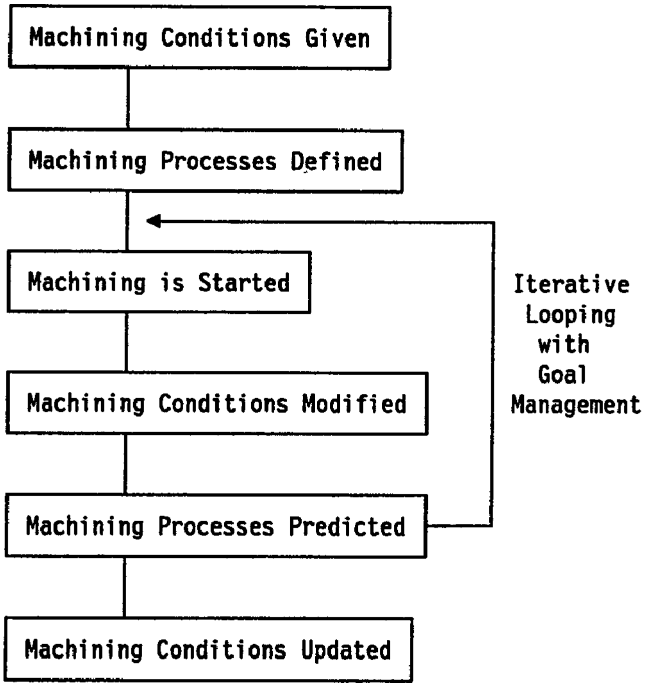

Figure 14. The Matsumura adaptive prediction procedure.

Weeks and Dickinson [74] describe an automated manufacturing system for sheet metal parts. This system is developed in terms of the essential and fundamental components required for a production system. The focus of this system is automation with open and flexible processing based in part on knowledge-based manufacturing. The authors present the fundamental components for such a system as *integrated*, *open*, *expert system*, *automated*, and *optimization.* These components are to be configured in an *automated flow.*

- An **integrated** software system would allow manufacturing direct access to the design database. This integration of the model database would provide "same kind of enquires" to be made with "shared functionality." Parametric capabilities of design can be applied to manufacturing data "family of parts" Associativity between design data and manufacturing is available through this level of integration.
- An **open** system will support *customization* of automated manufacturing through both the "user interface" and "underlying functionality." This fundamental component enables the replacement of system functionality and the associated calling procedures. The result is a powerful automated system that is easily extendable.
- An **expert system** can be employed for feature recognition and knowledge based machining rules. This approach *maps a design feature directly to standard manufacturing practices.* This approach restates the manufacturing problem into a *case of transcribing a machining sequence to a programming language.*
- An **automated** system will handle the generation of "machining paths" and "scheduling of the machine instructions." This scheduling activity can be supported with *indirect influence by optimization techniques.*
- **Optimization** of the "machining sequences" will be controlled for minimum machining time. The techniques presented to optimize machining are distance, time, area avoidance, and material considerations. *Minimum tool change* is discussed as a means to achieve optimization.

An *automated flow* will be necessary to define a system based on the previously stated essential fundamentals. Weeks and Dickenson present their view of this system. The system is a *structured and flexible automated flow.* Such a system must be *automated and supporting of user interaction.* The system must be *available at a workstation interactively and in batch sessions.* This system should be *unit independent* to support English and Metric data. *Databases* would be necessary to define *features, machines, and materials.* The critical component here is defined as a *standard, user replaceable, main control program.* The Weeks Dickenson system is described in the context of sheet metal manufacturing.

Tisza [71] presents an approach to the sheet metal forming. This treatment employs a knowledge based system structuring of a variant planning process. The principle of this work is a group technology classification system that combines both the geometrical and technological attributes of the part specification to code part families. The knowledge base system is utilized to select feasible process plans. The architecture of the Tisza system is presented in Figure 15.

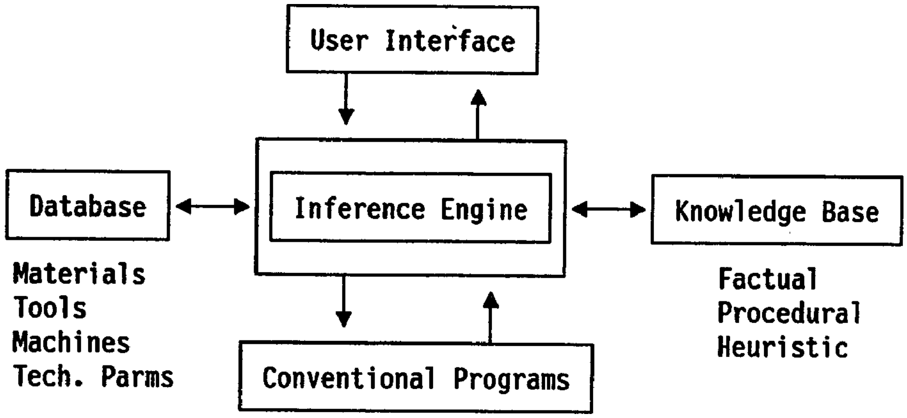

Figure 15. Tisza architecture with an inference engine system core.

Jain et al [33] provide a planning approach for Electro Discharge Machining (EDM). This work presents an interactive formulation for processing planning based on three primary modules. The first module, is concerned with the management of a machine tools database. This database would be current with the available shop capacity. The second module, provides the user with an interactive tool to identify machining requirements in the form of features. The third module, provides the core planning functions. Planning is approached with feature manipulations that provides for simultaneous machining of the base features. Planning continues with the selection of machining parameters and machining sequence. The machine tool is selected using both physical constraints of the part and the process capability. The result of the system is to store the plan with a part number key for later retrieval.

Brown and Gyorog [4] present the development of a generative process planning system for dimensional inspection. The name of their system is Inspection Process Planning EXpert (IPPEX). This system is configured with four components. These system components are *user interface, product modeler, relational databases, and artificial intelligence techniques.* The system is developed to generate the operational plan, inspection part program, and supporting information for Coordinate Measurement Machines (CMM)s. Planning for inspection equipment could contain *individual inspecting processes, process sequences, resource identification, and a detailed narrative.* The system is configured with the identification of nine activities. The diagram representing the main activities of the Brown and Gyorog inspection planner is presented in Figure 16 on page 41. The database support defined for this system provides details on the resources of *machines, probes, fixtures, and personnel.* The system has been developed to automate process planning and part programming for the inspection process.

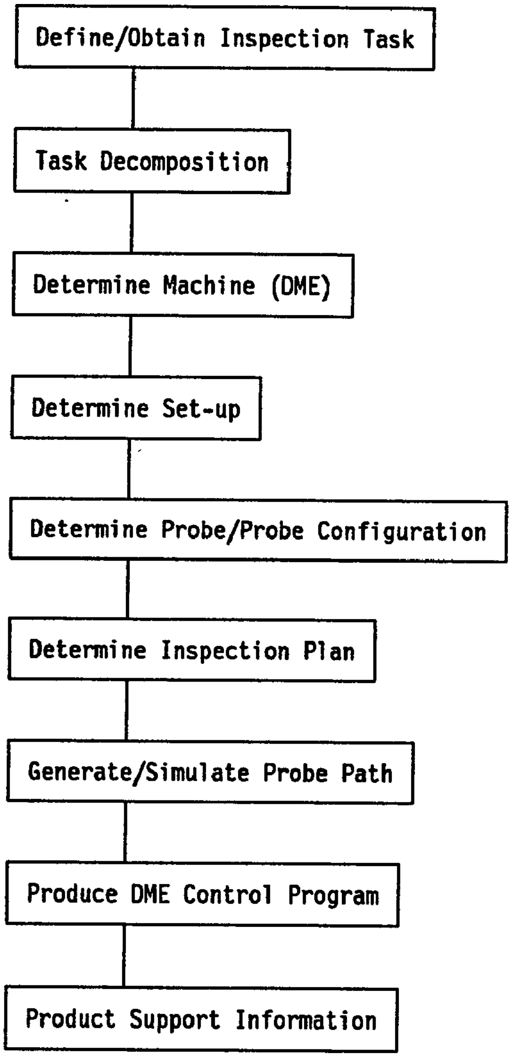

Figure 16. The Brown-Gyorog inspection planner procedure.

The expert system is applied to seven different decision making domains. Each domain establishes a separate rule base. The domains are dimensional measurement equipment selection, workpiece orienting, workpiece fixturing, probe selection, probe configuring, *inspection planning, and inspection technique.*

Gu and Zhang [23] present their work toward an object-oriented approach to generative process planning. This work is given the name Object-Oriented Process Planning System (OOPPS). The focus of this work is planning for a cellular manufacturing system. This system is proposed in Figure 17 to have a four-level hierarchical structure.

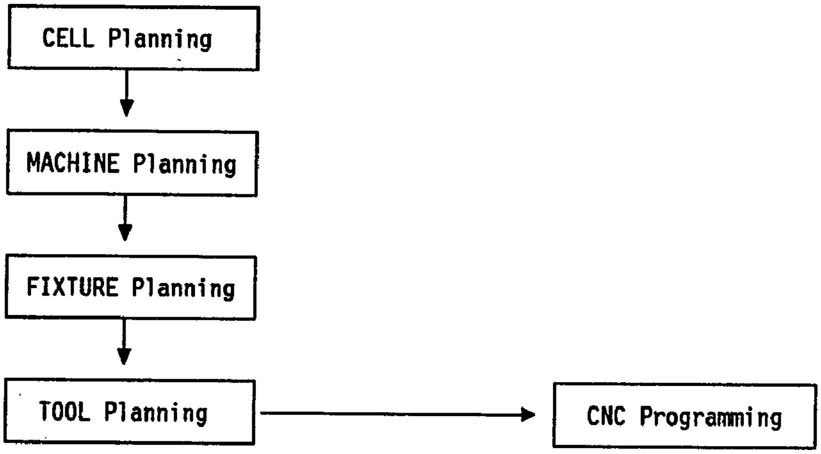

Figure 17. Gu-Zhang planner hierarchical structure.

Each of these planning levels is structured to achieve a specific goal. The goal of "cell-level planning" is to select a minimum number of machining cells for the part's manufacturing. The goals of "machine-level planning" are to select a minimum number of machines and fixtures on each machine. The goal of "fixture-level planning" is to select minimum number of setups on the machine. Likewise, the goals of "tool-level planning" are to select the cutting tools and cutting parameters required by the features in the setup. The *features are also sequenced within each setup* at this level in the planning hierarchy. The *creation of the CNC program from planning information is handled separately from the planner.* The placement of CNC programming is also noted in Figure 17. The diagram representing the activities of the four-level planning approach taken by Gu and Zhang is presented in Figure 18 on page 43.

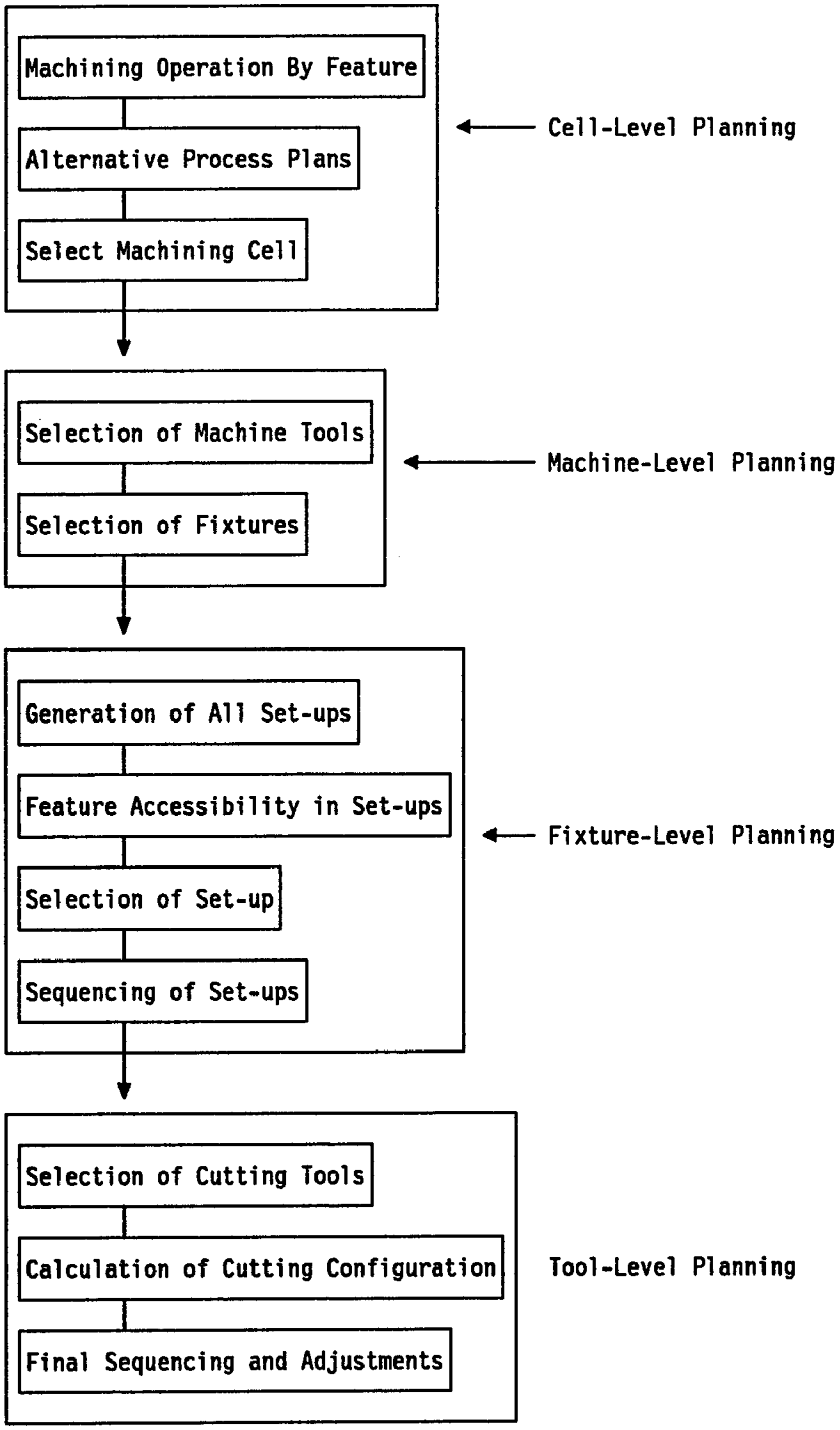

Figure 18. Gu-Zhang process planner.

Houten [27] approaches process planning with a system named Planning Activity Resources and Technology (PART). This planner was developed at the University of Twente, the Netherlands. The commercialization of this effort is being undertaken by ICEM Technologies. This planner is developed to interpret the product specification model and ultimately create NC part programs. This approach to process planning formulates an architecture where sub-tasks run in a parallel structure. Figure 19 presents this structure of the PART system.

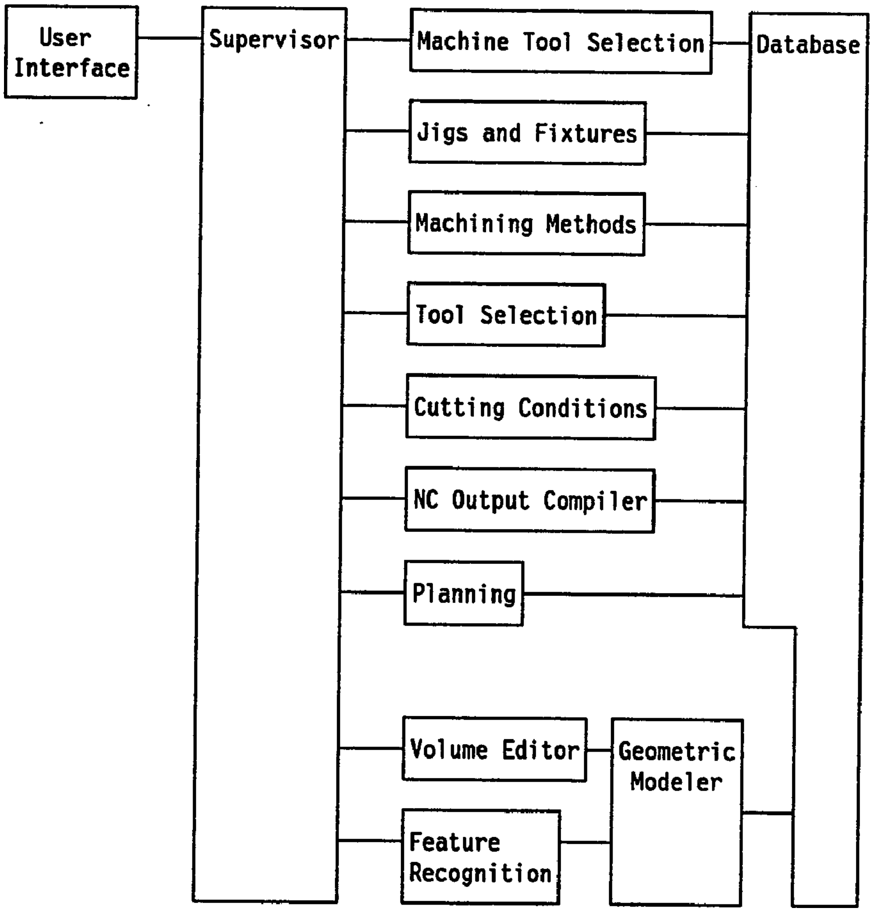

Figure 19. The Houten schematic structure to the planning problem.

The sequencing problem is addressed for the focus direction of *minimizing tool change and tool travel time*. The system defines a series of databases. These databases hold information on cutting tools, holders, adapters, fixtures, materials, and machine tools. For example, cutting tools are stored with *cutter edge, shank, and associated tool image*. There is no "expert system" defined within the architecture. Procedural structures are used throughout the system. The *NC processor* chosen by the PART system is provided through the NC functions of ICEM DDN. User control of the system is defined for two types of characteristic users. The *system administrator* will be responsible for all data and functions provided to the end-user. The *end-user* can access these data and function for the planning activities. However, change is restricted to the system administrator.

3.3.3 Early Commercialized Research

The I-DEAS [21] Generative Machining ™ [10,65] software is a commercially available system that captures "NC planning, tooling, and programming." This software is an *interactive system that assists the user* in these manufacturing activities. This description of the system is approached from the perspective of the software **general features, inputs, processing, and outputs.**

General Features of the System

- **Interactive Software** "assists" the user with planning and verification of a part program. The software can be executed in a fully interactive manner. Alternatively, the user can invoke methods files to assist in process data creation.
- **Run-time Process Change** are available in the system. Tools, fixtures, tables, and holders can be created during NC planning and programming sessions.
- **Design Change Re-Processing** are incorporated in the part program by re-processing a specific methods file or re-processing the part session.
- **NC Job Management** assists the numerical control planning and part programming "job" with the use of part, fixture, stock, and setup data. A solid model is defined for each component of this setup.
- **Library Approach** to tooling and fixtures allows user selection from these stored models.

---

[21] ™ I-DEAS Generative Machining is a trademark of the Structural Dynamics Research Corporation.

Inputs to the System

- **A Feature-Based Model** with solid definition of the part is used. This is supported with a limited number of standard feature types (pockets, faces, steps, single diameter holes, and double diameter holes, etc.) and associated manufacturing methods files. Software tools support the creation of additional *feature and method file pairs.*

Processing of the System

- **In-process Stock** can be recorded with each processing stage. This evolving feedback mechanism is used in visualization and can be the geometrical basis for related planning activities (ie. fixture design).
- **Methods Files** are used as a source of machining data and knowledge. These files provide machining conditions (speed, feed and depth-of-cut), tool characteristics selection, and machining strategy.
- **Machining Capability** is based on *volume clear, surface machining, and cycle* toolpath creation capabilities. *Volume Clear* provides rough, finish, and profiling. Depth-of-cut calculation is automatic based on part topology, tool capability, and stock allowance. *Surface Machining* is approached using variable or constant cusp height over the surface. Gouge checking is incorporated with surface machining. *Cycle Machining* supports the use of feature data to populate a call to *drill, center drill, ream, counter sink, counter bore, tap, bore, and plunge mill* trajectories.
- **Machining Sequence** minimizes machining time through operation order with *focuses on tool changes.* Machining rules control "drill before ream," shortest distance, tool sort, and increasing tool diameter.
- **Stock Engagement/Disengagement** control the entry and exit of the tool with the workpiece material. Methods to start a toolpath include plunge, ramp, path, circular, and drill. Toolpath end is accomplished by methods for lift, ramp, and circular.

- **Machining Technology** is introduced into the part programming task. *Intelligent feedrate control* provides entry of rapid, fast, slow, entry, exit, engage, and retract feedrates. These feedrates are used by the toolpath algorithms to make adjustments based on the monitoring of tool engagement in the workpiece.

Outputs From the System

- **Graphical Toolpath** visualization differentiates program element types (machining, engagement, etc.). This visualization supports a full tool display with the toolpaths. It also supports NC program editing capability. This capability allows post processor commands to be added to the part program.
- **Machining Time** is calculated from the planned machining operations and user entry made at setup time.

The Cimplex [22] Manufacturing Analyst ™ [10] software was the first commercially available "generative NC" system. This system is formulated by the principle that "any repetitious operation can be automated." Therefore, the system is build around a *family of parts* concept. Focus is placed on the NC programming task as opposed to process planning. The system is primarily configured as a batch system with interactive capability. This description of the system is approached from the perspective of the software **general features, inputs, processing, and outputs.**

General Features of the System

- **Automatic Software** generates the detailed plan and part program for the user. Alternatively, the user can interact with the system through the user interface.

---

[22] ™ Manufacturing Analyst is a trademark of the Cimplex Corporation.

- **NC Job Management** assists the numerical control planning and part programming "job" with the use of part, fixture, stock, and setup data. A solid model is defined for each component of this setup.
- **Library Approach** to tools, materials, and machine tools supports system parameter needs.

Inputs to the System

- **A Feature-Based Model** with solid definition of the part is used. Software tools support the creation of feature/method files.

Processing of the System

- **Hole Patterns** are used as a means to optimize the creation of machining operations.
- **Experience Base** includes features, methods, sequences, and machining parameters. These files provide machining conditions, tool characteristics selection, and machining sequences.
- **Machining Capability** is based on *open profiles, closed profiles, and pockets* toolpath creation capabilities.
- **Machining Functions** are performed with the NCL software from NCCS.

Outputs From the System

- **Machining Time** is calculated from the planned machining operations and user entry made at setup time.
- **Setup Drawing** is created as fully dimensioned output for the shop floor.
- **Graphical Toolpath** is approached through a solids based software. This visualization supports a shaded display of intermediate stages with the ability to save the "screen images."

3.3.4 Synthesis of the Prior Research

The prior research has provided insight into the solutions proposed toward automated planning systems. The intent of this section is the combine these proposals and project the goals of a generative and automatic solution. The structure used in this synthesis will be to organize the contributions of the literature toward utilization in a system definition quantified by *process, activities*, and *functions*. A translation is implied in this synthesis to move these research findings from "process planning" into "operational planning."

3.3.4.1 Database Support

The use of database is a common thread in the selected research. Arikan and Totuk [1] introduce the subject of *personal databases for machining parameters*. Yeh and Fischer [76] identify three databases to support *tools, machines, and materials*. Weeks and Dickenson [74] specify that *databases* would be necessary to define *features, machines, and materials*. Brown and Gyorog [4] have defined a system that requires databases for *machines, probes, fixtures, and personnel*. Houten [27] describes a database for cutting tools that are stored with *cutter edge, shank, and associated tool image*. Weeks and Dickenson add that the databases must support systems in a *unit independent* form to support English and Metric data. The need for a database to assist the automated operational planning and part program generation task is clear in the literature. This author will examine a refinement of this approach to database as the mechanism to hold *manufacturing environment object* instances. The objects are *machines, tools, standards stocks, materials*, and *machinability data*. The approach by this author is to define a *system component* that will provide a source of the *physical* and *logical* attributes of each object. The database will not require associated graphics or images. This database must be open in structure and content for customization to meet each machine shop's needs.

3.3.4.2 Intermediate Part Geometry

Intermediate part geometry is identified by Wang [73]. *Destructive Solid Geometry (DSG) techniques* are introduced as a means of creating this model. Delbressine [15] references an *intermediate design state* that includes the same notion. The foundation of the utilization of this technique is in *identifying the volumes to be machined.* The *tool swept volume approach* with successive subtractions from the stock is referenced as an approach. The important theme extracted from all these works has been that solid modeling techniques can be applied to create a *dynamic modeling* of the planning results. The author believes it is important to integrate this concept with *manufacturing objects*, a *specific NC processor*, and the *physical constraints of prismatic parts machining* to further develop this concept.

3.3.4.3 Features Support

The *feature* plays a significant role in most every paper on subjects relating to planning for machining. The feature is discussed in both the *design* and *manufacturing* context. Khoshnevis and Tan [38] *assign a new feature to each intermediate surface* they plan. Vandenbrande and Requicha [72] propose that *dimensional rules* should govern these feature definitions. Gu and Zhang [23] indicate that these *features are also sequenced within each setup*. The literature continues with a reference by Zhang for *machining feature patterns*. The author believes that the *feature* must be maintained as the central object in an approach to operational planning and NC part program generation. This feature can be approached in the *machining feature* context and applied to all the initial or secondary machining requirements for a part. Likewise, a *machining pattern* notion must be integrated within the architecture to promote optimal assignment and consumption of resources in an industrial production machining environment.

3.3.4.4 Features and Linked Processes

Discussion of *features* typically includes a concept of attached or linked processing methods. Delbressine [15] proposes a system that determines the *manufacturing processes in the design phase.* Weeks and Dickinson [74] indicate that using knowledge based machining rules in part design *maps a design feature directly to standard manufacturing practices.* Then, the manufacturing problem is simply a *case of transcribing a machining sequence to a programming language.* The author believes that a more efficient and optimal solution to the operational planning and NC part programming tasks can be approached by decoupling the feature and process pairs. This approach can provide access to a set of *first principle knowledge* that is employed to generate specific feature processing requirements. Khoshnevis and Tan [38] support this decoupling of feature and predefined processes. They propose a *forward chaining based rules processor* to determine a sequence of operations. This author believes that merit exists to examine a *backward planning approach.* This approach to machining planning would start at the part surface specification and work backwards toward the initial stock definition. This approach implies knowledge of *prerequisite conditions* to continue decisions toward the stock goal.

3.3.4.5 Feature Interactions

Planning toward NC programming generation is dependant on an assessment of *feature interactions* or *feature relationships.* This theme is prevalent in the work of Vandenbrande and Requicha [72] and Marefat and Britanik [50]. This theme provides the initial thought into the functional definition of *machining feature split and join* capabilities, *coaxial instances*, and *machining feature adjacency processing* leading to identification of *thin wall or thin floor* conditions.

3.3.4.6 Collision Detection and Analysis

Vandenbrande and Requicha [72] introduce *nonintrusion* or gouging of a part surface and *presence* or the contribution that a feature makes to the creation to a part surface. These notions can be combined with Guyder's [24] interactively defined *containment regions* and *avoidance regions.* The potential here is to define a *collision detection and analysis* function to be applied throughout the system to assist in the planning and verification of quality toolpaths. This proposal by the author will require consideration for *manufacturing objects modeling* and *dynamic workpiece.*

3.3.4.7 Sequence Precedence Orders

Many of the planning approaches are indicating an element of *precedence* to guide procedural efforts. Delbressine [15] uses a concept of *precedence based on reachability.* Vandenbrande and Requicha [72] verify a process with *accessibility* of the feature volume to the machine spindle. This concept can be formalized and expanded within a system definition to hold all the additional sources of sequencing order type data. This author proposes that *geometric access* data be combined with *technological* data and *user preferences* to complete the data requirements for machining sequence generation. This notion of precedence would also serve the system more efficiently if it can be applied to both the *machining features* and *machining operations.*

3.3.4.8 Process Sequencing

This topic is approached in many of the journal references. Arikan and Totuk [1] propose *process sequencing* by feature creation complexity. Weeks and Dickenson [74] propose a system concept for automation that involves *indirect influence by optimization techniques.*

They also consider *minimum tool change* to be a method to achieve optimization. Houten [27] proposes a goal to be *minimizing tool change* and *tool travel time.* These approaches to process sequencing provide a foundation for further research. This author contents that optimization of the machining sequence will consider *precedence, tool change, tool axis change, path distance, shop preference*, and *tool assembly management.*

3.3.4.9 Planning System Concepts

The literature has provided varied sources and system configurations. The commodity that a system has been designed for will tend to bias the resultant solution. This research focuses on a general solution to milling and holemaking for prismatic parts. Weeks and Dickenson [74] present the fundamental components for such a system as *integrated, open, expert system, automated*, and *optimization.* This type of system should be *automated and supporting of user interaction* available at a workstation *interactively* and in *batch sessions.* Tisza [71] proposes an architecture with an inference engine system core. Brown and Gyorog [4] indicate the system components to be *user interface, product modeler, relational databases, and artificial intelligence techniques.* Gu and Zhang [23] indicate that the *creation of the CNC program from planning information is handled separately from the planner.* There is a large variation in the approaches reviewed from the literature. This author examines the problem from the perspective that a solution must be definable in an industrial production machining environment. *System component* should be available and well structured for accessibility and maintainability. The *architecture* should be open to company methods and procedures. This architecture should have a fully automatic mode of operation. The *part programming functions* of this system require a comprehensive definition to enable a generative solution.

3.4 Framework for Generative Numerical Control.

A framework for generative numerical control will define and assemble the components necessary to achieve computer integrated manufacturing. Koloc [23] states "a high degree of automation can be exploited only if the initially prepared control programs are delivered as workshop ripe." This statement introduces the management of part program quality and methods productivity into a solution for generative numerical control. This statement also emphasizes the relationship of generative numerical control part program creation and the long-range goal of a computer integrated manufacturing environment. The goals of computer integrated manufacturing would also require a system that could be configured to meet specific needs. These specific needs would be apparent to support user specific methods, standards, and numerical control machine environments.

Generative NC part programming can be developed as an open manufacturing application. Lennon [24] states "while it is important to purchase a system with basic structure and abilities to suit the environment; it is equally necessary to be able to customize the system to give NC output in the form to best suit the organization and the people involved." The solutions defined within this research will strive to support the user methods for the integration of manufacturing knowledge and practice. The architected process flow is maintained for this open manufacturing technology environment. Weeks [25] states "above all, an automated manufacturing system must be open to customization at both the user interface level and at the level of underlying functionality."

---

[23] Koloc, J., "The Influence of the Programming Language on the Productivity and Reliability of Part Programming," Proceedings of the 13th Annual Meeting of Numerical Control Society, 1976.

[24] Lennon, M., Roth, R. Review of CAD/CAM Solutions for Numerical Control Part Programming, International Mechanical Engineering Congress, Sidney, July 1991, p 71.

[25] Weeks, N.J., Dickenson, S., "Toward an Integrated Parametric, Feature-Based, Sheet Metal Manufacturing Solution," AUTOFACT Conference Proceedings - SME, 1992, p 48.

3.4.1 Part Program Quality

Part program quality has been established as both geometric data quality and technical data quality. Figure 20 maps each of these contributions to the technology available to minimize each error source and manage part program quality level. From this mapping, the general concepts of the framework for generative numerical control are established and represented in this figure.

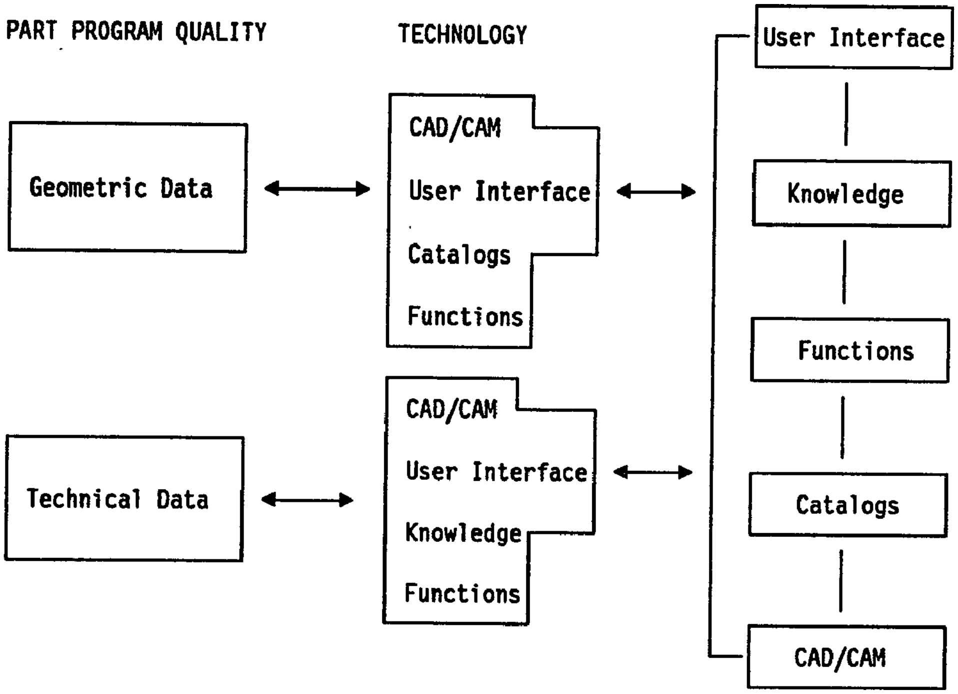

Figure 20. Part program quality and technology mapping toward framework concepts.

This exercise will also be undertaken for the methods productivity focus. The results will be combined and clarified for use in the development of the framework for generative numerical control.

3.4.2 Methods Productivity

Methods Productivity is previously developed to include the elements of Redundant Data Entry, Program Verification, Machining Methods Selection, Tool Motion Guidance, and Catalog Consultation. These methods are each mapped to available technology in an effort to realize the definition of the concepts of the framework for generative numerical control. Figure 21 represents the mapping of these elements of methods productivity to the available technology. Once again, the right side of the figure is used to organize these concepts of the framework.

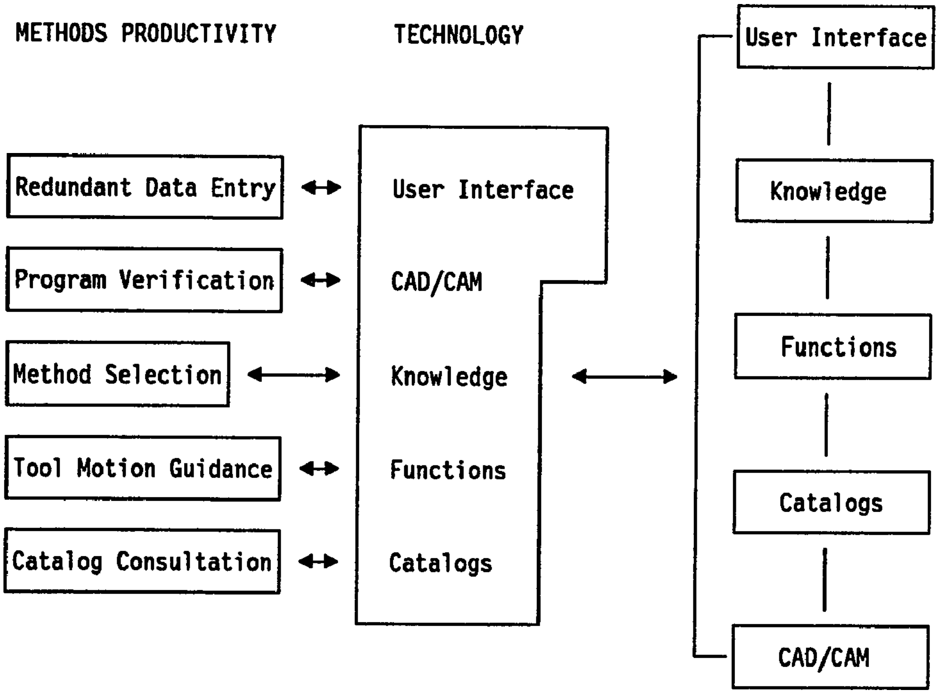

Figure 21. Methods productivity and technology mapping toward framework concepts.

The concepts established from each of the previous focus directions produce a consistent approach for a solution to *system definition* for generative NC part programming. This approach is corroborated by the available literature. These concepts introduce the basic technology that will be employed for a solution to the problem examined in the research.

- **Generative System User Interface** will be the guide to data entry and extraction for an automated manufacturing system.
- **Knowledge** will be the source of system available know-how for the operational planning and NC part programming solution.
- **Functions** will provide the manufacturing technology imbedded planning sub-problem solutions.
- **Catalogs** will provide the source of machining data and look-up access for the manufacturing solution
- **CAD/CAM** will provide integrated services for geometry and numerical control processing for an automated solution.

A framework for generative numerical control will be built to the general concepts as proposed in Figure 22 on page 59. This figure is developed by mapping the available technology to both the part program quality and methods productivity focus in this research. The basic concepts have been enabled for system development through the addition of an architecture level. This architecture will be developed to organize the combination of each concept. This architecture will also ensure that the synergy possible between these concepts is realized for a new technique to address numerical control part programming.

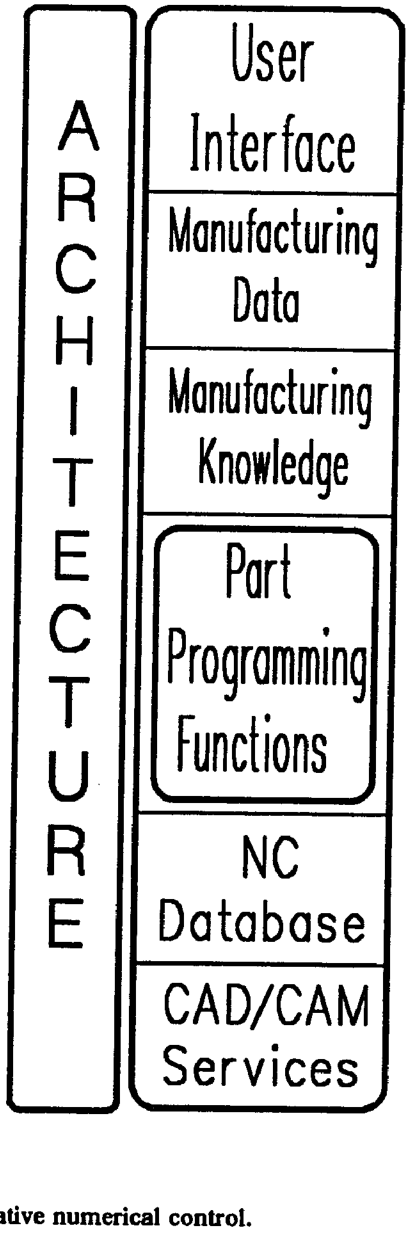

Figure 22. Framework for generative numerical control.

This framework will have the ability to transform the computer assisted part programming task as proposed by Figure 23. The framework is addressing each element to develop a next generation part programming technique that automates a solution to this task.

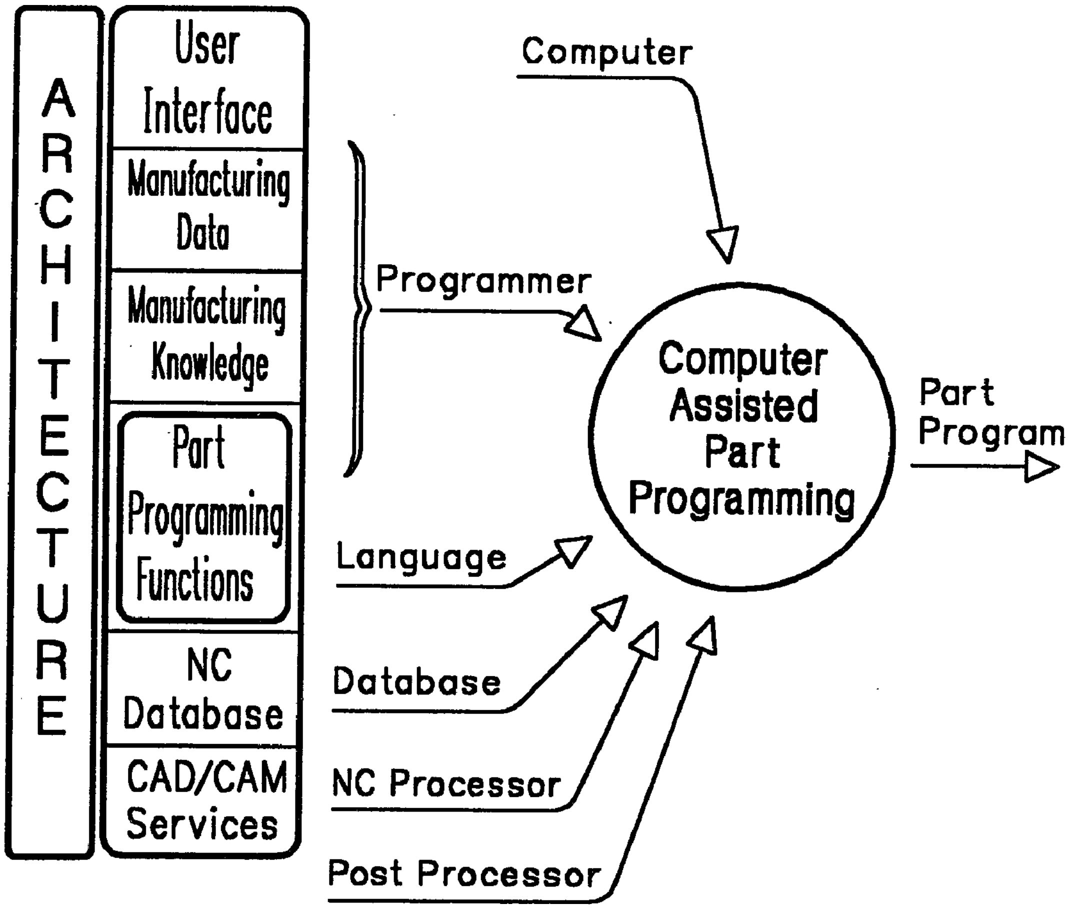

Figure 23. Research problem domain and framework.

The inputs to computer assisted part programming are approached somewhat differently in the framework for generative numerical control. The research problem domain remains consistent in the use of the *computer* and *post processor*. However, the amount of the *computer resource* necessary will increase as decisions are transferred from the programmer. The other inputs are evolving in form and content to establish an environment of advanced numerical control.

The *programmer* is the input element that is most dramatically transformed through the overlay of a framework of generative numerical control on computer assisted part programming. The programmer input will be designed to be minimized. This element may be developed to be constrained for initial input to automatic processing and then available for optional user review and edit sequences placed throughout the processing.

The *language* will be superseded by numerical control part programming functions that are available to and specified by the generative operation planner.

The *NC processor* will be superseded by a set of CAD/CAM services that will handle geometry within the part program. This element will also be developed to provide toolpaths and machining technology definition. The integration with the CAD/CAM system should allow geometry creation and manipulation that closely emulates the three dimensional world of machining. This creation and manipulation will be approached using consistent methods of the host CAD/CAM system environment.

Database will be superseded by a numerical control database. This transformation will evolve to a complete physical and logical data source for the machining environment.

The next generation of computer assisted part programming will be addressed in this context of the framework for generative numerical control.

3.5 Research and Agile Manufacturing

Agile manufacturing is an environment where the framework for generative NC can be implemented and contribute directly to the enterprise goals. These goals, in part, are establishing solutions that *enrich the manufacturer's customers.* This is approached through *organizational cooperation* where responsibility is shared. This organization is configured to *thrive on both change and uncertainty.* Agile manufacturing is approached through continued investment that *builds assets of people skills, knowledge, and information*

The nature of generative numerical control technology is an automatic solution to operations planning and part program creation for manufacturing processes based on computer numerical control machining equipment. This automatic solution is established through the assemblage of standard components in and about a leading edge CAD/CAM application. This system employes a "soft" definition of objects through a data dictionary. This technique supports the definition and interpretation of objects for use in a precise data structure established for the needs of the manufacturing problem solution. The imbedded manufacturing technology is open to modification and adaptation to return a part programming solution that is optimally generated to the current process capability of the machining shop. This solution employes configurable processing that assigns and consumes resource as required by the part specification as directed by the user. This emerging technology of generative NC is well suited to an environment where "process" is the basis of the exchange between a customer and supplier [26]. This is opposed to the more traditional "product" exchange. The traditional product exchange relationship is shown in Figure 24 on page 63.

---

[26] Miller, W., Preiss, K., Thompson, G., "A Model of Customer-Supplier Relationships for Agile Manufacturing," Agility Forum, AR94-08, 1994.

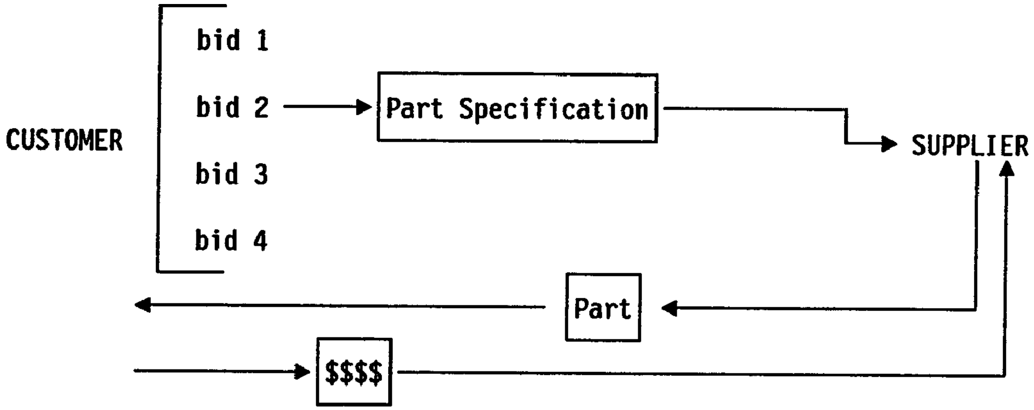

Figure 24. Traditional customer-supplier machining services relationship.

This traditional exchange between the customer and supplier is primarily "transaction based." These transactions are the *request for bid* that is made to multiple suppliers. The *job is placed* following bid approval. The supplier then *ships the part* and *payment is made.* This set of transactions is well defined with customer and supplier expectations at each level.

The contextual meaning for "process" would include an evolving solution. This solution can grow at an individual pace established by the customer and supplier as joint goals. Figure 25 on page 64 identifies a quite different dialogue that can be established with an implementation of the framework for generative NC within an agile manufacturing customer and supplier relationship.

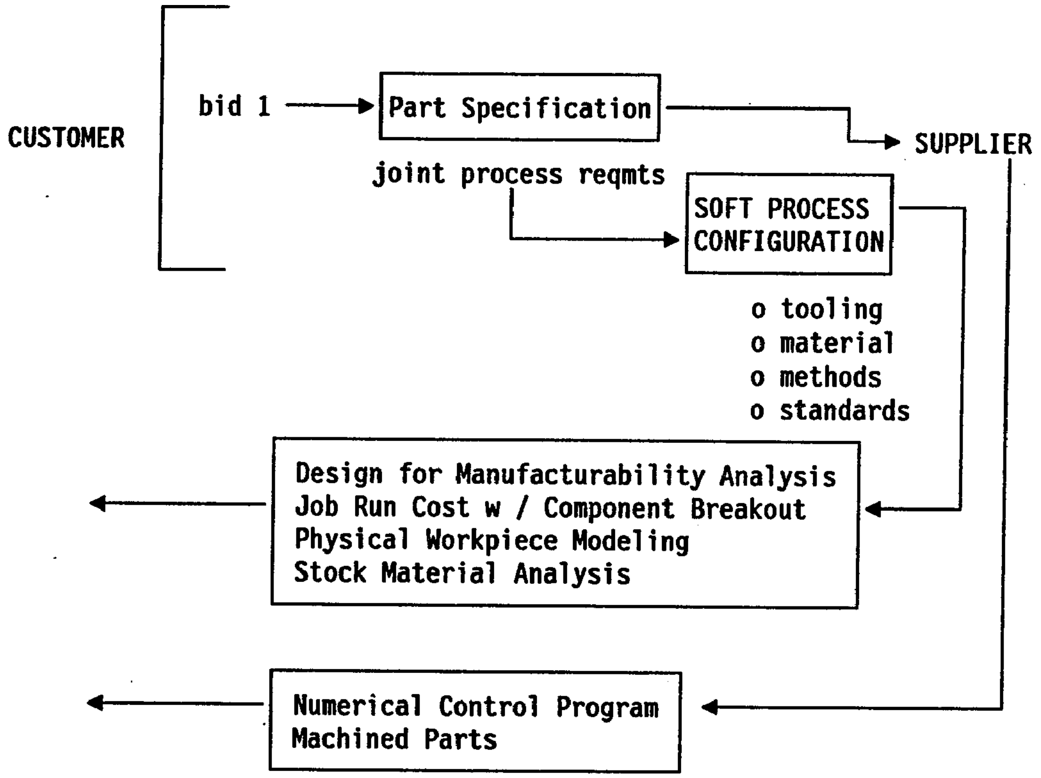

Figure 25. Agile manufacturing customer-supplier machining services.

The business practices support available to the *customer* from a *supplier* that implements the framework for generative NC include:

- **A Uniquely Specified Commodity, Part or Sub-Assembly**

  In the framework for generative NC, each request for machining is approached in a generative manner built on first principle knowledge and machining data. This provides for optimal workpiece machining based on available shop capability.

- **Pre-Sale Service of Just-In-Time Supply or Order Consolidation**

  The automatic processing available in the framework provides for scheduling and off-shift preparation of mathematically complete part programs. These part programs are available as required for machining and parts delivery.

- **Pre-Sale Special Design Service**

  Part models can be run at any stage of completion through the "soft process" provided by the framework. This process will identify machining trouble zones. The process will also identify efficiency in stock preparation.

- **Joint Product Design**

  This is accomplished by "pulling" the manufacturing knowledge available for generative NC to support product design's early decision making.

- **Joint Process Improvement**

  Machining capacity and process utilization are identified from the processes generated for given part specifications. This information provides data on where-used process components to formulate a basis for joint process re-engineering discussions.

- **Post-Sale Maintenance**

  Identification of the generative NC process as a part programming approach could be utilized in the case that machining services are being applied to the availability of spare and critical components. This will bring the customer and supplier together in this fulfillment activity.

- **Joint Marketing**

  The machining services customer and supplier can now work on the basis of partnership to establish precise delivery and cost schedules. This information is based on the data and knowledge generated throughout the generative NC process. This information is combined with process control and quality metrics.

3.6 Research Results Presentation

The framework for generative numerical control is developed in the following chapters of this dissertation. Chapter 4 provides a "System Components Definition." Chapter 5 establishes the "Development of the Architecture." Chapter 6 specifies the "Generative Part Programming Functions." The manuscript preparation conventions are to *italicize* terms that are developed in this research and integral to the communication of this system definition for generative NC.

This formalization of the research has been chosen to establish a progressive flow by which to record the framework. The components or building blocks of generative numerical control are first defined. The numerical control part programming task is then decomposed into the elemental problems. This decomposition is the basis of the architecture within the context of the components as previously defined. This architecture is ultimately given both breadth and depth with the specification of a set of functions organized by process and activity. These functions completely address the advanced numerical control part programming task. This set of functions is commensurate with a production solution to the goals of generative NC part programming. An Example Part is included in this presentation to introduce this next generation part programming technique. This example highlights the contribution made by the framework for generative numerical control on features extracted from a reference part configuration. This example begins with part geometry and continues through generation of machine code.

Recommendations for Further Study will address additional applications of the framework of generative numerical control as developed herein. This further study is proposed in the area of manufacturing design advisor, machine selection, tool/fixture design, and additional manufacturing processes support.

3.7 Research Results and Analysis

The analysis of research proof-of-concept systems and the commercialization of such work is a difficult task. The analysis criteria for next generation NC part programming solutions are taking a radically different form, just as the changes in the technology itself. The proposed criteria for such an analysis can be found with heavy bias toward the agile manufacturing concepts. The software "features and functions" priority of yesterday gives way to today's requirements of the user, openess to his in-house machining practices, and enabler for growth well into the future. An analysis of the implementation of the system definition proposed by these research could include:

- Documentation Quality and Completeness
- Part Programming Functions Completeness
- Solution Approach - Variant/Generative/Hybrid
- Solution Perspective - Automatic/Assist-the-User
- Level of Automation/User Interaction Available
- Machining Features Handling Capability
- Machining Operations Types and Control Level
- Level of Basic Tool Trajectory Control Available
- Database Support
- Knowledge Form and Content
- Knowledge Processor Integration with CAD/CAM Services
- Architecture - Manufacturing Open System
- Hardware/Software Configuration - Single/Multiple Vendor
- System Performance

4.0 System Components Definition

The generative numerical control framework is constructed from six fundamental building blocks. These building blocks are the *system components* defined by this research into the technical problems of a generative approach to the NC part programming process as stated in Figure 1 on page 3. This chapter creates a definition for these components unique to a generative numerical control system. These system components are:

1. Generative System User Interface,
2. Manufacturing Data,
3. Manufacturing Knowledge,
4. Manufacturing NC Database,
5. CAD/CAM System Services, and
6. Part Programming Functions.

These components have been carefully defined through this research and validated through associated proof-of-concept and development efforts. The modularity that is implied with each individual definition is deliberate. This modularity of components is an important contributor to the overall framework.

These system components are also fundamental to the development of next generation conventional NC part programming techniques. This type of implementation would take a different perspective to both *manufacturing data* and *manufacturing knowledge*. The conventional part programming approach will not establish and maintain the strong integration of these system sources of data and know-how. This becomes apparent through the substitution of user supplied data and know-how at each point of use throughout a conventional and interactive parts programming system [10,16].

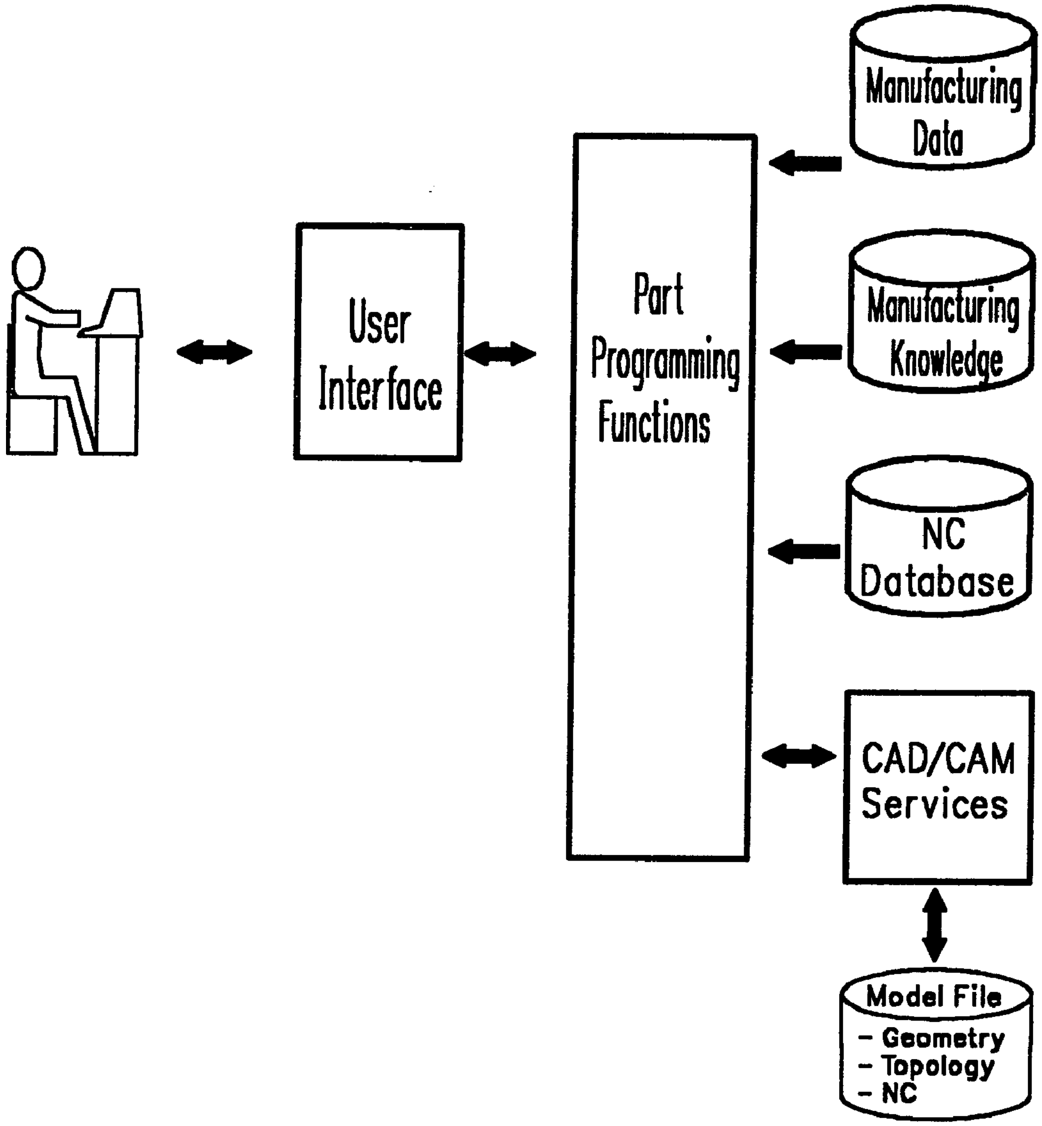

Figure 26. Topology for generative numerical control system components.

The system components are given a topological perspective by Figure 26. This figure also positions the user with respect to the components defined for generative NC. This user perspective presents an execution or run-time part programming scenario. In this figure, attention should be directed to the sense of the arrows providing connectivity for the system components to the user. The topology indicates that the user's contact during run-time with these system components is exclusively provided by the generative system user interface. The user can access the components of manufacturing data, manufacturing knowledge, and NC database with read-only connectivity. The CAD/CAM system services component is available with both read and write privileges. These relationships are further developed with each system component definition.

The connectivity defined from the topology of Figure 26 on page 69. provides a first indication of the system's ability to produce part programs that are generated by a structured approach. The structured approach to NC part programming begins with providing a consistent definition of the programming environment to each user. The user in this topology reads the system source for manufacturing data and knowledge into each part programming session. Thus, all users are provided with the same bound on the part programming environment. This topology can meet the goal of a generative numerical control solution that provides a *structured approach to the part programming* problem.

The definitions that follow will identify the significant elements of each system component. These *elements* will begin building each component's contribution to the generative numerical control framework. As appropriate, the component definitions will be compared or contrasted to other approaches in the literature. The *Part Programming Functions* system component specified as a result of this research will be addressed in Chapter 6. The contribution made by this component to the framework and its relationship with the *Architecture* directs this separation.

4.1 Generative System User Interface

The user interface is the *system component* that provides primarily run-time support for user interaction with an implementation of the generative numerical control framework. This component will be employed for the initial startup of the system. The user interface also defines the guides for development of user review and edit scenarios to be placed throughout framework. The development of the *Architecture* in Chapter 5 will place these optional opportunities for user dialogue within the system structure.

This run-time user has the task to provide the data entry to initiate the creation of a detailed machining plan. This operational plan will ultimately lead to the generation of an NC part program. This task is accomplished by a part programming function that provides access to the implemented *manufacturing data, manufacturing knowledge, manufacturing database*, and *part programming functions*. This run-time user is typically referred to as a part programmer NC engineer, or toolmaker. The run-time user task is shown in Figure 27 on page 72.

The *generative system user interface* has received a very limited treatment in the literature. This is evident from the works reviewed in Chapter 3 - Research Toward Automatic NC Program Generation. Research has generally identified this component as sensitive to the acceptance of the system. This component also provides a series of unique characteristics proposed for generative NC part programming technology. The unique characteristics arise from the fact that the generative NC technology by definition is available as a fully functional and automatic part programming solution. The research on automatic systems indicates that the point of data collection must be unique. This *unique or singular point of user interface* carries the requirement of both completeness and accuracy. The completeness requirement affects the processing selections and levels defined by the user. Data collection accuracy affects the usability of the part programming results.

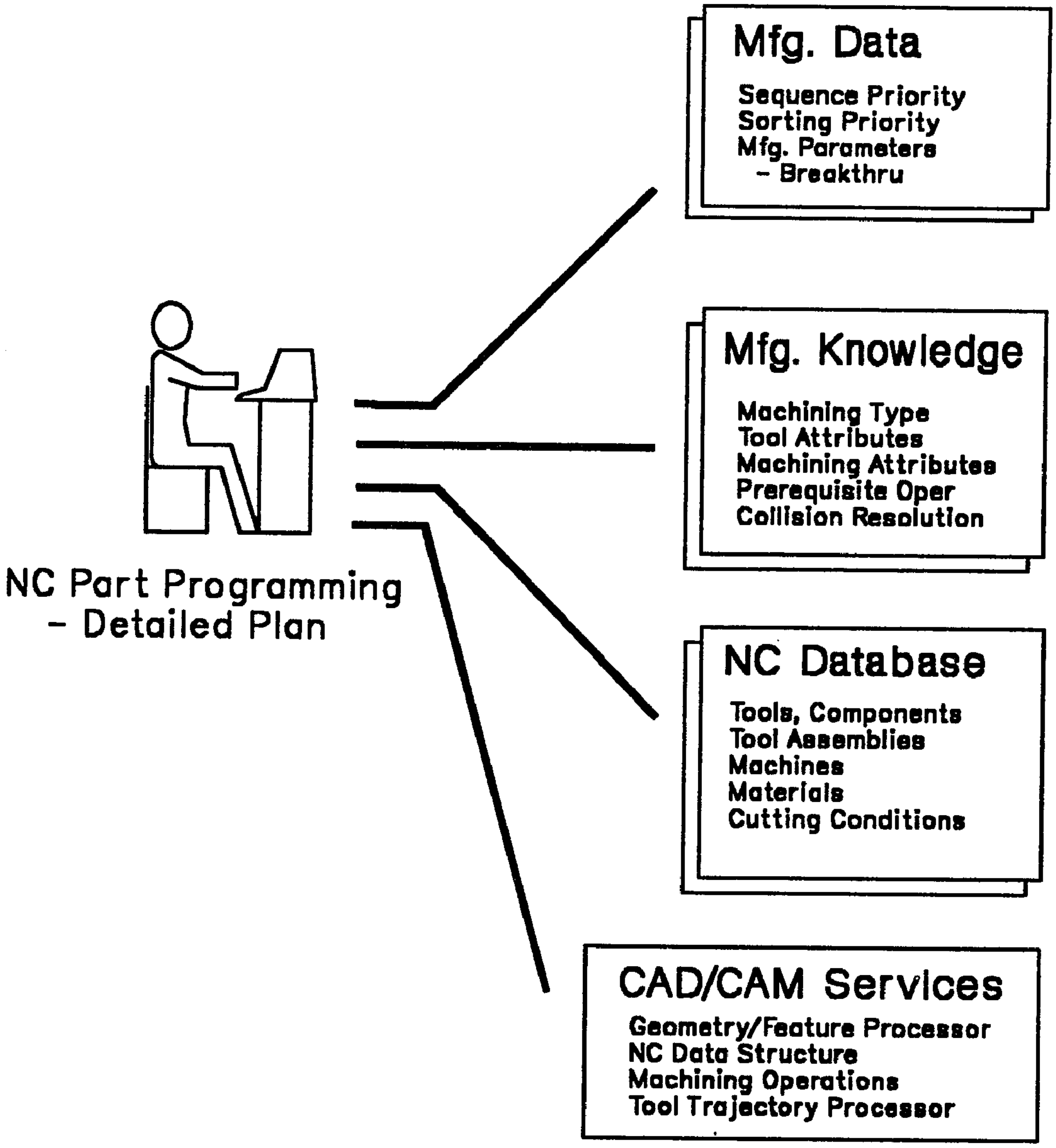

Figure 27. Run-time user perspective for generative numerical control.

Figure 27 on page 72. also provides a first indication into the depth of the generative solution provided to the run-time user of a system that implements the framework. The end-user is accessing a source of *manufacturing data*. The examples chosen for the figure display "sequence priority" and "sorting priority." This data is clearly a system source of know-how that is made available to the user. However, "breakthru" is a manufacturing parameter provided to the user. This parameter has an entirely different connotation to the user and system than sequencing and sorting priorities. Breakthru amount is a parameter used by the system to control final tool depth in "machine through" operations. Holes and cutouts are good examples of the features that may require these operations. This "allowance" for material variation provides trajectory specification that results in complete material removal.

Similar insight is provided by Figure 27 on page 72. into the types of data handled by the user interface. *Manufacturing knowledge* provides rules that are available to select the type of machining, tool attributes and machining attributes.. Rules would also identify when prerequisite operations are required. Handling for collision resolution following the detection of collision in the machining environment is also noted in manufacturing knowledge. The *NC Database* is providing the user with information on available tool components and tool assemblies. This information has two distinct forms. The tools components can be defined to the user in terms of attribute lists. However, tool assemblies will be defined through attribute lists at the assembly level and pointers to component data. Machines, materials and cutting conditions each have their unique form of data that must be handled by the generative user interface. The *CAD/CAM services* examples as chosen for Figure 27 on page 72. provide for the user geometry and features processor access. These services also provide for access to numerical control processor entities as noted in the figure. The reader must consider that these examples are representative elements of a generative part programming system definition.

4.1.1 User Interface Mechanisms

The generative system user interface developed for the framework has the internal elements of *menus, panels, user prompts,* and *messages.* This user access to an implementation of generative NC receives graphical display support from basic CAD/CAM system services.

4.1.1.1 Menu Selections

This mechanism provides the pathways into an implementation that are available to the run-time user. Effort must be placed here to keep these pathways short. A two-level menu structure seems optimal for a generative system. Let us consider a menu selections for pathways to be defined for the *Manufacturing Results Analysis* activity. The main level menu selection "ANALYSIS" provides access to a set of analysis related functions. The second level menu would provide the user a selection to system object information to be analyzed. The "PRINT" function can be implemented as a "command button" on the second-level menu as available within the support routines for graphical user interfaces

1. ANALYSIS + PART MACHINING OPERATIONS
2. ANALYSIS + MACHINING ENVIRONMENT
3. ANALYSIS + MACHINING OPERATION
4. ANALYSIS + MACHINING FEATURE
5. ANALYSIS + MACHINING SEQUENCE
6. ANALYSIS + TOOL LIST + print
7. ANALYSIS + WORKPIECE + print
8. ANALYSIS + ROUTING + print
9. ANALYSIS + ERRORS
10. ANALYSIS + DECISIONS (machining feature/operation/sequence)
11. ANALYSIS + PRECEDENCE (machining feature/operation)

4.1.1.2 User Prompts

This mechanism guides the user through the menu selection chosen. The framework requires a consistent handling of manufacturing terminology in this mechanism. This system initiated guidance directs the user in accessing the related geometry, technology, or system processing.

4.1.1.3 Panels

Panels are the primary mechanism of dialogue with the user for an implementation of the framework. The panel is used to present information and collect data from the user. A part programming environment would be established through a *declarative file* that provides system default values on these panels. Control is established by a *panel generator service* as to which parameters are available for modification by the run-time user.

ANALYSIS + TOOL LIST: The information available to the user for a menu selection of ANALYSIS + TOOL LIST would be presented on a panel. This panel may also include a tool assembly icon.

- Tool Assembly Identifier *DYNA TOOL ASM 1*
- Tool Assembly Set Length *78.0 MM*
- Number of Components *2*
- Tool Type *Endmill 2 Flute*
    Tool Identifier *EH2I002500Y012500L*
- Extender Type
    Extender Identifier
- Holder Type *DYNA*
    Holder Identifier *DYNA #3*

ANALYSIS + ROUTING: The information the user would receive from the ANALYSIS + ROUTING menu selection is presented as follows. In this case, this information would also be available in "PRINT" format.

- Operation Number and Type
- Tool Assembly Identifier
- Spindle Speed
- Machining Feedrate
- Approach Feedrate
- Slowdown Rate
- Finishing Feedrate
- Special Processing Requirements
- Operator Instructions
- Feature Identifier
- Stock Preparation Instructions
- Setup Data and Instructions
- Operation Time

4.1.1.4 System Messages

This mechanism provides the user with immediate feedback in the *optional interactive execution* of the part programming session. User feedback is structured as both a simple message and detailed message. This two-level approach serves both experienced and inexperienced users. Messages provided during *program controlled system execution* of the part programming process are captured on the *error report.*

4.2 Manufacturing Data

This *system component* provides a source of the *detailed manufacturing process data* that the generative NC framework will employ throughout the part programming process. Therefore, this data source is likely to be referred to as *manufacturing process data.* This data is required by the system to establish a broadened manufacturing technological context that is expanded from a precise core of installed functions and manufacturing knowledge. This point is established in Figure 28. This figure displays an expansion of both system functions and manufacturing knowledge by *manufacturing process data.* This expansion addresses a problem from the literature that indicates the amount of data and knowledge required by a generative system was an inhibitor of early efforts [8,16,66].

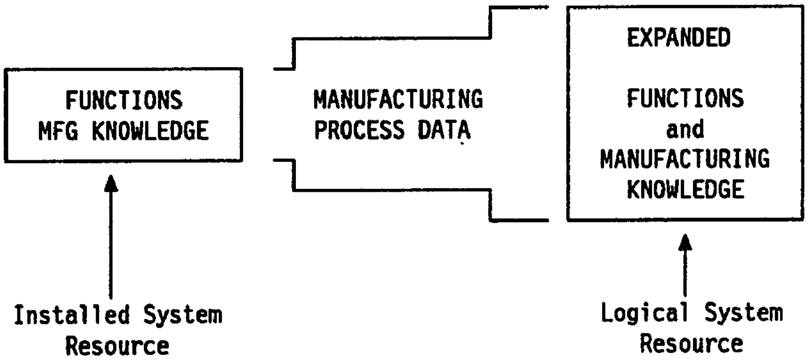

Figure 28. Manufacturing technology expansion through manufacturing data.

This manufacturing data is defined to be a resource to the framework and is primarily found in the nature of machining processing related parameters. These parameters can be an extraction of the characteristic elements of an NC part programming function. The data is available in *single value* and *multiple values* formats. A multiple values parameter can perhaps be exemplified as being dependent on machining material types and conditions.

4.2.1 Manufacturing Data from NC Functions

An example function calculates the machining operation depth for a tool flute length requirement. This calculation is performed with machining approach, feature depth, breakthrough amount, and tool tip as shown in Figure 29. Feature depth is read from the machining feature. Tool tip is calculated from the *manufacturing tool database*. However, the parameters *machining approach* and *breakthrough amount* [27] are extracted from the design of the function and are specified in the system as *manufacturing data.*

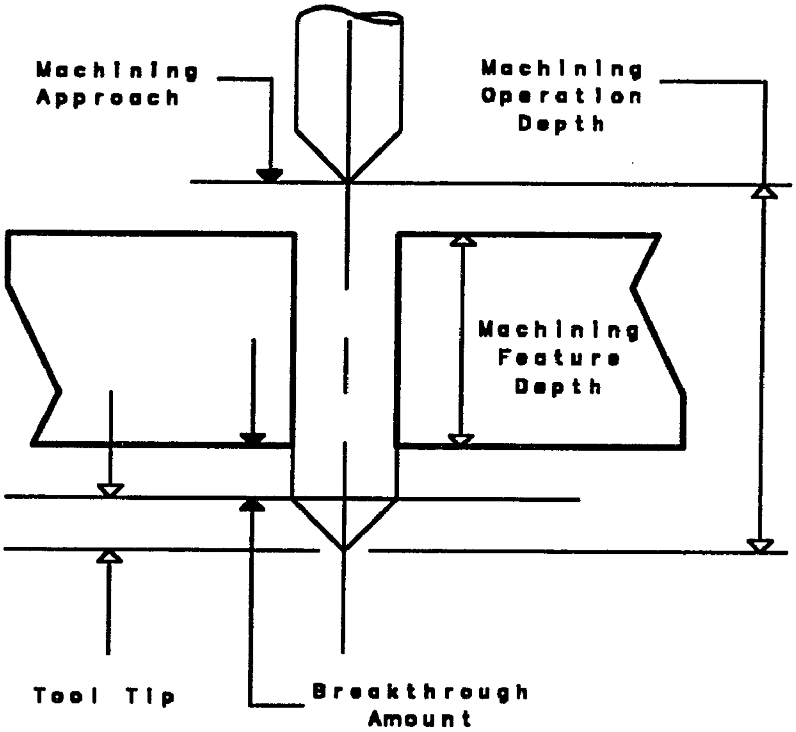

Figure 29. Machining operation overall depth with manufacturing data.

---

[27] Machining Approach is the distance in the tool approach direction above a machining feature that the tool should prepare to engage material. Breakthrough Amount is the distance in the tool approach direction below a machining feature that effective diameter of the tool should sweep.

4.2.2 Manufacturing Data from Manufacturing Knowledge.

Another type of *manufacturing process data* can originate from *manufacturing knowledge*. This type of data is a "named variable" that is used within the manufacturing knowledge. This data provides the part programmer direct access and control of the technology that will be used to program parts. This control is accomplished through a simple change in a manufacturing data parameter as opposed to a rule base change and compilation.

A rule is being defined for holemaking operations that will make a changeover from conventional trajectory processing to deep hole trajectory processing. In this case, the *manufacturing data* parameter "Deep_depth_dia_ratio [28]" is defined to expand the manufacturing technology that can be applied to a part program that fires this rule. An IF-THEN formulation for a rule may take the form:

```
IF feature depth/feature_diameter > DEEP_DEPTH_DIA_RATIO
THEN constrain possible operations = deep_hole
ELSE constrain operations = hole_making
```

The *manufacturing data* system component will be assembled in the framework with major contributions. This component establishes the promise for simplified customization of an implementation of generative NC. This customization is accomplished without the need to re-write functions or knowledge base rules. A simple edit of *manufacturing data* can provide a completely different result from a system execution. This edit has provided a different configuration of manufacturing know-how to the system.

---

[28] Deep_depth_dia_ratio is the ratio between the depth of the hole machining feature and its diameter.

4.3 Manufacturing Knowledge

This *system component* is defined to store knowledge as rules for use within the framework functions. This manufacturing know-how is introduced into the core of the framework as applying specifically to the *machining process generation* and *machining sequence generation* domains. Each of these processes will require a unique set of supporting knowledge. The author proposes that an implementation of manufacturing knowledge will maintain this discrete handling of knowledge bases. Figure 30 on page 81 provides an example content statement of manufacturing knowledge for an implementation of *machining process generation* to support milling. This figure has been generated as an overview of the rule definitions for an industrial implementation of the manufacturing knowledge.

Siores [66] confirms the relevance of a manufacturing knowledge component to advanced manufacturing systems. However, his approach to the overall problem of automated processing places the knowledge base as the system core. Siores makes no distinction between knowledge and database in his approach. Tisza [71] also places the inference engine as the core of sheet metal planning solution. In contrast, the *manufacturing knowledge* system component defined in this research is positioned to provide a specific resource to the system. Ghosh [19] uses a similar approach for his specification of decision tables that support his two-dimensional cutting system. These knowledge supports his selection of machining operations, sequencing, and machining data. This research has defined discrete handling for *manufacturing knowledge*, *manufacturing data*, and *manufacturing database*. These unique components are each providing specific resource to the framework. This segmentation of components is proposed as a means of offering a hierarchy of customization to the user. This hierarchy is designed to structure the generative system customization task for human capabilities. This segmented resource structure will also support the efficient validation of expected results for many of the tasks chosen for manufacturing technology customization.

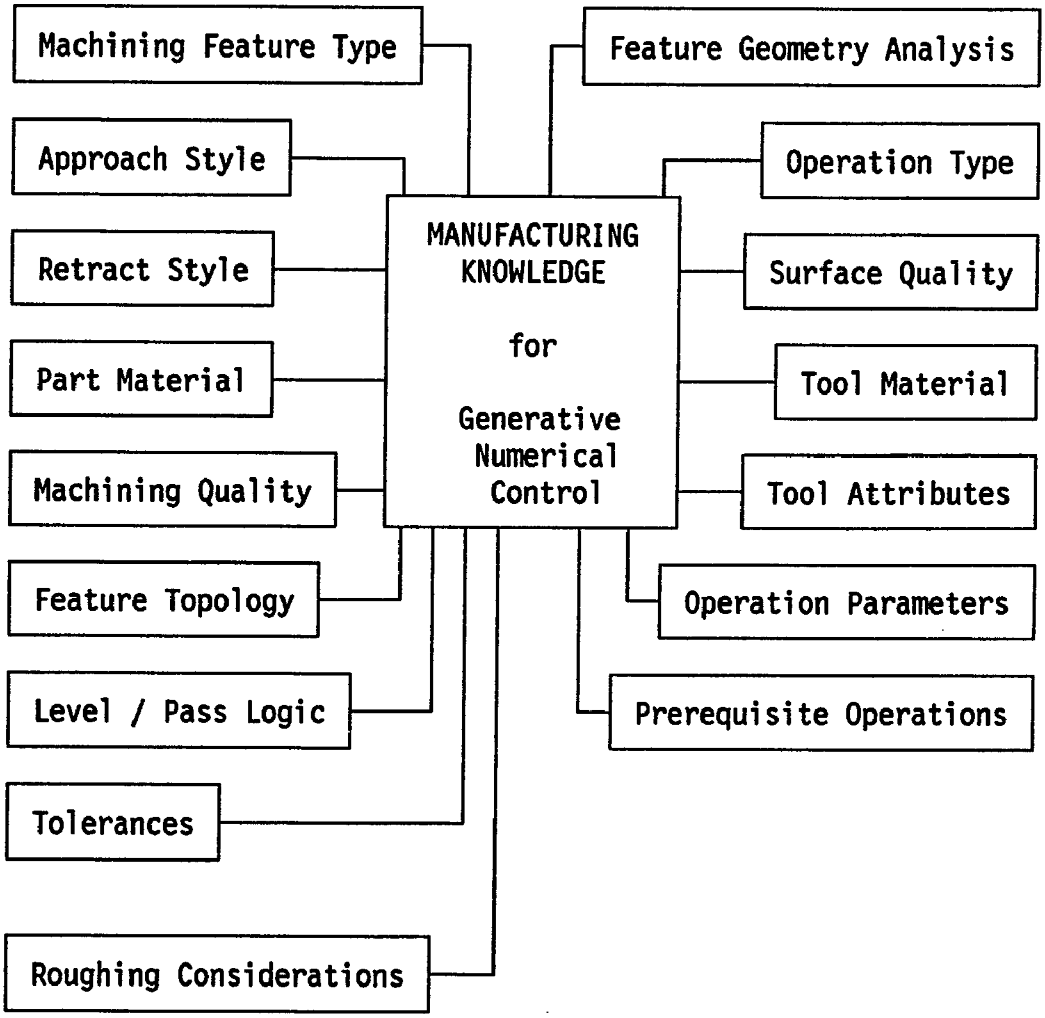

Figure 30. Milling rules content for machining process generation.

Manufacturing knowledge provides greatest utility when it is structured [29]. This structure promotes the representation of know-how as knowledge that can be:

- understood and known by the system user,
- available for user customization,
- accessible by the system programs,
- validated with the user's anticipated results,
- maintained timely and efficiently, and
- enables incremental system growth.

Knowledge is structured as rules and rule sets. The *manufacturing rule* is a sequence of tests and actions that provide structure to machining knowledge. The rule level is reserved for first principles as a starting point for a generative solution. An example of this first principle data can be "set tool diameter for simple drilling." This rule would provide actions to set a minimum value, preferred value, and maximum value of tool diameter. The actions specified within a manufacturing rule can create both additional facts and direct system processing by routines within the CAD/CAM services. The *manufacturing rule set* is a grouping of rules to determine a specific elemental process direction or parameter. The rule set would organize a collection of all knowledge that is implemented to be applied toward that elemental processing direction or parameter valuation.

---

[29] Personal communications and observations of the author through industry specific contacts, Boeing, Chrysler Motors Corporation, Sikorsky Aircraft, Audi, Damler-Benz, Hella, Rockwell Graphics, Pratt and Whitney, Volkswagen AG, Fokker Aircraft. 1989-1995.

In the previous example, the rule of tool diameter for simple drilling would be combined with rules for other operation processing types. The typical holemaking related tool diameter rule set would include knowledge to support a simple drilling, boring, reaming, spot drilling, center drilling, and continue with rule entries for all implemented operation processing types.

4.3.1.1 Manufacturing Knowledge and CAD/CAM Services

This *manufacturing knowledge* system component is defined here to exist within the targeted CAD/CAM system domain. The reasoning supporting this proposal is that a fully integrated rules processor and rules base for industrial production must consider the unique methods and practices of the implementing company. As such, object descriptions will be highly unique in both the CAD/CAM system and the knowledge base system. Full integration will allow these unique object descriptions to be available to both the procedural and knowledge base components of the operational planning and part program generation solutions. The proposal made here is predicated on the CAD/CAM system supporting robust manufacturing object definitions. This proposal is clearly justified as a pragmatic requirement for production systems upon consideration of the *consistency management requirements* for hundreds of manufacturing objects between multiple systems. Lawler [42] confirms the desirability of this level of integration based on his experiences with the ICAD system "linked to" multiple CAD/CAM systems. However, this author's proposal is made in contrast to the implicit separate placement of *manufacturing knowledge* and CAD/CAM services in other documented work [32,36,59]. This contrast can be readily understood upon consideration of the differences between a production system definition and laboratory systems formulation. Gu and Zhang [23] bridge both these perspectives with their consideration that the CAD/CAM services will not hold a robust object definition schema. Therefore, their work proposes a system component external to both *manufacturing knowledge* and CAD/CAM Services for object-oriented data management.

4.4 Numerical Control Database

The *NC Database* provides the structure for machining environment data that supports a generative NC system. This *system component* is defined to store the detailed data that describes the physical and logical environment for the objects related to manufacturing technology. This system component will also be referred to as the *manufacturing database*. This database is defined as a basic entity, attribute, relationship model.

- Entities
    - Catalog - (Endmill_2_flute_HSS)
    - Objects - (Endmill, 2 flute, HSS, 12 mm, ...)
- Attributes

    An Attribute is an elementary fact attached to an object. These attributes are used to define both the geometry and technology of the object as shown in Table 1.

Table 1. Abridged Example of Endmill Tool Attributes.

Attribute	Value
Number of Flutes	2
Tooth Material	HSS
Diameter	12 mm
Flute Length	32 mm
Corner Radius	0
Tooth Rake Angle	7
Rotation	Right Hand

- Relationships

    A relation is a link which connects 2 entities which can be catalogs or objects. Examples: Tool-to-Machine, Tool-to-Material

The objects specified by this research to be available from the manufacturing database:

- Machine
- Tool Components and Assembly Support
  - tool
  - extender
  - tool holder
  - shank/pocket coupling
  - tool assembly
  - inserts and body
- Stock Standard Form Definitions
- Material Type and Condition
- Machinability Data

This *NC Database* system component is comprised of only object definitions, attribute values, and application assertions or relationships. There is no requirement nor provision defined for geometric model storage with an object in the database. These objects are precisely defined and manipulated through an application program interface that isolates the framework from the data definition and storage techniques.

The structure and the content of this database is proposed to be open so that it can be implemented to serve the need of a set of users. One such need is the requirement to emulate current techniques used for the organization of the cutting tools and tool assemblies. On this topic, the manufacturing database holds information as to physical and logical equivalent of the tool crib typically found for larger machining shops. One shop indicates that there would be 20,000 tools required for current and production ready process support.

Lawler [42] identifies a "resource model" as a component in his GCAPP process planning approach. This GCAPP component contains both data and knowledge on machines, tools, fixtures, and transfer equipment. In contrast, the NC Database defined here as a component for generative numerical control is absolutely reserved for machining environment data. The author proposes that this configuration of the manufacturing database contributes to the modular concepts developed and maintained for the framework. Component modularity will be a key point of this research. This becomes evident when approaching the problem of assembling these components. This modularity is also critical to developing the basis of a run-time and build-time that can be managed within a part programming user domain.

The example of an abridged 3-axis machine database entry for use within the framework is presented here. Attributes typically used by the NC post processor [30] for final data validation are considered in the framework for part program generation.

- Machine Axis Limits
  - X, Y, Z Axis Minimum and Maximum
- Rotary Table
  - Rotary Axis
  - Rotary Motion Type
- Spindle
  - RPM Minimum and Maximum
  - Discrete or Continuous
  - Spindle Key Lock
- Motion Control
  - 2D Linear / 3D Linear
  - 2D Circular / 3D Circular

4.5 CAD/CAM System Services

This *system component* is defined to identify critical elements to support generative NC part programming. This component is an integral part of the framework to define a production machining part programming solution. The elements identified as CAD/CAM services are:

- Geometric / Features Processor,
- Technology Based Machining Operations,
- Numerical Control Data Structure, and
- Numerical Control Utilities.

4.5.1 Geometry / Features Processor

This component is defined in the framework as the source of all modeling related functions. Model files are built in terms of part, stock, fixture, and workpiece. Likewise, the objects defined to support decision making internal to the planning activities of the manufacturing processes are also modeled using this system component. As an example, a tool is modeled from the parameters defined in the *NC database* and placed at the critical points of the toolpaths through the use of this solid based modeler. Collision checking is then undertaken to identify potential gouging of the part or unexpected workpiece engagement. The part model is the description of the entity that must be manufactured. It contains all the specification of dimensions, technological attributes, and tolerances. This input model to the framework is constructed with feature primitives. These primitives provide a parametric building block that holds technological data along with the traditional geometric data. Each feature instance found in this model is a candidate for machining within the framework for generative NC.

The framework will be most productive being implemented to work within a set of feature primitives that have a machining semantic. This research has found that the definition of set of discrete machining features can be accomplished working from existing research efforts such as MetCAPP [28], CAM-I [5], and PDES [67] and company research feature definitions. The machining features identified for an implementation should be evaluated within the constraints of the targeted NC processor. This will identify any adaptations necessary for machining features to map to the basic forms producible by the NC processor and current NC machining equipment. This observation is validated by Case [6].

4.5.1.1 Machining Feature List

The following list of machining features has been formulated from the research for the general milling and holemaking in prismatic parts. This feature list includes multiple class definitions for several feature structures to support a more optimized processing flow within the framework. These multiple class definitions are also supported in the literature CAM-I [5]. The features processor is isolated to promote evolution to an emerging standard PDES [67] as suitable for a leading-edge manufacturing solution.

- Hole
- Counterbore
- Countersink
- Internal Thread
- Complex Cutout, Rectangular Cutout
- Complex Pocket, Rectangular Pocket
- Complex Slab, Rectangular Slab
- Blind Slot, Through Slot
- Complex Step, Rectangular Step
- Open Profile, Closed Profile

4.5.2 Technology Based Machining Operations

A *machining operation* is a well defined and parameterized machining procedure. This procedure [30] specifies control for both the tool trajectory and the technology of machining. This procedure typically refers to machining trajectories with one tool. The machining operation is designed to be applied within a part program for the complete processing of a qualified material removal requirement. The technology based machining operations would be comprised of elements that introduce the "process level" view to each material removal requirement. This process level approach includes the elements that specify:

- machining technology based trajectory and toolpath,
- parameterized numerical control geometry operators,
- multiple level, multiple pass, one-pass, and spring pass,
- machining start point and end point definition,
- cycle trajectory syntax and parameter evaluation, and
- machining approach and retract path definition.

This *CAD/CAM services element* is defined for use in the framework of generative NC due to three principle factors established within this research. These factors are that the *technology based machining operations* support feature based machining, are oriented to process level trajectories with technology, and automatically manipulate toolpath geometries. In contrast to many numerical control processors, toolpaths are not defined about a part by the user creating specific NC geometry and then following this geometry with a graphics cursor.

---

[30] Aber, D.M., "NC Application Platform: Technological Requirement Perspective," Proceedings: CATIA Operators Exchange (COE), 1993, p 22.

- **Feature Based Machining** is supported by technology based machining operations that define basic producible forms that map well to the geometry definition available from material removal processes.

- **Process Trajectory with Technology** is supported by the technology based machining operations. These operations offers a complete description that provides for unambiguous toolpath computation with the associated machining technology in a unique package.

- **Toolpath Geometry Manipulation** is managed by the technology based machining operation. These operations internally create and control in-process toolpath geometry with operation parameters. This minimizes external manipulation of the low-level path elements to achieve the required toolpath output. Reference can be made to Figure 4 on page 13.

The following technology based machining operations are proposed as representative of the general holemaking and milling machining problem. Each of these selections would specify unique machining process related trajectory control and technology parameters.

- Boring
- Boring w/Spindle stop
- Countersinking
- Counterboring
- Drilling
- Reaming
- Drilling w/Peck, Dwell
- Spot Drilling
- Tapping
- Facing
- Pocketing
- Area Clear
- Contouring
- Slot Machining
- Profiling
- Surface Machining
- Milling Pilot Holes

4.5.3 Numerical Control Data Structure

This *element* provides the complete description of the part from the manufacturing point of view. It contains all data references and instructions to create a part program to machine the workpiece. This is accomplished by structuring objects that express the complete manufacturing process elements. This approach supports definition of machining instructions at both the technological (machining feature) and the geometric (toolpath element) levels. This structure is supported by object definitions that identify the relationships between the manufacturing elements. This model is managed through an NC data structure application program interface.

- Contents of the Data Structure
  - Components - Manufacturing entities
  - Relationships - Application assertions [31]
- CAD/CAM System Data Structure
  - Application Platform Objects
  - Shared Object Definitions
- Application Program Interface Support
  - Read entity
  - Write entity
  - Modify entity
  - Delete entity
- Data Structure Consistency Management
  - Data Validity Management

---

[31] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, pp 78-111.

The following objects are the NC entities managed by the new numerical control data structure proposed by the CATIA ® Manufacturing Infrastructure [12]. This leading edge manufacturing industrial product maps well to the requirement for numerical control data structure proposed by this research [32]. This is due to an intensive collaborative effort with this author based on the early formulation of this research. The theme of this effort was to build a piece of software that would provide a technological platform for next generation numerical control products. The technology breakthrough sought was an extensive set of manufacturing objects and shared services on those objects as callable routines made available to product developers. The definition of these objects and services would be based on manufacturing technology implemented with the latest computer programming techniques. Figure 31 on page 93 presents the topology of the NC data structure.

- **Part Operation** links all the *machining processes* necessary for the complete machining of the *machining view* based on a unique part registration on a machine. The *part operation* links these processes with the associated *stock, fixture, material*, and *setup* entities.

- **Machining Process** is an ordered list of *machining operations*. It is the result of a machining strategy applied to a *machining feature*.

- **Machining Operation** is a predefined and parameterized machining method applied to a *machining detail*. The *machining operation* is linked with a *tool, tool requirement*, and *tool assembly*.

---

[32] Aber, D.M., "NC Application Platform: Technological Requirement Perspective," Proceedings: CATIA Operators Exchange (COE), 1993.

® CATIA is the registered trademark of Dassault Systemes.

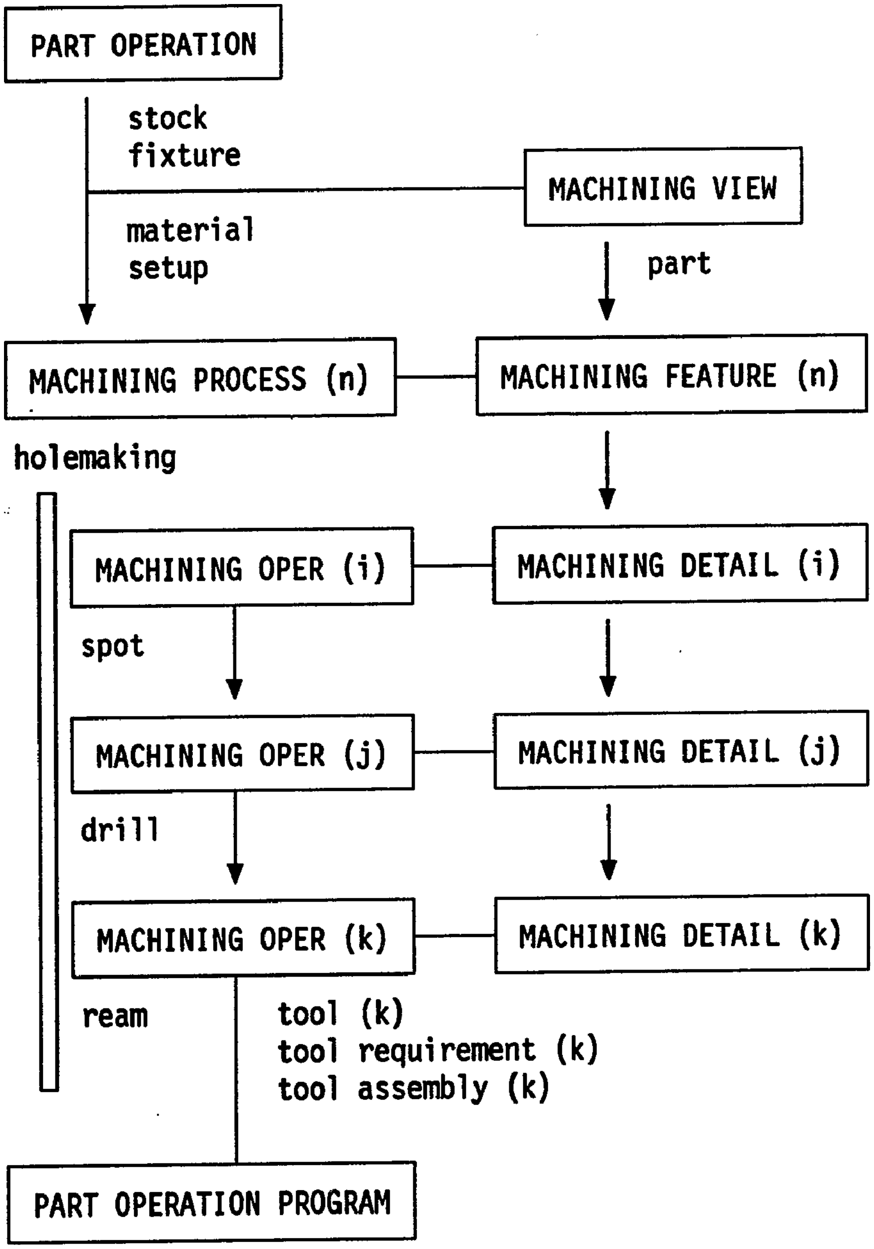

Figure 31. Topology of the numerical control data structure.

- **Machining View** is a set of *machining features* to the machined within the *part operation.*

- **Machining Feature** is a *feature instance* composed of a material removal volume, tolerances, machining approach vector, and other technological attributes.

- **Machining Detail** is a specification of the *machining operation's* material removal volume and machining approach vector.

- **Part Operation Program** is the sequenced order of NC commands that define the total processing for the *part operation.* The *part operation program* contains the above entities with *program start point, transition paths, auxiliary commands* and *program end point.*

The NC Data Structure element is defined for use in the framework for generative NC due to two principle factors.

Robust numerical control processor - the numerical control data structure has the capacity to store and access a comprehensive set of manufacturing data at the process level.

Reduced data structure - the numerical control data structure will place the responsibility for this component at the CAD/CAM services level. This results in a generative numerical control implementation being free of the data management task yet remaining data consistent with other numerical control applications.

4.5.4 Numerical Control Utilities

This *element* provides a set of standard services for selected numerical control functions. These services are manufacturing processes unto themselves dedicated at the task level. Each service has a clearly defined input stream, processing flow, and output. These services manipulate objects defined by the *NC Data structure*. The *numerical control services* that would contribute to the framework for generative NC are defined as:

- Cutter Selection,
- Cutter Assembly Management,
- Toolpath Computation, and
- Machinability Data Identification.

4.5.4.1 Cutter Selection

This service processes a tool requirement based on attribute constraints. The attribute values may be range sensitive and will return one selected cutter and its actual attribute values.

4.5.4.2 Cutter Assembly Selection

This service handles the task of selecting tool assemblies for a manufacturing application. The required tool assembly may be constrained by component physical connectivity and toolpath collision avoidance data.

4.5.4.3 Toolpaths Computation

This service provides access to the toolpaths generated for a machining operation. The math unit of the numerical control processor is called to return both toolpaths and machining operation area data. This data is used to build machining volumes.

4.5.4.4 Machinability Data Identification

This service identifies, selects or calculates the machining technology components of the machining operation with respect to cutting conditions. This data would be required in the framework to provide visibility to the machining operation for depth-of-cut, machining speed, and machining feed rate recommendations.

The Numerical Control Utilities component is defined for use in the framework of generative numerical control due to two principle factors established within this research.

Callable Manufacturing Processes - the numerical control services provide functions to the framework. This type of functions are repeatedly called with a precise set of data to return a standard output.

Sub-Process Consistency - the numerical control services also provide a consistency of handling for identical manufacturing processes occurring in the framework in numerous activities. The services packaging allows evolution in the handling without changing input or output as defined to the framework.

5.0 Development of the Architecture

The *architecture* for generative numerical control is approached by this research as a logical and progressive development. The architecture's foundation is in applying the technology of state-of-the-art solid modeling software within the context of operational planning. The architecture development is guided by the "module development process" as proposed by Lehtihet and Aber [43]. This process is a means of realizing the opportunity of geometric modeling to computer-aided manufacturing problems as proposed by Wang [73]. This architecture is directed toward a solution to the programming of NC machine tools in the domain of automatic operational planning. Chan and Voelcker [7] discuss the "ill-defined" interface with this domain and manufacturing planning. Therefore, the first step to building an architecture in the framework for generative NC is to develop the *system level inputs and outputs* as structured in Figure 32.

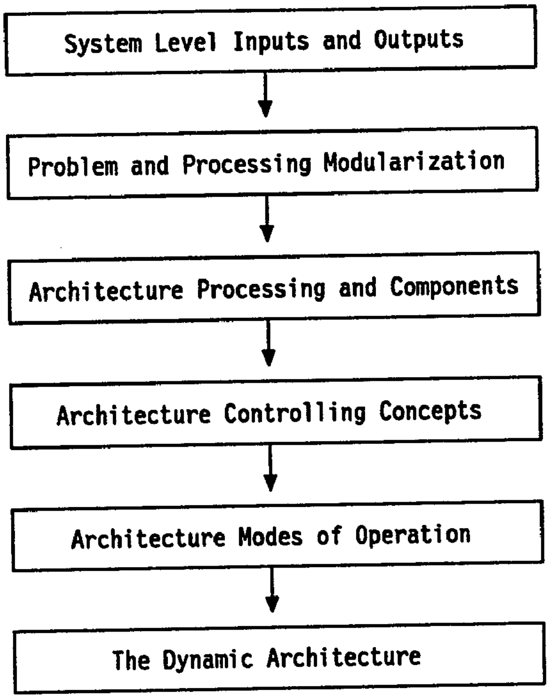

Figure 32. The system architecture progressive development.

5.1 System Level Inputs and Outputs

The inputs and outputs defined at the generative NC system level proposed as resulting from this research, are formulated in Figure 33. These definitions identify the data flow into and out of a generative system. The development perspective chosen establishes the absolute data requirement for an automated industrial production part programming solution. This system level view is represented in an input-processing-output (IPO) format.

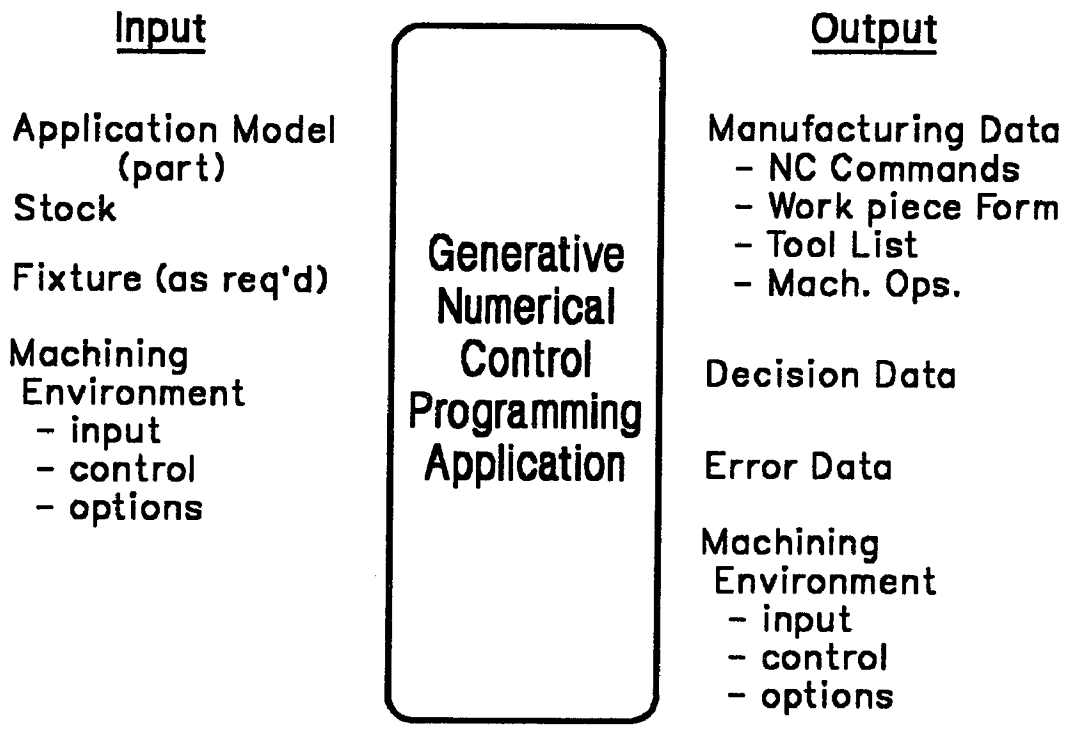

Figure 33. Application level inputs and outputs.

5.1.1 System Inputs Development

Application Model: The *application model* is a solid based model that has been prepared with feature descriptions that are available in a feature based design and manufacturing application. This model originates in engineering design and includes the manufacturing addendum necessary to identify specific part manufacturing requirements. This addendum may be necessary to allow for additional material or tooling requirements to fixture the workpiece for manufacturing processing. The "feature-based design" system development philosophy that enables a strong contribution to computer integrated manufacturing goals, must be tuned here as realized from the current production part programming environment [18,27]. Provision is made for a solid model that is not created and populated with feature primitives as defined in CAM-I [5], IAMS [28], or PDES [67]. This solid form of a part model will consume additional pre-processing to formalize the machining features. Feature recognition is not examined in this research. The reader can consult Houten [27], Joshi and Chang [35], Matsuda and Kimura [52]. or Vandenbrande and Requicha [72] for treatment of this problem.

Stock Model: This model represents the initial geometric state of the workpiece prior to any machining operations. The model is made available to the application in solid form. The source of this input data may be an explicit stock design or reference to a near-net-shape model. This stock may alternatively be selected from a *manufacturing database* standard stock preparations or dynamically sized from the "minimum part envelope." This object can also be identified as an system output model or resultant *workpiece form* from prior manufacturing operations. The stock entity takes a solid form and may include both regular (parametric) and complex primitives.

Fixture Model: This model represents the fixture [33] created for a specific manufacturing purpose. This *system input* is optional. The model is proposed to be in solid form. This *fixture* data defines an object of the architecture that is treated by the framework's *system components* and *functions.*

The fixture object is used as a spatial constraint in the generation of machining operations and path elements determination. The data is used to identify the fixture object as an "avoidance region" that must not interfere with the part's processing requirements. Guyder [24] introduces the "avoidance region" and "containment region" terminology in her work on path planning for machining cavities. This fixture data represents the information necessary to construct hardware that secures, supports, and registers the workpiece stock for NC machining. The fixture model may include the complete fixture design or only elements that directly engage the workpiece. The model content, as available, will be used for *collision detection with analysis* throughout the architecture processing flow. This continuous approach to quality is used to generate and validate part program content. The creation of this object model is not undertaken by this framework. In the literature, Houten [27] defines a module reserved for this task. Pande and Prabhu [58] also make reference to a fixture creation function. This research considers fixture creation as structured in "Recommendations for Further Study."

Machining Environment: This *system input* completes the data required to execute an implementation of the framework. The data includes the primary geometric and technology forms as previously developed as *part, stock,* and *fixture.* Machining environment also includes *user inputs, processing algorithm control,* and *application control options.*

---

[33] A fixture performs the functions of positioning, supporting, and clamping the part in a pre-defined position and orientation and keeping it in that position and orientation under the forces of the manufacturing process. Delbressine, F.L.M. deGroot, R. van der Wolf, A.C.H., "On the Automatic Generation of Set-Ups Given a Feature-Based Design Representation," Annals of the CIRP, Vol 42/1, pp 527-530, 1993.

5.1.2 System Outputs Development

Manufacturing Data: This *system output* contains manufacturing results specific to the current programming session. This data is the primary result of the system execution. This data is categorized as *NC commands, workpiece form, tool list* and *machining operations.*

- **NC Commands** are the *system output* in the format of *CAD/CAM services NC Processor.* This internal format, representing proprietary data structures of the CAD/CAM system, is reserved for numerical control data. The data can generally be extracted as standard APT source or cutter location data using a CAD/CAM numerical control product utility for data conversion. This data will typically require post processing [30,44,64] to translate to the command streams expected by the machine tool controller.

- **Workpiece Form** is the *system output* that represents the workpiece at the conclusion of the particular part programming session. A solid model records the final state of the workpiece. The material removed in each successive machining operation has been subtracted from the initial stock object to create this model.

  This model can be identified as a *stock object model* for input to the "next" system execution for additional processing toward the part's final specification.

- **Tool List** is the *system output* that collects tool assembly specifications made throughout the part program and presents them in *tool list* format. This information contains tool assembly components and the critical tool setting dimensions. Graphical support is available as built from parameters in the *Manufacturing Database.*

- **Machining Operations** are the *system outputs* that represent the machining to be performed in the manufacturing process. Each machining operation is a complete specification that includes trajectory control, manufacturing technology, and tool definition.

Error Data: This *system output* contains information on the error conditions that were determined throughout the system. These processing perturbations are reported here and made available for user review and analysis. "Warning" level information is also an element of this system output.

Decision Data: This *system output* contains the functions developed in addition to those which are normally available through an interactive knowledge based rules processor. Historically, rules processors are implemented as a series of interactive transactions. Each interaction is supported with a contained "why did the transaction return this information?" Automated program interaction with the knowledge base at the part machining requirements level identifies that the trace of rules fired and feedback be held for the user review and analysis at the close of a programming session.

Machining Environment: This *system output* retains the user entries made to establish the part programming session. This output is developed to store and establish the conditions set for system execution. Processing problem diagnosis and corrective action specification is facilitated with this output.

5.2 Problem and Processing Modularization

The general elements of the manufacturing operational planning problem with concentration toward NC part programming are extracted from Chan and Voelcker [7] and Joshi and Chang [36]. The context of these treatments requires refinement of these elements to determine an interrelated and structured view. This problem view must be manipulated for utilization as a guide in the development of an architecture for an automated system.

The operational planning and programming problem for generative numerical control is formulated by the author. Figure 34 presents this *problem formulation.* The problem structure is arranged in contrast to the approaches from literature. A note should be taken that the literature is typically addressing the process planning problem. The differences in the objectives and constraints of the process planning and operational planning problem substantiates the contrast. The broad base of possible descriptive manufacturing terminology was exhaustively reviewed. The result was to assign the nomenclature that best characterizes the basic nature of the individual problem. This nomenclature is chosen to direct the reader and to minimize perception of overlap in problem domains.

This hierarchical representation of the manufacturing problems depicts the relative position of each problem domain with respect to the others. The linear formulation of the problem hierarchy should not distract the reader from the complexity of each individual problem as undertaken for automatic numerical control part program generation. The conformity of problem flow will be maintained through a rigorous structure of problem-to-problem data interface requirements.

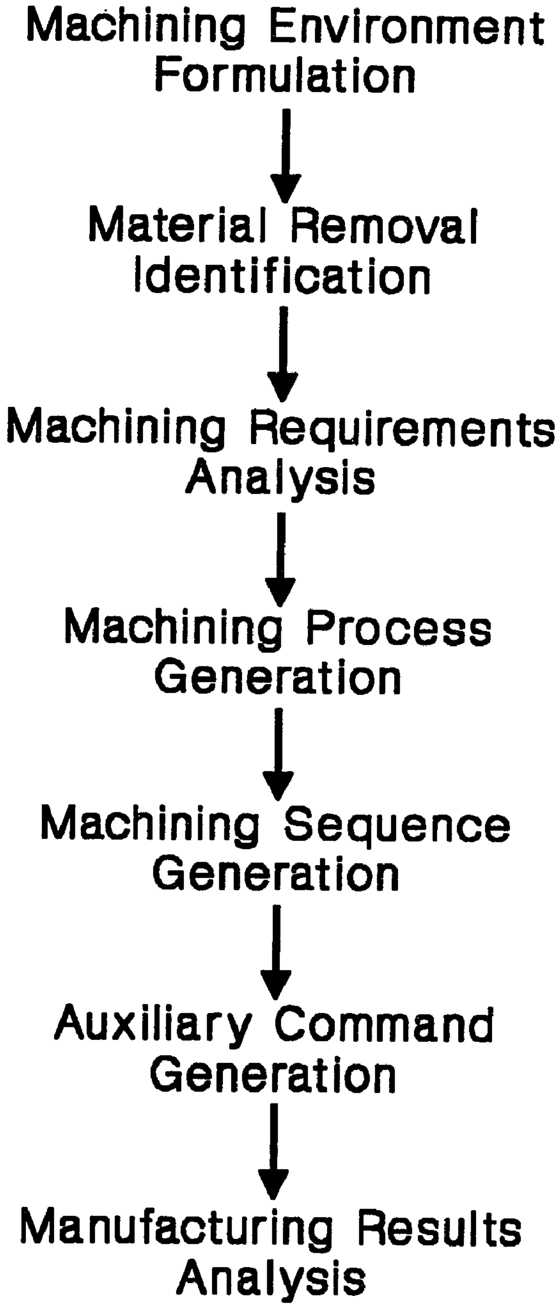

Figure 34. Numerical control problem statement formulation.

5.2.1.1 Current Part Programming Scenario

In the current state-of-the-art situation, part programming would be approached by the following scenario. The part model will be available from engineering design. This model will be converted to NC geometry. It is not uncommon that a re-creation of the engineering model is required to accomplish this task. A similar part program is typically extracted from the NC engineer's file to serve as a guide and assist with machinability data specification. This data was previously used with some degree of success with the machining equipment identified. The part model is now reviewed to determine the tools that will be necessary. These tools are mapped to the areas of the part where machining will be required on the stock. This machining plan is written to create the final part. The part programmer now begins the task of pushing the graphics cursor around the part geometry. This task is generally interrupted by geometry construction for multiple drive and check elements. These toolpaths are created at each machining level in programmed sequence. Instructions for the machine tool are entered where they will be converted in the part program. The resultant part program is now available for verification procedures or tape tryout techniques.

5.2.1.2 New Part Programming Paradigm

The *problem formulation* proposes a different paradigm for advanced numerical control [39]. Each individual problem that the NC engineer would address is included in this paradigm shift. The part programming scenario is now structured within the overall processing context. The result is a formulation of the numerical control problem that is better suited for application of the technologies imbedded in a generative NC system solution.

5.2.2 Process Modularization for the New Paradigm

The numerical control problem statement is used to build a solution that can deliver the new part programming paradigm to the industrial production NC part programmer. The problem statement is employed as the basis for the development of generative numerical control processing modularization. The rigor expended in formulating the numerical control problem statement simplifies the nature of the next task. The processing modularization will essentially be the definition of the spine or structure of the generative NC part programming technology.

The author proposes a direct mapping of each individual numerical control problem to generative numerical control processing. These problems were defined from the decomposition of the larger, unstructured task of *machining operational planning*. This mapping will define the modularity of each processing step. These steps adopt the nomenclature proposed for the numerical control problem definition. The linear hierarchy of the problem decomposition will also be mapped to the architecture processing flow. Figure 35 presents the graphical representation of the architecture flow with process modularization. Joshi and Chang [36] contend that a hierarchical structure with four levels is sufficient to solve the problem. This approach seems to add complexity to the module data interfaces.

Development of the architecture processes follows. These definitions for each of the seven *process modules* will be further developed in the next chapter. That treatment identifies the primary *activities* included in each process module. These activities are further developed by the *generative part programming functions* that provide completed system definition of each generative numerical control processing module.

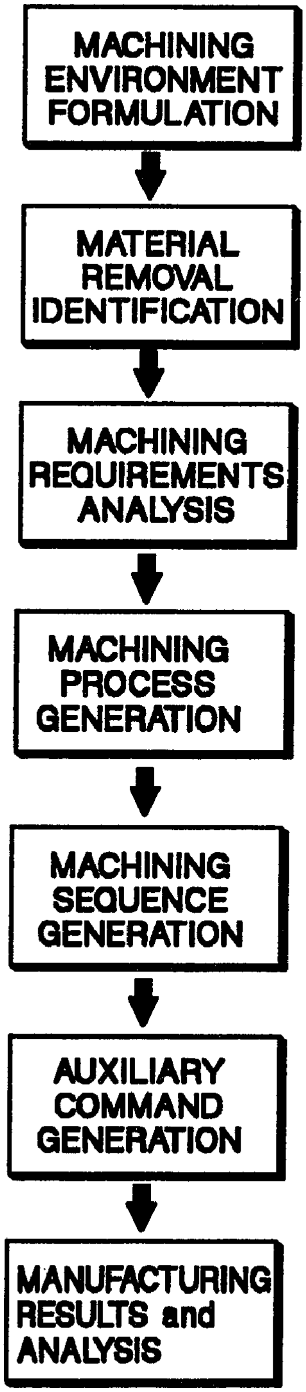

Figure 35. Architecture processing modules.

5.2.3 Machining Environment Formulation

This module collects user input to structure the part programming problem into the machining environment to be used for production. This process provides the function to retrieve the part, stock, fixture, and machine description. This process places the part within the stock and initializes the parameters needed by subsequent processing. Panels are provided for those functions requiring user inputs. This is the starting point for the architecture processing.

The *algorithm control* processing level of the application run will be defined within this module. The *mode of operation* for the system is also set within processing control. Part programming *application control* options are identified at this time by the user.

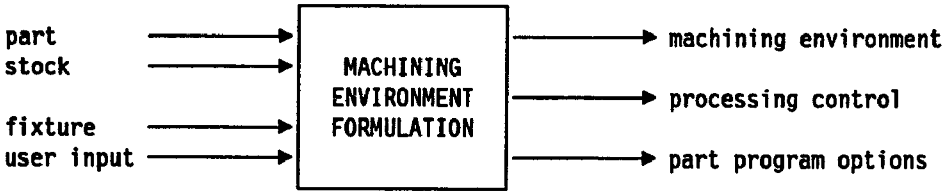

Figure 36. Machining environment formulation.

5.2.4 Material Removal Identification

This module, through a combination of automated and interactive techniques, transforms design geometry and features to *machining features* The process will record user optionally defined *user sequence precedence* at the machining feature level.

The process will identify a single orientation *machining approach vector* for each machining feature. Part-in-Stock analysis defines features outside of the part and within the stock bound. This processing is based in the manufacturing regularity [34] of machining features.

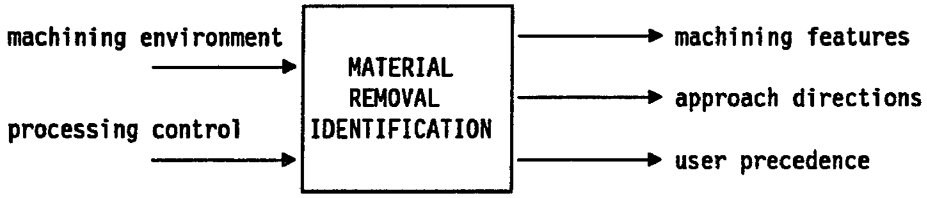

Figure 37. Material removal identification.

5.2.5 Machining Requirements Analysis

This process extracts the geometric constraints on machining and passes them on in the form of *machining requirements*. These constraints will be used in the generative formulation of a machining process for each machining feature. The machining features are sorted to be presented to *machining process generation* in a logical order.

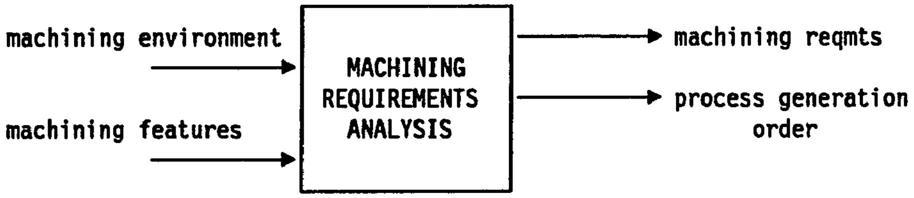

Figure 38. Machining requirements analysis.

---

[34] The regularity of machining features is their inherent promise to be applied to the general machining problem. This promise is founded in the fundamental geometry and technology primarily mapped to the manufacturing characteristics for material removal processing.

5.2.6 Machining Process Generation

This process module generates the *machining process* to be employed based on feature geometry and parameters. The process also selects a *cutting tool* as input to toolpaths generation. Collision analysis verifies the *machining operation* and *cutting tool* pairs specified by the machining process at each machining feature level.

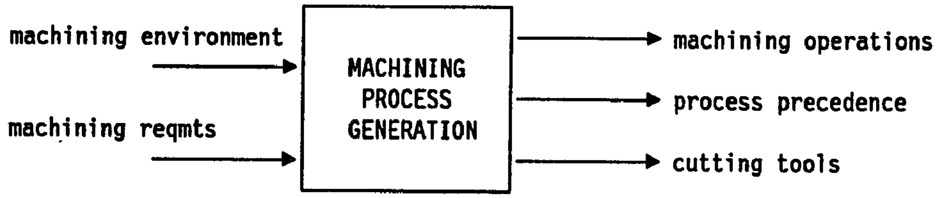

Figure 39. Machining process generation.

5.2.7 Machining Sequence Generation

This process module determines the sequence of machining operations based on *sequencing strategies*. The process observes *precedence relationships* (geometry, machining process, and user preference). The specification for a *tool assembly* will be generated within this module.

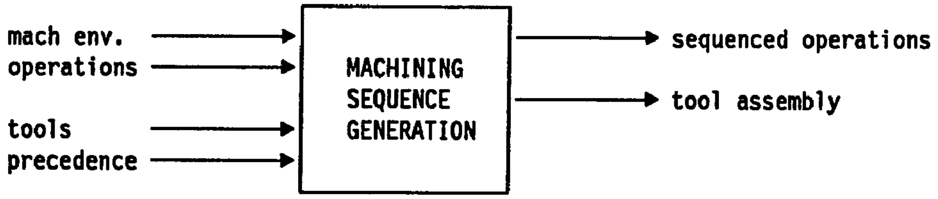

Figure 40. Machining sequence generation.

5.2.8 Auxiliary Command Generation

This process module generates the specification of a *complete part program*. *Transition paths* between machining operations, tool axis changes, and tool changes are generated. These commands are generated in machine and controller *specific post processor syntax*. The generation of these commands is approached with *implicit data* found in the machining operation definition and sequence.

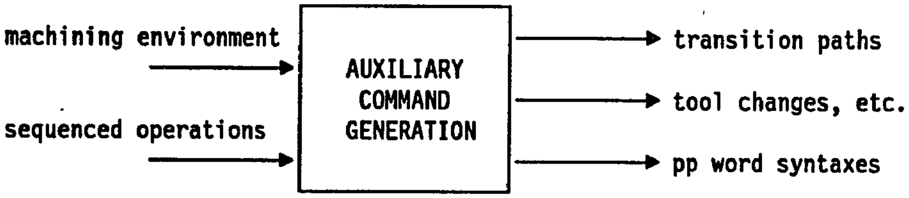

Figure 41. Auxiliary commands generation.

5.2.9 Manufacturing Results and Analysis

This subsystem will provide the capability to access the information generated by an implementation of the framework. This information is the *manufacturing output*, *processing errors*, and *decision support data.*

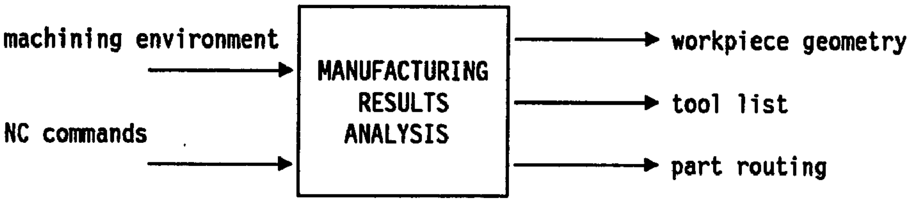

Figure 42. Manufacturing results and analysis.

5.3 Architecture Processes and System Components

The *architecture processes* are further developed to identify their connectivity to the *system components* defined for the framework of generative NC. Each of the seven processes will engage these system components in the manner that provides an optimal architecture for advanced numerical control. This connectivity is a key element in this research. It is also a major contributor to the reduction-to-practice of a technology for industrial production NC part programming solution. System component connectivity to the architecture process modules is approached here in a stepwise manner. The order selected for this development is found below. The *generative part programming functions* are developed in the next chapter

1. CAD/CAM System Services
2. Generative User Interface
3. Manufacturing Data
4. Manufacturing Knowledge
5. Numerical Control Database

The first component selected is *CAD/CAM System Services.* This is a logical starting point as this system component provides the underlying services that enable the building of a generative NC solution. It is meaningful to iterate here that these services are specified from a solids based geometric modeler. This concept is given graphical structure in Figure 43 on page 113. An "interactive part programming solution" for next generation numerical control systems is incorporated in the perspective provided for *CAD/CAM system services* with generative NC. The reader's review of this figure will determine that this solution is built on the same basic platform or *system components.* The relative sizes of the applications are due to the fact that a subset of the *part programming functions* are now interactive user responsibility.

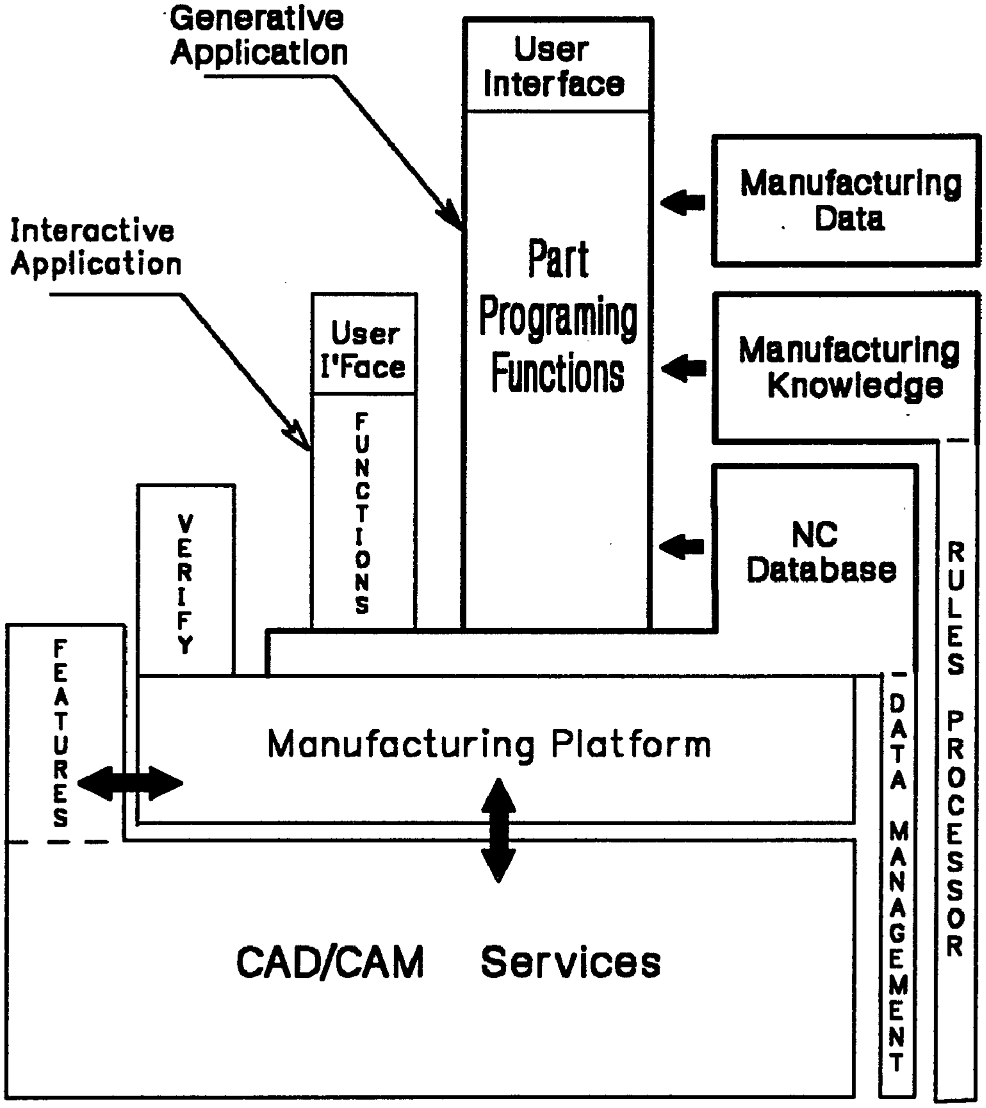

Figure 43. Generative numerical control placement in a CAD/CAM system.

5.3.1 CAD/CAM System Services

This *system component* of generative NC is an integral resource to the *architecture process modules*. The connectivity established for the architecture provides the services of data definition, storage, and utilities to the following modules as identified in Figure 44 on page 115.

- Machining Environment Formulation
- Material Removal Identification
- Machining Process Generation
- Machining Sequence Generation
- Auxiliary Command Generation
- Manufacturing Results and Analysis

These common data definition, storage, and utilities services are expanded as available throughout this advanced NC architecture. The implementation of the architecture would employ native capability within the CAD/CAM System to establish this service level. This constraint on the *CAD/CAM services* system component selection may be overly restrictive based on a review of current leading edge solutions [19,27]. The native data structure of these systems can typically be expanded to meet the requirements on this component through the use of "user data blocks" and "application elements." These elements are managed through the same facilities as the native data structures. Therefore, this component of generative NC establishes data and access consistency for the manufacturing solution through each process module.

Error and *decision data* are also stored within the confines of the CAD/CAM system model. They are designated separately in the graphic for conceptual clarity purposes.

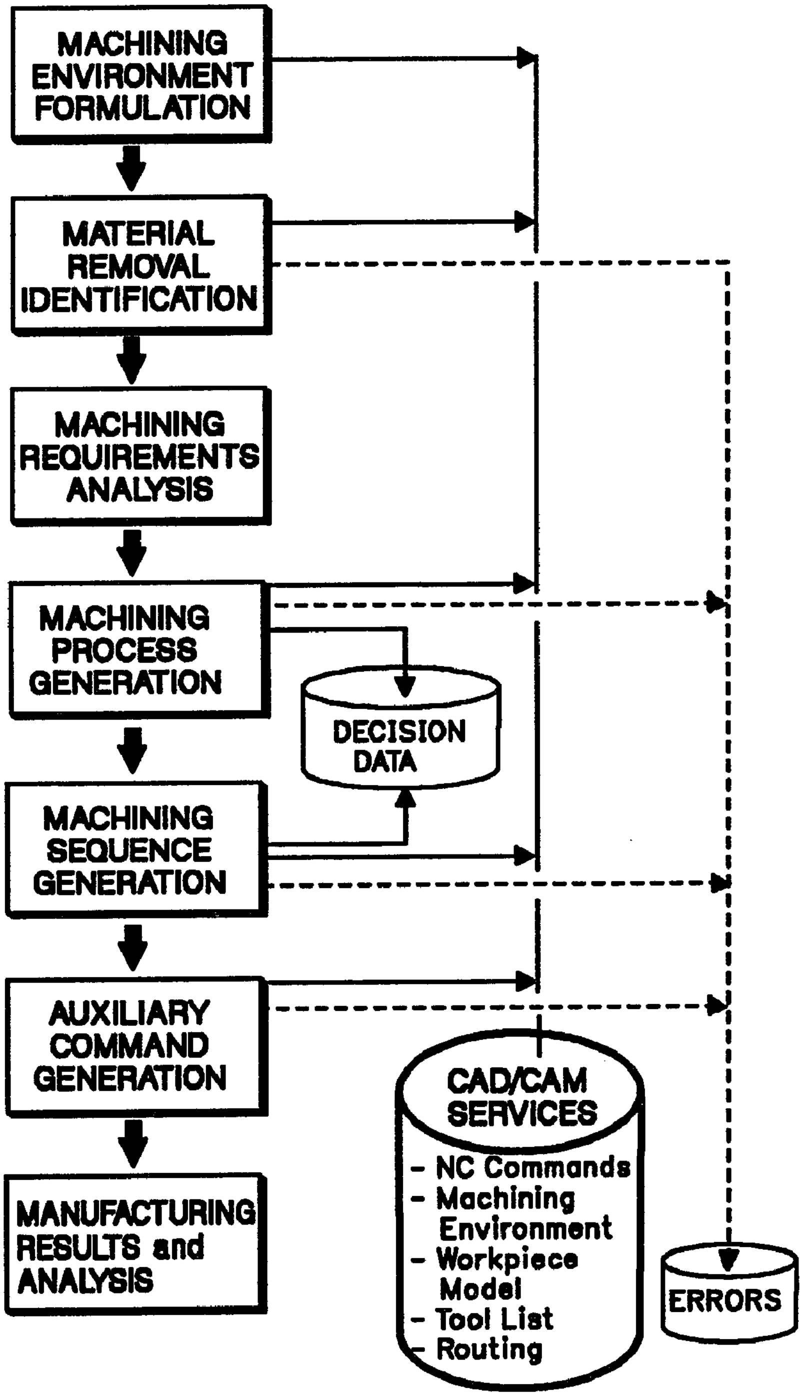

Figure 44. Generative numerical control processing with CAD/CAM System Services.

The following point-of-use examples highlight the resource that *CAD/CAM systems services* provides to the generative numerical control architecture.

Machining Environment Formulation: This process module employs CAD/CAM system services to provide a home for *machining environment data.* The CAD/CAM system data definition as provided will most certainly be expanded specifically for this data element for generative numerical control.

Material Removal Identification: This module uses the feature object and associated structure of the CAD/CAM system to store and access the *machining features* identified in the process. The module employs standard access methods to complete this connectivity with the architecture. These machining features are "feature" objects that differ only in class definition from the design features structure used within the CAD/CAM system.

Machining Process Generation: This module is using the CAD/CAM services NC processor to determine toolpaths from the definition generated for a machining operation. These toolpaths calculations are used with *Machining Process Generation* to verify complete material removal and to anticipate gouge conditions. The geometric modeler provides low-level function to build information from this manufacturing data.

Machining Sequence Generation: This module requires its connection to the CAD/CAM system services to store an ordered set of *sequenced machining operations.* This order is based, in part, on other services of the CAD/CAM system. The *geometric accessibility* of each machining operation has been determined through geometric queries and manipulation. This effort is building the "as machined" solid that provides real-time information to the sequencing process module.

Auxiliary Command Generation: This module connects to the CAD/CAM system services to record the results of *transition path generation* required between two machining operations. This path is stored as elements provided by the *numerical control data structure.* The CAD/CAM system also provides geometric modeling services for the *tool object* that supports the path planning activities to generate these transition paths. Additional auxiliary commands are determined for implied data on the *technology based machining operations* to record a completed set of NC commands.

Manufacturing Results and Analysis: This module accesses CAD/CAM system services to extract for the user the manufacturing process information created by architecture processing. The connectivity to this module in the architecture processing is therefore most comprehensive. The accesses required by this module to CAD/CAM system services are:

- Geometric / Features Processor,
- Technological Machining Operation,
- Numerical Control Data Structure, and
- System Elements for Errors and Decision Data.

5.3.2 Generative User Interface

This system component of generative NC is a valued resource to many of the architecture process modules. The connectivity established for the architecture provides this resource to the following modules as placed in the evolving architecture of Figure 45 on page 119.

- Machining Environment Formulation
- Material Removal Identification
- Machining Process Generation
- Machining Sequence Generation
- Manufacturing Results and Analysis

The *generative user interface* system component of the architecture is placed in these processing modules for the run-time user. The graphic representation of this placement uses size of the user interface icon to identify primary and secondary placements. Secondary placements would reserve "optional" user interfaces preserving a fully automated path through the system. This placement is developed as a structured access by the architecture to facilitate:

- Collection of User Supplied Data and Know-how,
- Presentation of System Information / Results, and
- Modification of System Results at Point of Creation.

Each user interface is identified and enabled by *panels, menus, prompts,* and messages. The utilization of a generalized *panel development facility* promotes consistency and provides a common start point for user panel tailoring.

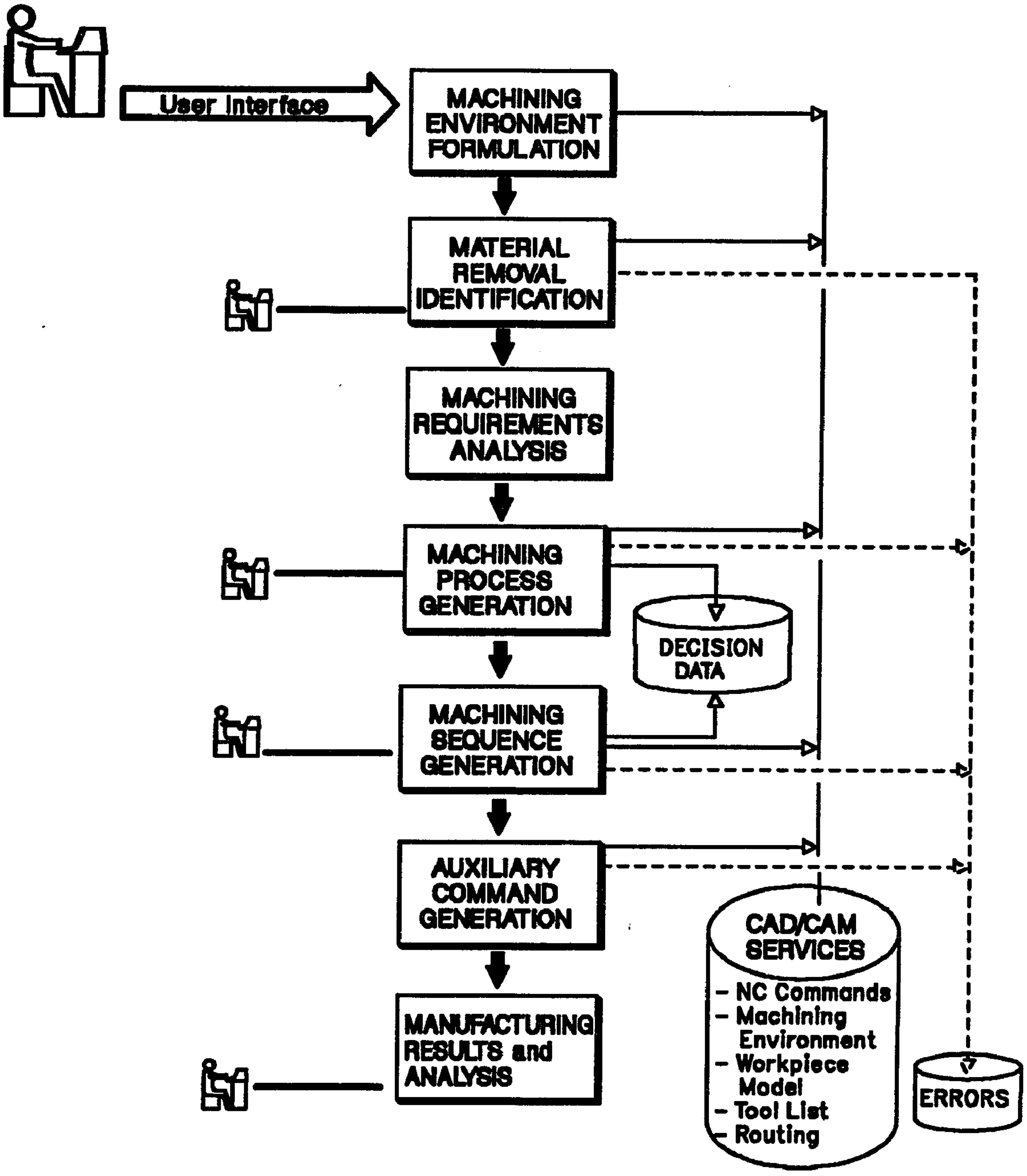

Figure 45. Generative numerical control processing with User Interface.

Run-Time User: This user has the task to employ the architecture developed here for generative NC for manufacturing operational planning. This task will result in a completed set of NC processor commands. The terms applied to this user vary with organizational structure and overall user responsibilities. Typically, the *generative user interface* will be used by a NC part programmer, engineer, or toolmaker. Their use of the user interface remains consistent for entry, review, and edit purposes. However, the subject of their use varies greatly. The engineer may be planning the NC fabrication for a high-value material that is requiring weeks of machining time. This shop processing can be run on a multiple head milling machine producing three parts simultaneously. On the other hand, the toolmaker may be using tool steel to fabricate a clamp that will secure a part feature. This processing requires a milling and slotting operation. The *generative user interface* system component is connected with the processing modules in the architecture to provide consistent treatment to the concerns of both these hypothetical users. This connectivity for user interface concentrates on user entry, review, and editing in a part programming automated system.

The following points of availability highlight the resource that the *Generative User Interface* provides to the generative NC architecture processing modules.

Machining Environment Formulation: This module is a primary use of the *generative user interface* system component of the framework. This location in the architecture is the only required user interface. Consideration must be maintained at this point for a *declarative file* that emulates this primary user interface for execution of the system by a utility or shellscript.

Material Removal Identification: This module provides a user interface to assist the system in the qualification of machining features. The user can optionally be presented with the task of selecting a primary *machining approach vector.* This user interface is available when multiple approaches are "candidates" based on the *machining environment.*.

Machining Process Generation: This module presents to the user a *machining operation* following its generation by the system. The user can review the system generated information and perform selected modifications. The first-level modifications would be those that do not alter tool trajectory. These modifications are principally related to *machinability data.* All user modifications are checked by the system to determine feasibility with respect to machine tool capacity.

Machining Sequence Generation: This module provides to the user the opportunity to participate in the review of the *machining operation sequencing.* The user can review the toolpaths planned for a *machining operation* and *transition path elements* to the next machining operation. The user may invoke function that "machines" the stock. This user can also dynamically view "tool where used" graphical queries on the workpiece.

Manufacturing Results and Analysis: This process module employs the user interface to provide the user with all of the information generated by an implementation of the framework. This user interface is primarily available for user analysis of the system generated results. This application of the *generative user interface* is the other primary access point by the run-time user along with *machining environment formulation.*

5.3.3 Manufacturing Data

This system component of generative numerical control is a valued resource to many of the architecture process modules. The connectivity established for the architecture provides this *manufacturing process data* to the following modules as placed in the evolving architecture of Figure 46 on page 123.

- Machining Environment Formulation
- Material Removal Identification
- Machining Process Generation
- Machining Sequence Generation
- Auxiliary Command Generation

The run-time generative NC user will access this data in read-only mode. This static source of manufacturing processing technology is represented as parameters that have been valuated to bring company procedures or standards into the part programming session. Each part programmer receives an exact instance of this data at the point of use for a given parameter during the processing flow. The user retains the authority to modify his instance for best application to his specific programming requirements.

The following point of use examples highlight the resource that *Manufacturing Data* provides to the generative NC architecture.

Machining Environment Formulation: This process module receives the instance of *manufacturing data* and provides it to the user for acceptance or modification. The data elements that have proven most interesting to generative numerical control are the standard values for *algorithm control* and *application control* options.

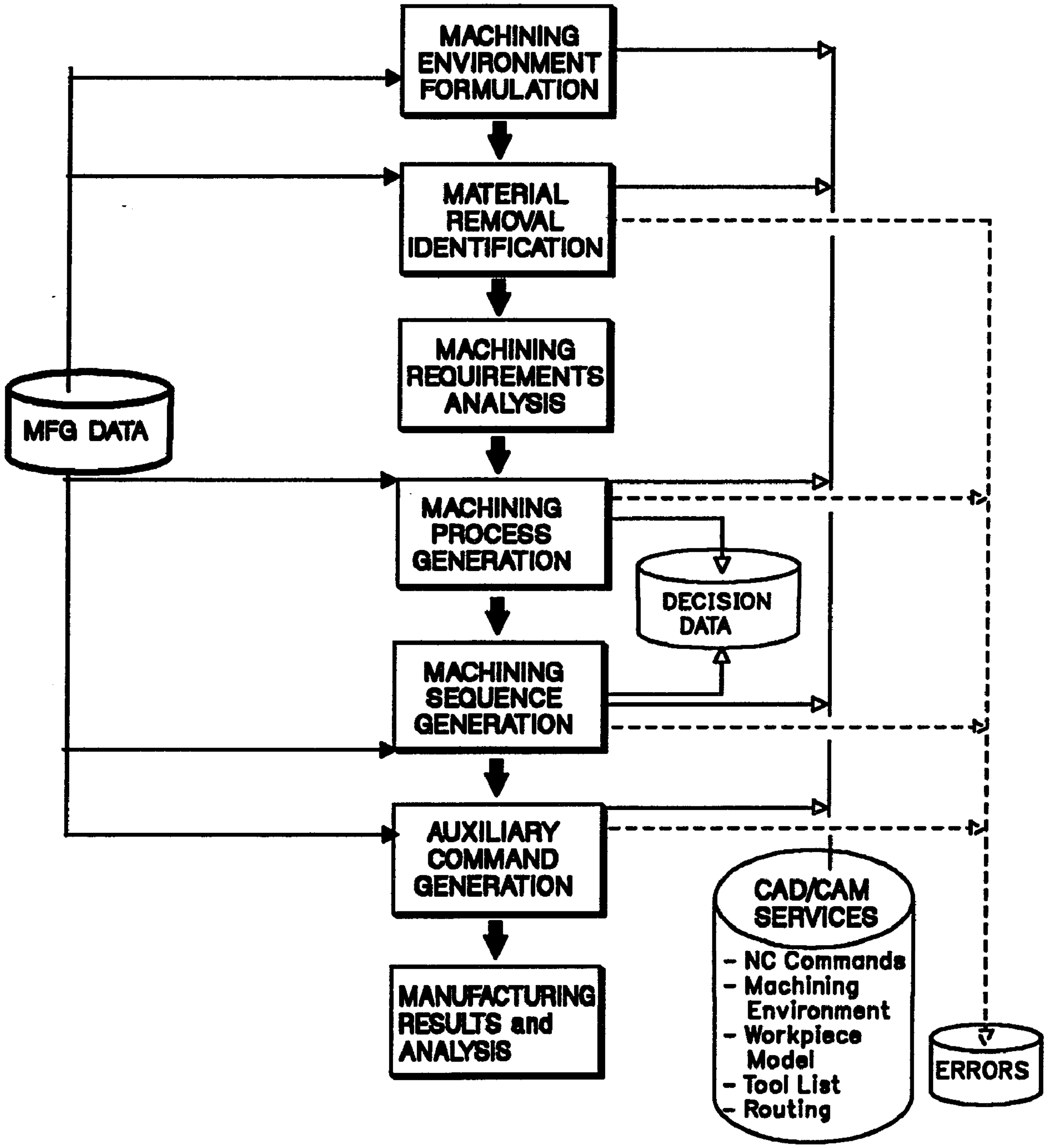

Figure 46. Generative numerical control processing with Manufacturing Data.

For example, the user could invoke a panel to review the parameter standard value for "part_clearance_during_transitional_paths." The user task is to accept or modify, for this part program, this parameter value. This value will specify the distance that will be maintained between workpiece and cutter during "rapid" feedrate moves.

Material Removal Identification: This module would look to *manufacturing process data* for the possible *machining approach vectors* of each machining feature. These possible presentations of the feature to the machine spindle would initiate functions to create a subset of candidate spindle approaches. These candidate approaches would be established by satisfying the set of constraints provided by part geometry, fixture, setup, and spindle configuration.

Machining Process Generation: This module looks to the *manufacturing processing data* to identify standard values for "finish_material_thickness." This value is used in the case that a finish machining pass is specified in the machining process generated for a machining feature. This parameter is typically found in multiple values dependent on the workpiece material.

Machining Sequence Generation: This module is in need of *manufacturing process data* that provides the "sequence candidate selection." This multiple values data identifies the priority of sequencing between *tool change, tool axis change, tool size, machining operation type*, and *minimum path distances.*

Auxiliary Command Generation: This module accesses *manufacturing processing data* as a resource to determine the "number of go-around paths" that will be attempted by the *transition path generation.* When this number is reached the strategy will be changed to again attempt a collision free path that observes the spatial constraints defined for that problem.

5.3.4 Manufacturing Knowledge

This system component of generative NC is a unique resource to providing a very discrete service to just two of the architecture process modules. The connectivity established for the architecture provides this manufacturing knowledge to the following modules as placed in the evolving architecture of Figure 47 on page 126.

- Machining Process Generation
- Machining Sequence Generation

This connectivity provides *manufacturing knowledge* to the core of the generative numerical control problem. One can review the process modules as developed for this framework that establish the processing spine of the architecture. This review could characterize the process modules of *Machining Environment Formulation, Material Removal Identification,* and *Machining Requirements Analysis* as sets of "preparation activities." These pre-processing type modules are undertaken in support of the manufacturing operation planning. As the suggested review continues, the process modules of *Auxiliary Command Generation* and *Manufacturing Results Analysis* are found. These modules can clearly be characterized as post-processing sets of activities. Once again, this processing supports manufacturing operation planning.

The resource connectivity to *Machining Process Generation* and *Machining Sequence Generation* is a development direction that is based on matching available technology to the problem solution requirements. The knowledge base can aslo deliver resource to other modules in the future. The author proposes that these modules will be in addition to those developed here as the architecture spine. "Recommendations for Further Study" provides this visibility to the reader for further application of *manufacturing knowledge.*

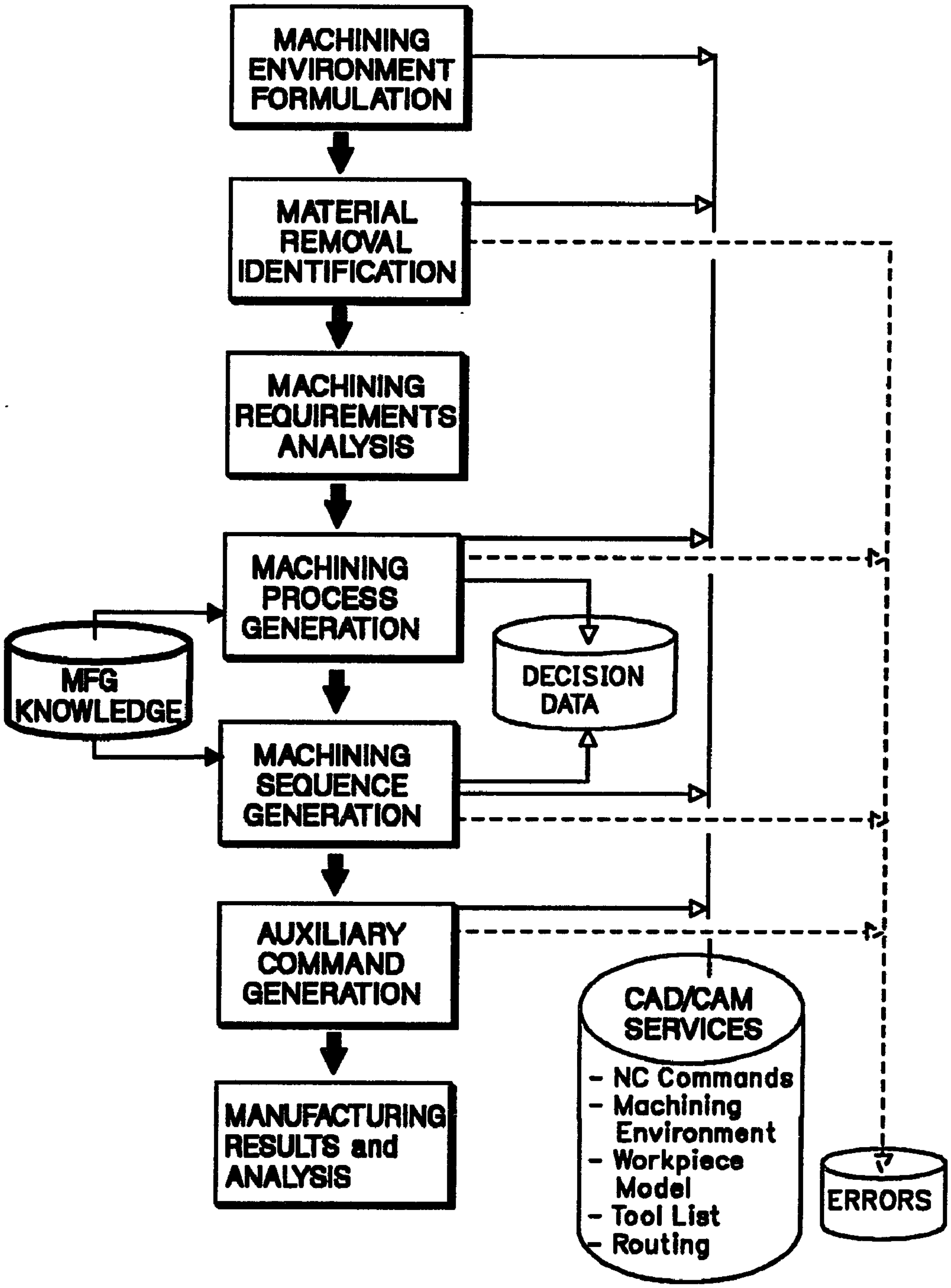

Figure 47. Generative numerical control processing with Manufacturing Knowledge.

The following point of use examples highlight the resource that Manufacturing Knowledge provides to the generative numerical control architecture.

Machining Process Generation: This module relies on *manufacturing knowledge* to set the machining attributes of the machining operation generated for milling a large hole feature. This feature's physical properties excluded it from receiving holemaking strategy and related axial type processing from the architecture. Manufacturing knowledge, in the form of rule sets, is executed for each of the following types of *machining operation* attributes:

- **Trajectory Type** may be a decision between pocketing and area clear trajectories as supported by the NC processor.
- **Machining Direction** would include back and forth, helical-in or helical-out control.
- **Finish Passes** is the specification with *finish thickness* as required by surface finish and surface controlling tolerance.
- **Approach Strategy** and **Retract Strategy** to determine entry and exit motions.
- **Machinability Data** will be consulted to identify the depth-of-cuts recommendations in both the tool axial and radial perspectives leading to *level and pass logic* specification for the machining operation.

Machining Sequence Generation: This module is in need of *manufacturing knowledge* to determine resolution for tool assembly collision when *component substitution* provides no results. Manufacturing knowledge is used to "qualify" the situation. The facts derived from this activity are then used to specify a system solution to the tool assembly collision. This structure of *manufacturing knowledge* also provides the process module with the capability for evolutionary growth as more collision resolution knowledge is implemented.

5.3.5 Numerical Control Database

This system component of generative NC is a resource providing a broad service of manufacturing object instances available to the architecture process modules. The connectivity established for the architecture provides these NC objects catalogs to the following modules as placed in the evolving architecture of Figure 48 on page 129.

- Machining Environment Formulation
- Material Removal Identification
- Machining Process Generation
- Machining Sequence Generation
- Auxiliary Command Generation

The run-time generative numerical control system accesses this data in read-only mode. This source of *NC objects* is defined as parameters that have been valuated to identify the machining capability standards into the part programming session. These standards are representing the physical and logical characteristics of the objects such as machines, materials, stock forms, tools, tool assemblies, and machinability data. Each part programmer or session receives an exact instance of this data at the point of use for a given parameter during the processing flow. This instance is essentially a snapshot of the machining shop at the time of manufacturing operations planning. The *NC database* remains a "static" resource during system execution for a part programming session. There can be little merit for run-time user modification of many of the parameters that define the NC objects. For example, modification of "tool_change - auto" to a machine's database entry of "tool_change - manual" proposes implications in the shop that are logically impossible to achieve during NC program run.

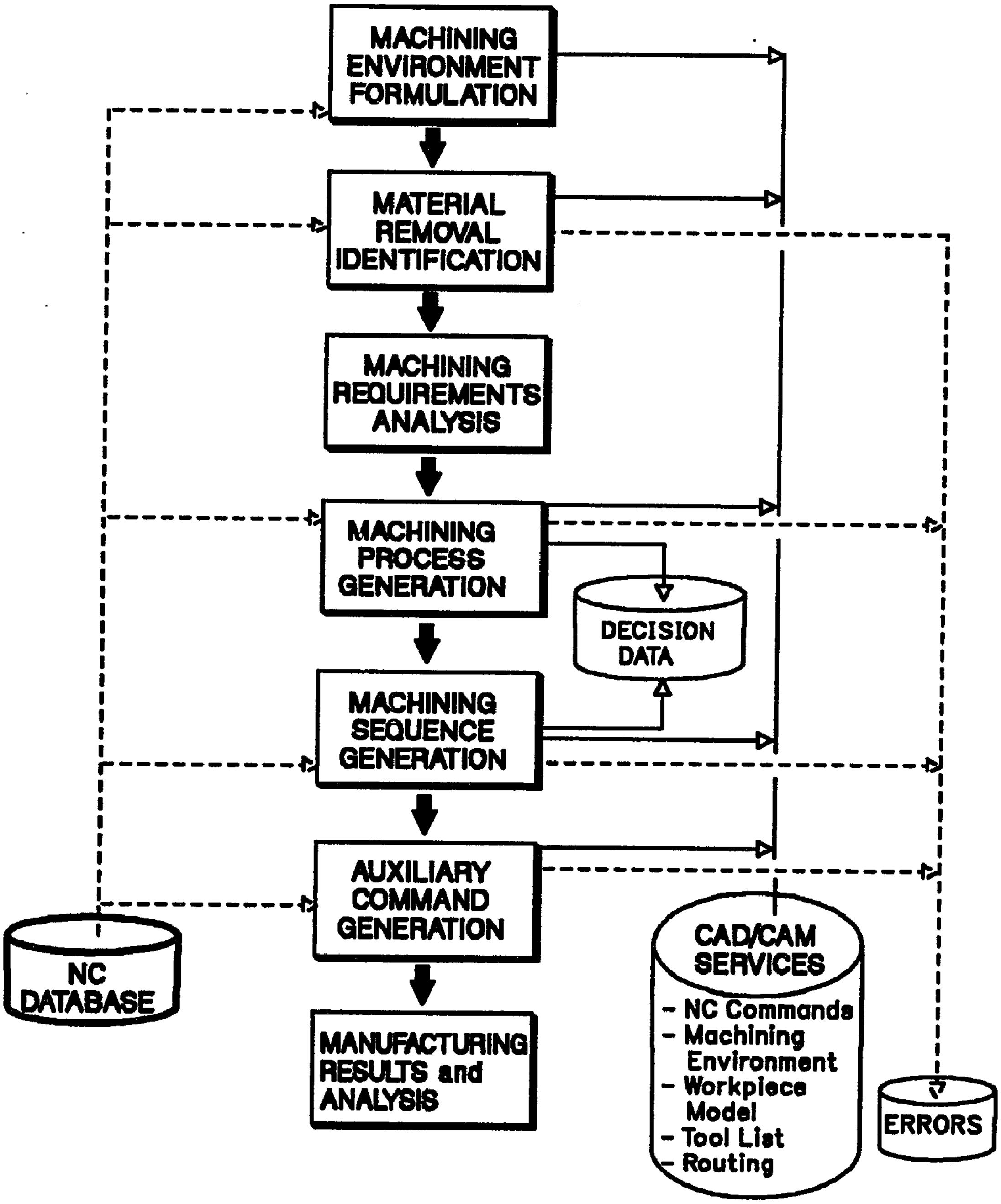

Figure 48. Generative numerical control processing with NC Database.

However, specific parameters are made available for modification to the run-time user of an implementation of this architecture. Parameters such as "program_ends_home" can readily be modified for one part program to "program_ends_X,Y,Z." This theme continues for "tool_change - manual" with attribute "tool_change_point = X,Y,Z." For these types of parameters, the user retains the modification authority to apply the *NC database* standard object definition to his specific part program requirements.

The following point of use examples highlight the resource that NC Database provides to the generative numerical control architecture.

Machining Environment Formulation: This process module engages NC database to validate a user specification of *machine* or *workpiece material.* The database can be also accessed to provide the implemented alternatives prior to a machine or material specification. In this form of application, the NC database is providing the standard shop configuration available to the part programming session. Having accessed a *machine object* in the catalog, specific parameters are extracted. These parameters are used to build the *NC workspace* for the part programming session. One element of the workspace, the *machining envelope,* is built using the "X,Y,Z axis_travel_limits." The applicable "dead_zone" parameters are also extracted from the NC database to complete the modeling of this allowable machining region spatial programming constraint.

Material Removal Identification: This module would look to the MC database for the definition of possible spindle configurations on a machine. This data is extracted from the parameters that identify the capabilities for translation and rotation within the *machining envelope*. This movement is necessary to align a machining feature's possible *machining approach vectors* with the machine spindle. The result is to confirm "accessibility" for each machining feature in the current setup.

Machining Process Generation: This module looks to the NC database as the source for available *cutting tools data*. This data is accessed through a request on the database to identify a tool meeting an explicit set of constraints. The tool object parameter values would identify both the physical and technological composition of the cutting tool. In this module, the focus of tool selection is on the "effective material removal portion" of the cutter.

The user has the ability to retrieve and modify an existing NC database tool object. Using this facility, the user assumes responsibility for these modifications. This responsibility is to provide a cutting tool with the specified dimensional and technological integrity for this extrapolation of implemented NC database.

This module builds a model of the tool from extracted database values. This model is then used by the module to determine the quality of the generated *machining process*. The context of "generated path quality" is based in providing collision or gouge free machining. This module uses the cutting tool model with the *machining area, approach path,* and *retract path* to validate incremental machining process quality.

Machining Sequence Generation: This module packages requests on the NC database for required tool assembly configurations. The database provides information on available tool assembly components to dynamically build and configure a tool assembly. The database is also a source of *standard tool assemblies* that exist for a shop and implemented to be available at a *tool assembly object* level. This information is used in this process module along with explicit "tool set lengths" as available in the database.

The Machining Sequence Generation module also makes use of other parameters that define the objects in the NC database. This module retrieves the "tool_magazine_capacity" from the *machines* object. This parameter value is used to manage the total number of tool assemblies specified within a given part programming session. Additional *machines* parameters are retrieved for "maximum_tool_length," "minimum_tool_length," "maximum_tool_diameter," and "maximum_tool_diameter_two_adjacent_magazine_pockets" to provide a complete definition of the *tool assembly formulation* environment.

Auxiliary Command Generation: This module accesses the NC database as a resource to identify the "post_processor_name" as a parameter of the *machines* object. This parameter provides information as to which post processor will be invoked to translate the *NC commands* output to the job stream expected by the machine controller. This parameter is also used as a "pointer" to machine controller specific syntaxes that will be used in the formulation of *program auxiliary commands* by this module. Theses syntaxes are used to evaluate and populate commands for the purpose of program start, tool change, tool axis change, and program end.

5.3.6 Generative Numerical Control Architecture

The *architecture* is formulated by assembling the incremental connectivity established between each *system component* and the *processing modules.* This procedure will establish the overall view of the system components of generative NC and the processing modules. Collectively these elements of the framework form the *architecture of generative numerical control.*

The architecture diagram is created by a simple overlaying of each *system component connectivity figure.* The resultant figure provides the comprehensive view of the architecture for generative numerical control. Figure 49 on page 134 depicts five of the system components defined for generative NC and the seven *processing modules* developed as a solution for the industrial production NC machining environment.

This formulation clearly displays a paradigm shift for *manufacturing operational planning* and *NC part program preparation.* This paradigm shift is also evident for the part programmer, NC engineer, and toolmaker. This paradigm shift is enabled by the:

- Introduction of Solid Modeling Techniques to this Manufacturing Problem,
- Structuring of the NC Part Programming Procedure, and
- Formulation of a Machining Process Perspective.

This figure does not represent the synergy established by multiple system components being introduced simultaneously to a *processing module's activities* This synergistic effect is not intuitive through a stepwise development of singular component connectivity with the processing modules. This topic is reserved in the research to a following section on "Architecture Controlling Concepts."

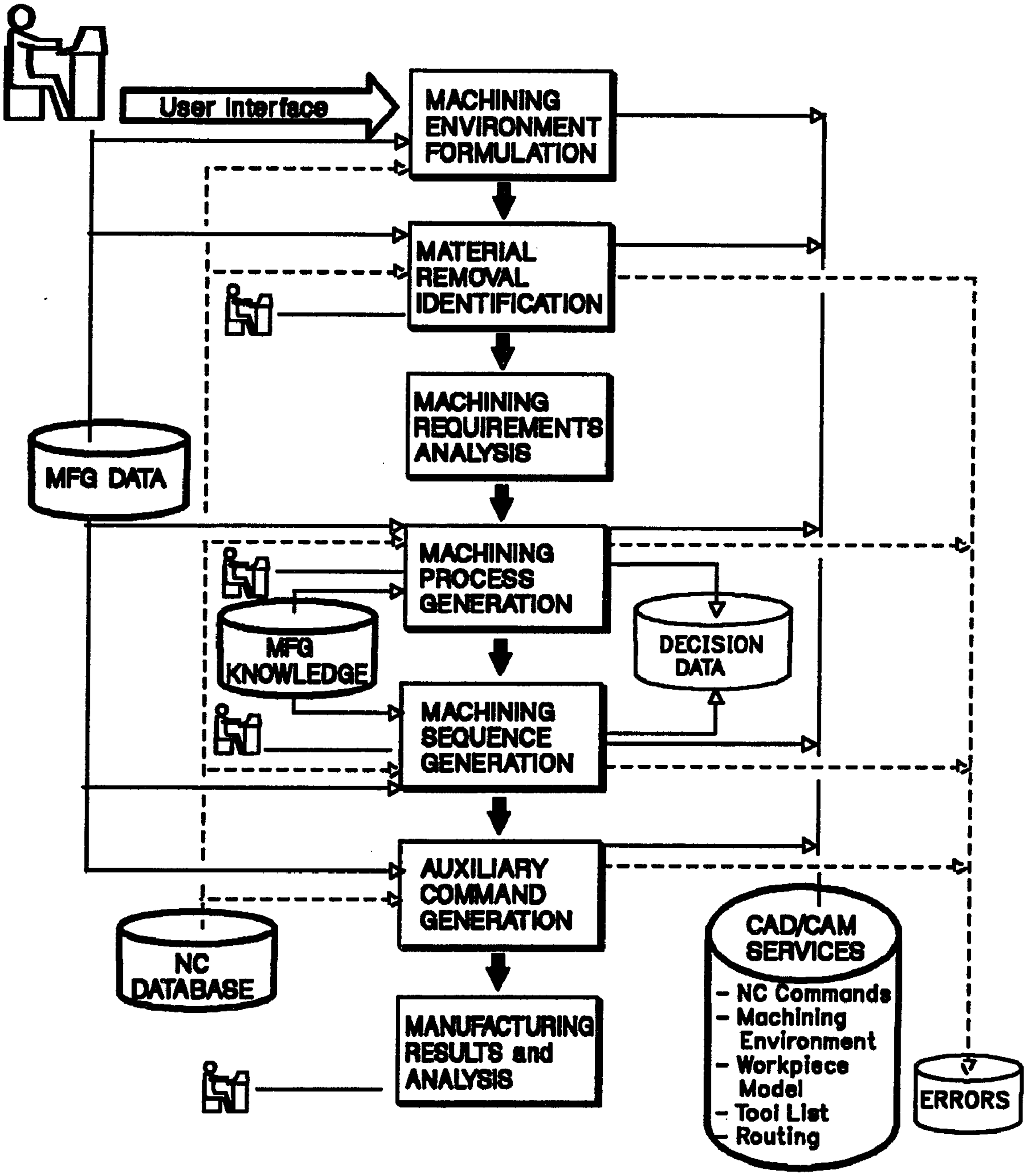

Figure 49. The generative numerical control architecture.

5.3.7 Generative NC Architecture IDEF Model

Purpose to define the functional architecture of a generative solution to the *operational planning* and *NC part programming problem task* as shown in Figure 50. The **Viewpoint** chosen is that of the automated manufacturing system engineer.

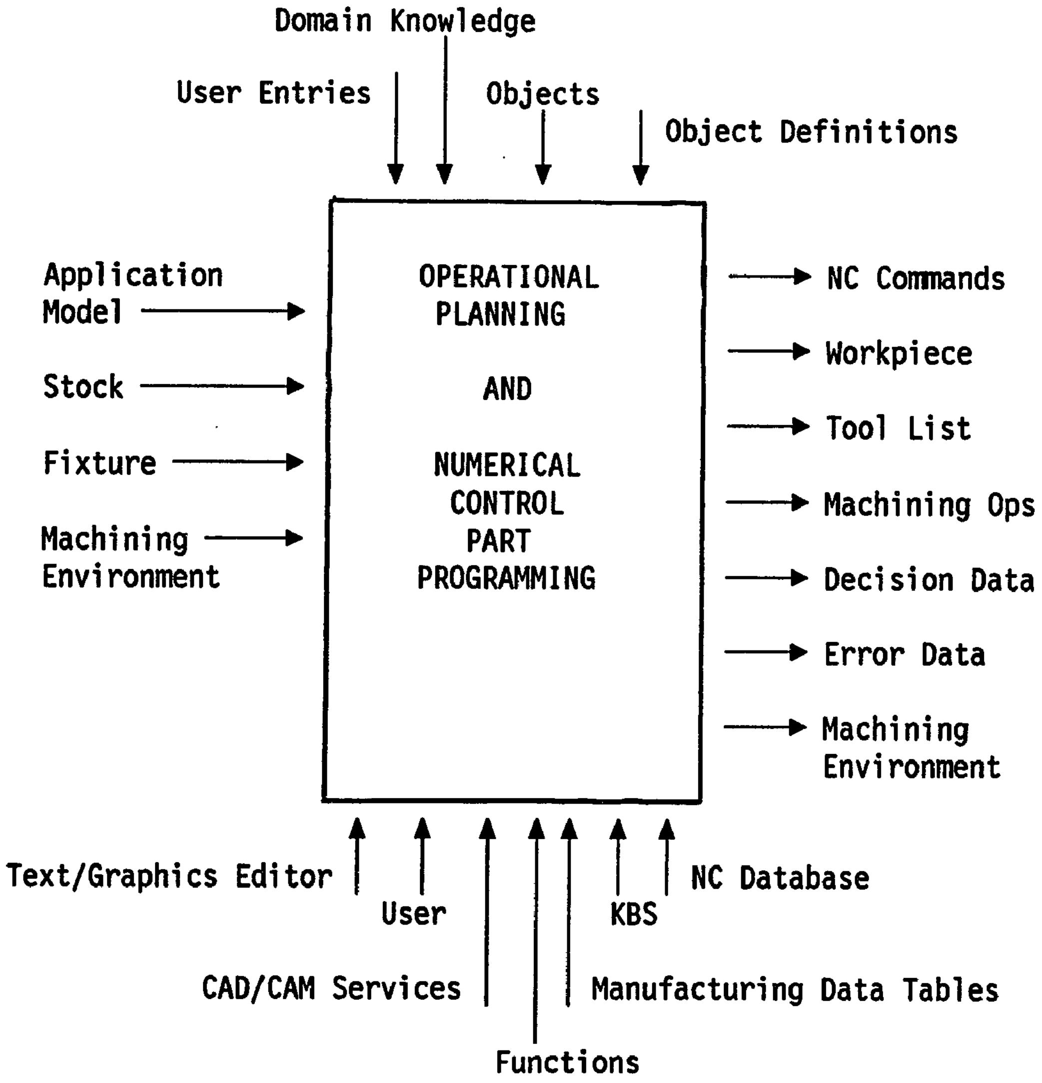

Figure 50. IDEF functional model for the architecture.

5.4 Architecture Controlling Concepts

The development of the framework continues with the identification of a group of concepts that provides additional specification of the solution and control of the architecture. These concepts were not intuitive in the development of the processing modules. Also, the concepts did not specifically present themselves as the systems components were assembled with the processing modules. Nevertheless, these concepts are critical to the generative numerical control framework and are developed herein.

This segment of the research has taken the nomenclature of *architecture controlling concepts* due to the type of contribution it proposes to establish with the framework. These topics are providing the architecture with the capability to be the spine of a fully automated NC part programming system. This spine is designed to establish a full function part programming solution where customization or tailoring of the system will not require redesign effort at the architecture level.

Each concept is developed within the context of the *system components* and the *architecture* established to this level in the research. In many cases, this section will provide this research presentation with the techniques that clarify this proposal for the a *framework for generative numerical part programming.*

These concepts promote the bounding of the manufacturing operational planning problem. This additional specification of the technology of generative NC further establishes the reduction-to-practice for an industrial production part programming solution.

5.4.1 Machining Environment Data

Machining environment data is developed as an object of generative numerical control. This architecture controlling concept is defined to be the collection of data necessary to enable system execution of an implementation of generative NC technology. This object includes *user inputs, algorithm control processing data,* and *application control* options as defined in a specific implementation. This object is documented and maintained by the application as an externalized data structure. This controlling concept to the generative NC architecture is graphically presented in Figure 51.

The contribution of this object is realized in the architecture to enable the capabilities of *off-line data preparation.* This object also provides an efficient entry point for *manufacturing results analysis* with simplified re-submission of a job after *error* diagnosis and correction.

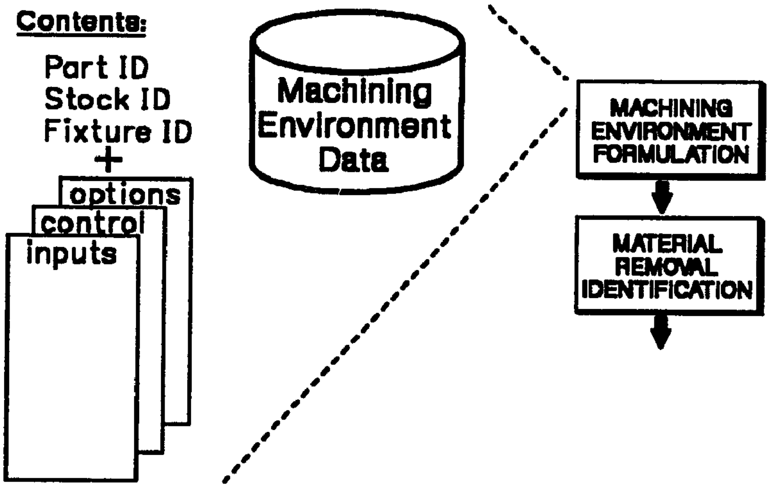

Figure 51. Machining environment data element of the architecture.

5.4.2 Off-line Data Preparation

The data preparation requirement for automatic NC part program generation can be accomplished in this framework through several techniques for *off-line data preparation*. The architecture controlling concept must emulate the contribution of *machining environment formulation*. This concept is therefore enabled by the *machining environment data* object. Off-line data collection is integral to the generative numerical control architecture. The concept provides a greater flexibility to extract the required machining environment data in preparation for a part programming session and system execution. Off-line data preparation can take two forms within the context of the architecture. The data can be entered by a user or by an external application program. Figure 52 presents these two forms in the context of the more conventional run-time generative user interface. In all cases, the structure and content of this resultant data is exactly the same. The *machining environment data* is the object instance that is created by these three techniques of system startup data preparation.

Data preparation from an external application program is of particular interest to a computer-aided manufacturing system. This capability will promote the goals of computer integrated manufacturing in the creation of NC part programs. Part programming requirements can be generated by the manufacturing records system that prepares the required data (in whole or part) necessary for *machining environment formulation*. This result of an implementation of this concept is that the data preparation tasks of *machining environment formulation* processing module can optionally be disjoint from the system execution. This capability minimizes the requirement for utilization of the traditionally more expensive graphics display devices for the data preparation requirements of the generative NC programming system.

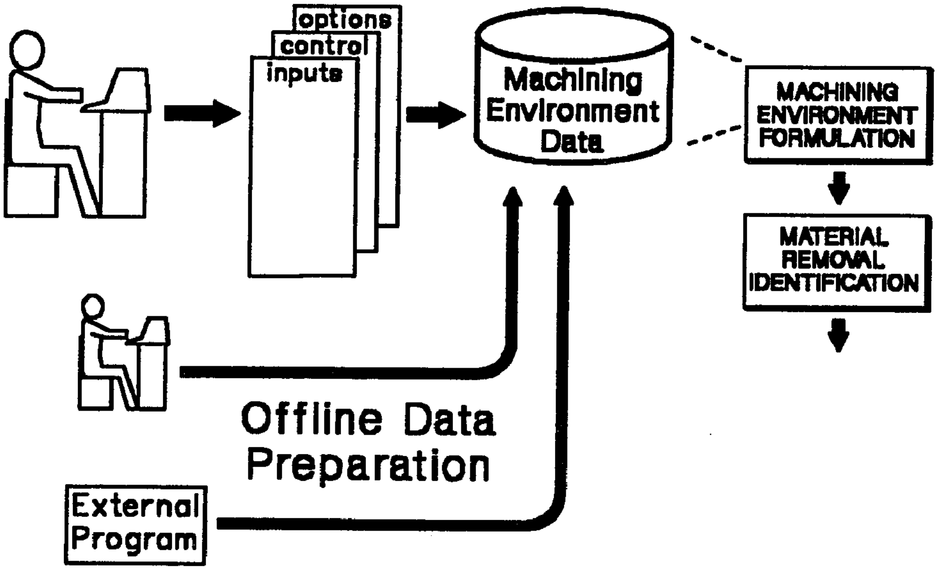

Figure 52. Off-line data collection integration with the architecture.

The creation of an off-line data preparation facility enables another contribution of the architecture developed through this research. Off-shift computing resources can now be scheduled for generative numerical control part programming. These unmanned job executions use *machining environment data* as available in *declarative file* form. The machining environment data could have been written as a result of an off-line data entry session or from a production requirements job scheduler.

The implementation of this architecture controlling concept is not trivial in current production CAD/CAM systems. Attention must be given to the unique detail of job submission and data flow requirements specific to the target CAD/CAM system.

5.4.3 Knowledge Implementation with Turnkey Alternatives

The *manufacturing knowledge* brought to the user through the generative numerical control architecture requires the structuring of manufacturing know-how. This know-how is formalized in rules and rule sets as previously developed. However, this concept remains the most intimidating aspect to the generative numerical control framework. For this reason, *knowledge implementation with turnkey alternatives* is developed. This controlling concept provides a logical transformation of rule based knowledge to manufacturing data or method files. These derivations of the traditional expert system approaches still constitute knowledge processing [64]. Implementation considerations must remain cognizant of "system source master level" while providing knowledge tailoring functionally.

Turnkey implementation can be delivered that provides treatment of this system component as "data." The *manufacturing knowledge* is imbedded within a full function machining solution. This task can be undertaken to minimize the commonly perceived complexity of dealing with a knowledge base or expert system. This form of knowledge introduction into a system will be similar to the expert system implementation defined for some parametric modelers. Explicit rule formulation and traditional rule processing is transparent to the user to define and constrain a parametric model [13].

Another turnkey implementation will reduce the manufacturing knowledge to the data level and proposed to the user as a formatted "methods file." This solution can be implemented by proposing to the *manufacturing knowledge* a structured set of basic machining problems. The output *machining processes* are captured and form the basis for "methods file" based knowledge implementation. The structure of the part programming approach is also modified as the knowledge methods become variant objects of manufacturing operational planning.

5.4.4 Imbedded Algorithm Hierarchy

Many algorithms will need to be assembled to provide the functions of the generative numerical control. The execution of these algorithms represents a significant portion of computing resources for the overall processing and solution. Therefore, it is essential that these functions be invoked "as required" for each part programming problem. The *algorithm hierarchy* controlling concept provides for level of processing developed within the architecture. This concept directs the processing efficiency and performance that is realized in this part programming system. The *algorithm control* processing hierarchy is specified within *machining environment data.*

The effective level of solution to this situation in many implementations of this technology is proposed to provide user function that will be presented in the *Machining Environment Formulation.* This capability "sets" the applicability of the functions or level of processing required in the specific part program. Setting is established by user review and anticipation of the processing requirements for that specific part. These levels of processing questions will also be answered by commodity type and available to system execution through the *declarative file default settings.* This user input cannot be readily substituted with programmed code in the application of current industrial tools and techniques to this problem.

This concept is specified by the architecture to apply throughout the processing modules. A candidate typical of algorithm hierarchy is identified below for machining drop-off.

Drop-off Permitted Y/N: This portion of the algorithmic processing analyzes that the creation of a slug for drop-off as opposed to reduce-to-chips machining is an acceptable strategy to incorporate in a machining process. The required geometrical and technological queries and computation are executed or not executed based on the user's answer to the question.

5.4.5 Precedence Orders Data

This concept provides a singular relationship used to record and manage all orders between *machining features* and *machining operations*. A *precedence relationship* is a link between two machining features or machining operations. This link establishes an order data ultimately used for *machining operation sequencing*. The sources of the order data are varied as established within the generative NC architecture. *Process, access*, and *user* are the types of precedence as represented in Figure 53. The source of each precedence type is also identified in this figure. Delbressine [15] limits precedence to "reachability" or access only. Zhang [78] proposes this same limited approach and then adds a "good machining practice" modifier.

Weak Precedence establishes a relative order within the pair. This allows machining operations to be sequenced between the pair. This relationship constrains "relative position." **Strong Precedence** establishes an immediate order within the pair. This constraint would require that the two objects be sequenced one directly after the other.

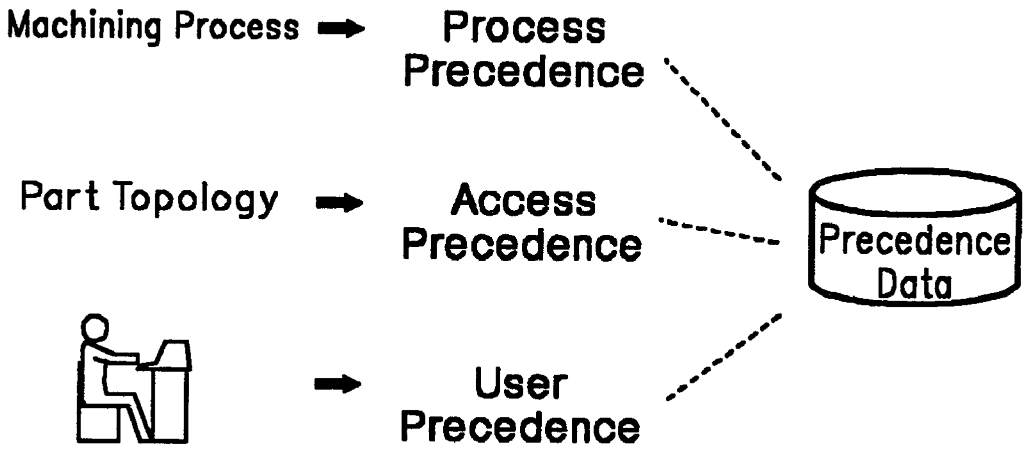

Figure 53. Sequence precedences for generative numerical control.

5.4.6 Machining Feature Based Processing

This concept of the generative NC architecture is directed toward the object used both external and internal to the system core level processing. The *machining feature* is this object. The machining feature is comprised primarily of the material removal volume, machining approach vector, and technological attributes. Qualification of a machining feature is a critical activity in the generative NC processing. This activity establishes instance completeness for utilization by the complement of the part programming functions.

This same data structure is maintained for every volume of material that receives treatment from the generative system. This treatment employes the same machining feature class definitions to:

- Represent a Machining Feature,
- Represent a Near-Net-Shape Machining Feature,
- Qualify a Solid Volume to a Machining Feature,
- Translate a Design Feature to a Machining Feature,
- Quantify a Part-in-Stock Machining Feature,
- Qualify Open and Closed Profile Type Machining Features,
- Represent Not-Machined Volumes from Machining Process Generation, and
- Represent Finish Geometry During Rough-Only Processing.

Figure 54 presents the rigor of this controlling concept in the context of the *machining process generation.* The reader recalls this is a core processing module of the architecture. This process accepts a machining feature instance as input. The processing of this step in the architecture applies *machining strategy* to this *machining feature* to determine a *machining process* composed of one or more *machining operations.*

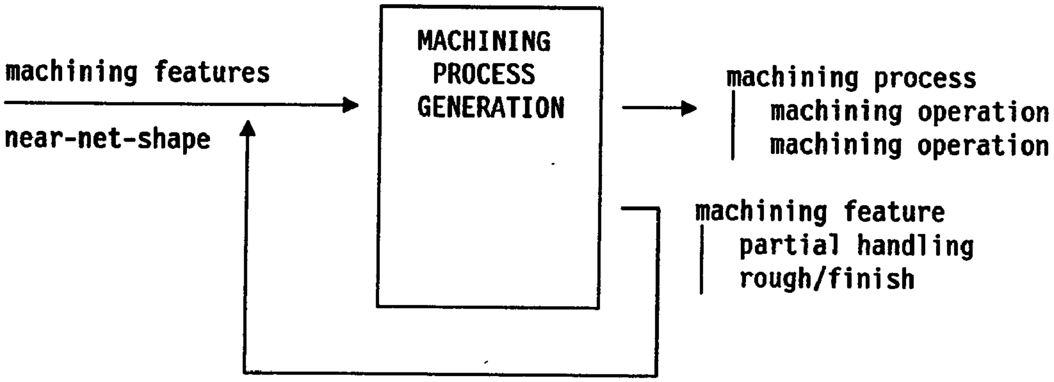

Figure 54. Machining feature concept rigor for internal data modeling.

This processing remains consistent over machining feature class extensions. The external design of this process carries an input definition for matching feature instance. This definition is qualified without differentiation as to the source of the material removal volume. The sources map directly from the possible treatment of machining feature class instances:

- machining feature,
- near-net-shape solid,
- part solid,
- design feature,
- part-in-stock volume, and
- machining profiles.

The figure also identifies another output from *machining process generation.* This output is the representation of a *remaining volume* of material after the machining process has been generated. The form of this remaining volume is also a machining feature instance. This feature is available to the architecture for subsequent processing.

5.4.7 Machining State Variable Control Processor

This concept is integrated with the generative NC architecture to establish a processor that can complete a part program specification. As such, this concept provides control for *auxiliary command generation.* The task is to read *sequenced machining operations* and extract "implicit" data that would indicate a requirement for tool change, machine table rotation, inter-operation path, and any other program complete requirements. This processor is extendable as evident in the Figure 55. This extension is affected through state comparison invoking function processing. The *state variable processor* adapts to any of the possible configurations required by a sequenced machining operation specification. This control reads "next" state, compares with "current," and directs to required function. This architecture controlling concept is proposed as an alternative to a complex procedural solution.

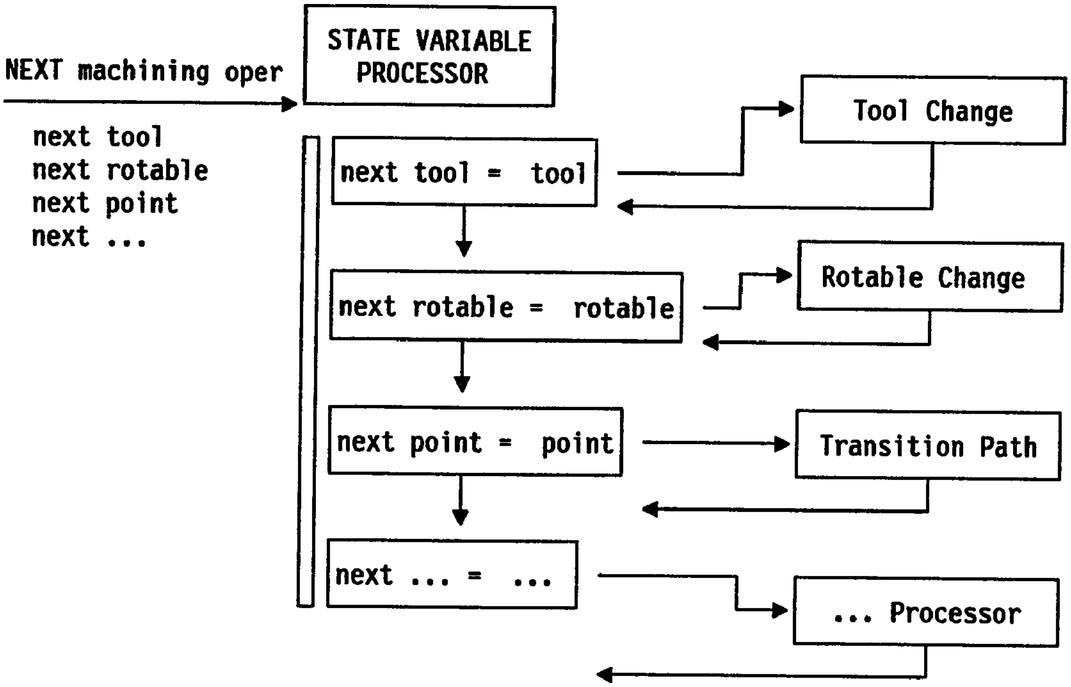

Figure 55. Machining state variable control processor.

5.4.8 Manufacturing Objects Modeling

This concept creates the models for manufacturing objects used by the generative NC architecture. This modeler employs the techniques available in *CAD/CAM services* to create a representation of a machining environment physical entity. However, this modeler uses a parametric approach whose source data is found in the object instance of the *NC database*. This approach provides variable object model forms based on their usage in the framework. This usage could require a complete representation of the object or a simplified symbolic form. This concept will be further developed in the context of a multiple stage tool assembly object as shown in Figure 56. This assembly can consist of a *tool*, *extender*, and *tool holder*.

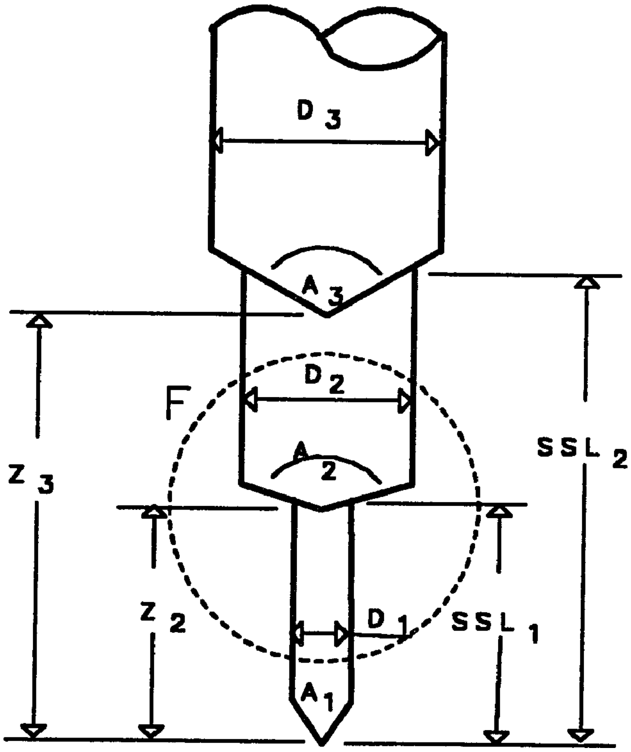

Figure 56. Tool assembly object modeling.

The model creation technique for each object is based on an extraction of source data from the *NC database*. This approach differs from the common theme in the literature in that a secondary "graphics object source" does not complement the *NC database*. This disjoint "graphics object source" seems to offer disadvantage to planning quality NC toolpaths in an environment supported by a robust manufacturing database. A single source of manufacturing database is advocated in this research to provide all *geometrical* and *technological* attributes needs of a generative system.

The following treatment defines the controlling equations to be used for Z-translations of primitives in the modeling of a multiple-stage tool assembly. Tool assemblies are systematically defined with this generative NC solution as a set of components and the associated tool stage set lengths. The reader can consult *Analysis+ Tool List* on page 74 for a typical description. The problem is the determination of the Z-translations for the physical modeling procedure. The definition for each primitive can be build using a rotational sweep technique. The attributes for this sweep primitive are extracted from the *NC database*. The assumption made in the development of the controlling equations is that the interconnection between each stage can be approximated with conical volume. The equations will also be reduced in form as this interconnection approaches a flat surface. This assumption emulates the hardware complexities at this point of connectivity with a "conservative geometric form." This choice also allows the treatment to handle modeling for *multiple stage tools*. The readers attention is focused to Figure 57 on page 148. The Z-translation value for the stage 2 primitive is defined from the analysis of this geometry as below:

$$Z_2 = SSL_1 - [h_1] \qquad (1)$$

For stage 2 this equation is expanded in terms of the *included angle A* with the result being:

$$Z_2 = SSL_1 - \left[D_1 - 2TAN\left[\frac{A_2}{2}\right]\right] \tag{2}$$

Similar reasoning applied to stage 3 will develop *equation 3*:

$$Z_3 = SSL_1 + SSL_2 - \left[D_2 - 2TAN\left[\frac{A_3}{2}\right]\right] \tag{3}$$

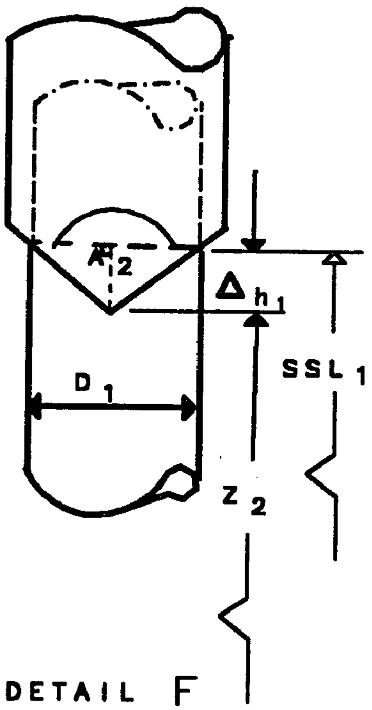

Figure 57. Tool assembly object geometry detail.

These equations can be generalized as functions of the *stage set length (SSL), component diameter (D)* and *included tip angle (A)* as shown in *equation 4* where "n" represents the stage number.

$$Z_n = \sum_{i=1}^{n-1} SSL_i - \left[D_{n-1} - 2TAN\left[\frac{A_n}{2} \right] \right] \tag{4}$$

This *equation 4* degenerates to a simple summation of the *stage set lengths* as the *included tip angle* approaches 180 degrees. This results are shown in *equation 5.*

$$Z_n = \sum_{i=1}^{n-1} SSL_i \tag{5}$$

This solution is proposed as a general treatment for building *variable set length tool assemblies* formulated in the generative NC framework. The solution will also model *multiple stage tools.* However, focus to the concept developed by this example must be maintained. Manufacturing objects are modeled within the system using a single source of data supplied by the *NC database.* This *architecture controlling concept* is developed to constrain programming error introduction through a single source of machining component data.

5.5 Architecture Modes of Operation

The architecture is developed to provide fully automatic part program generation. However, this mode of operation is not always preferred by the user. Unique part geometry configurations or manufacturing processing requirements may suggest user involvement or monitoring of the architecture processing. Early user involvement during each individual's "certification" of the architecture's solution is another requirement for expanding the automatic mode of operation. Houten [27] confirms this approach with his "recommendation" to allow for more user interaction in his PART system. The run-time user has three *modes of operation* available for system execution within the generative NC architecture. These modes are defined as *automatic processing, run/stop processing*, and *hybrid processing*. Hybrid processing combines the capabilities of both automatic and run/stop processing. These modes of operation are presented to the reader in graphical form in Figure 58 on page 151. In this figure, automatic processing of a module is designated with a solid box. User controlled execution of a module is designated with a dotted box.

5.5.1 Automatic Processing Mode

This mode of operation provides the run-time user with the part programming functions as a single treatment. The application receives *machining environment data* as loaded from user input *panels* or a *declarative file*. The system is then run according to the specified data and processing instructions. This processing level will reference only system supplied *manufacturing data* and *manufacturing knowledge* for all algorithmic and rules based processing toward a system goal of complete part machining. This mode of operation is preferred by users with well structured part programming requirements. This mode of operation also serves those users being conscious of and managing their off-shift computing resources.

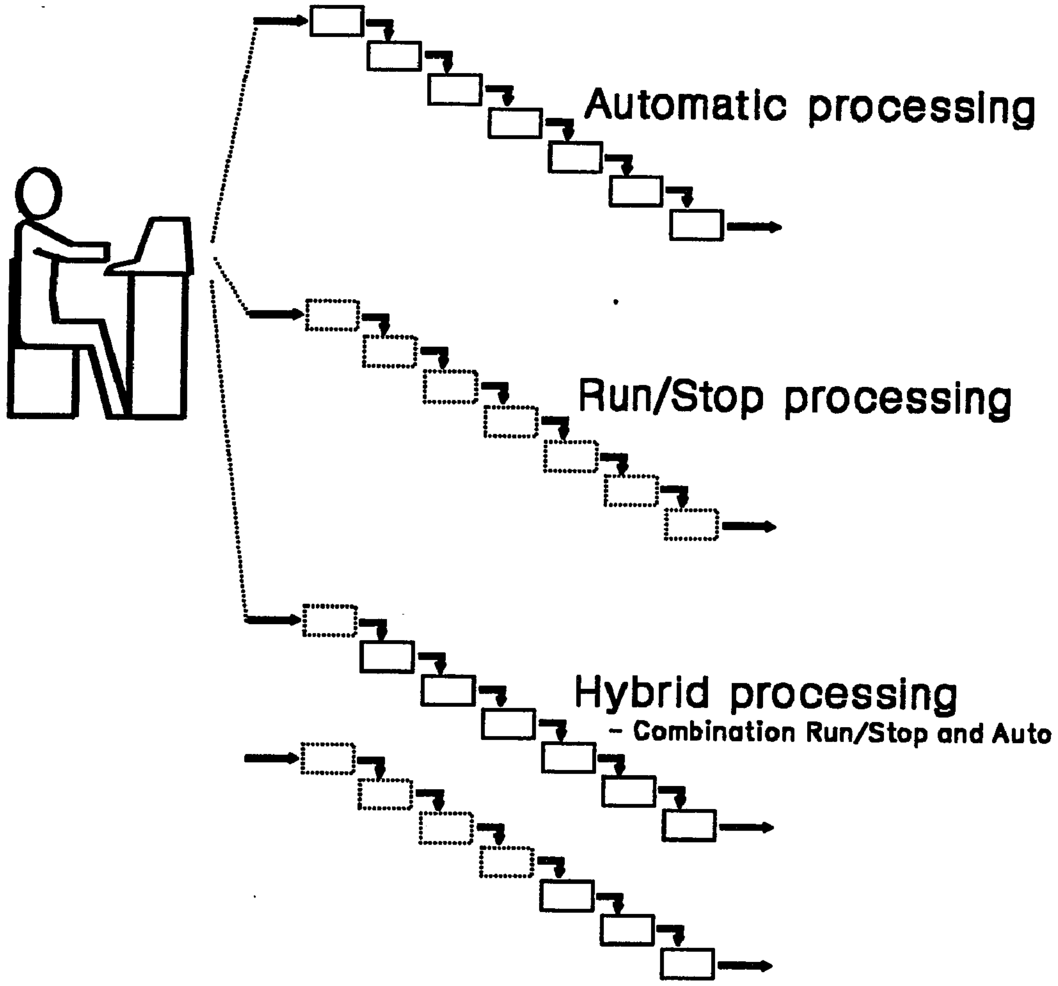

Figure 58. Architecture modes of operation.

5.5.2 Run/Stop Processing Mode

This mode of operation provides user control of the architecture part program processing at the module level. There are seven such modules in the generative numerical control framework. This mode of operation provides run and stop or stepwise treatment of each of the modules to the run-time user. This treatment supports a partial run of the architecture modules within a part programming session. Programming of a particular part can be stopped and restarted in a following session. The stop and start points of the architecture are reserved to be at the logical problem level or module begin and end points. The selection of these points formulates a manageable quality of data problem for generative NC.

This mode of operation will also support iteration at the module level with different user inputs. This concept is presented in the figure below. This use of run/stop processing control broadens the capability of the architecture by introducing *alternative process generation.* The promise of a more optimal part program drives this micro level of manufacturing alternatives formulation. This theme of processing is proposed to be restricted to the user with a high-degree of competence and complex part configurations.

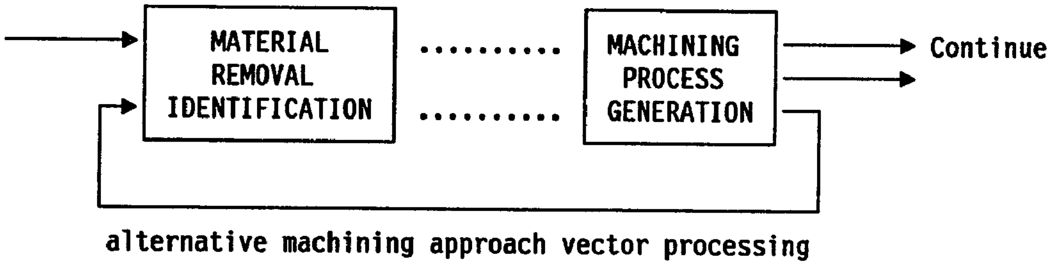

Figure 59. Architecture module iterations toward alternative processes.

5.5.3 Hybrid Processing Mode

This mode of operation combines the *automatic processing* and *run/stop processing* to allow the run-time user to selectively mix treatments. This combination of the basic modes of operation provides the user with maximum flexibility to define generative numerical control processing based on an individual part's anticipated machining requirements. The user may require run/stop mode through a particular point and then review results. These results would then be used to complete execution in the automatic processing mode.

This use of the hybrid processing mode is exemplified with an application model received with complex geometries and having design feature information. The user would typically want to *run/stop process* through the *machining environment formulation* and *material removal identification* modules. The user has taken the opportunity to understand and guide the generation of *material removal volumes*. Then, control is returned to the architecture to continue with *automatic processing* for the remaining modules.

5.5.4 Modes of Operation Domains

The modes of operation for the generative NC architecture are applied based on user competency and part complexity. Empirical data [35] suggests that user competence levels characterized as low, medium, and high can be charted against similar ranges of part or workpiece complexity. The result is the identification of domains for the application of each mode of operation.

---

[35] System mode of operation preferences are based on the author's early implementation of the generative numerical control architecture in industry for production machining. Observations recorded are based on varied industries and user organizations.

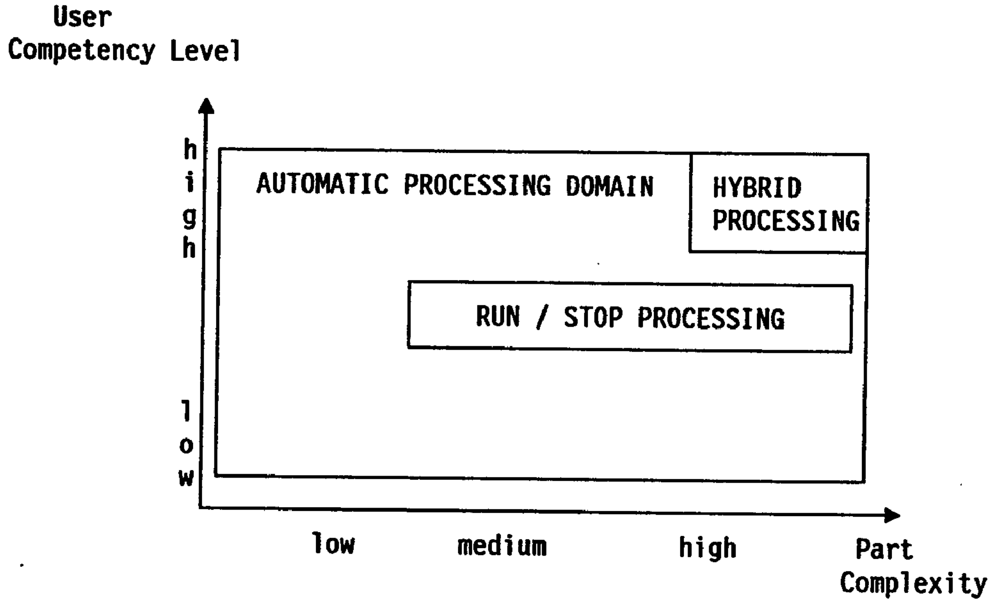

Figure 60. Architecture mode of operation domains.

User Competency Level: There are many facets to user competency with new technology such as generative NC control brought to a production machine floor. In earlier work, Aber and Zimmers [36] propose a technique to introduce this new technology. One facet is the user's competence in the manufacturing technology imbedded in the implementation. Another facet is the skill level of the user. This skill is a requirement for effectively working within the framework to the solution domain serviced by the automatic programming tool. The first competency can be approached with "user education." The second facet will be the focus of "user training." Another facet of user competency is the degree of professionalism that a user directs toward a next generation part programming technique. The author proposes that the organization climate is the most productive manager of this facet of user competence level.

---

[36] Aber, D.M., and Zimmers, E.W. Jr, "The Integration of Engineering with CAD/CAM Technologies," Joint Report Series: Manufacturing Systems Engineering Program, Center for Design and Manufacturing Innovation, and NET Ben Franklin Advanced Technology Center, Technical Report #85-002, 1985.

The user that exhibits a high-degree of overall competence will be more readily able to accept a standard approach applied through automatic programming techniques. This user will be made available to apply this competence to the engineering content of the detailed planning problem where the system may lack imbedded data or know-how. This user will likely provide his knowledge and return control directly to the architecture.

Workpiece Complexity: Once again, workpiece complexity consists of many facets. The number and type of machining features in a part represents one facet. Feature placement in the workpiece and topology of the stock add to part complexity. Relationships between features add yet another layer to workpiece complexity. Technology constraints such as the allowable configurations of material removal volumes contribute to part complexity. These constraints are typically introduced by the selected numerical control processor. Physical part size continues to expand this degree of part complexity. Multiple setups, with specific user machining preferences, creates additional part complexity for NC processing.

The generative numerical control architecture now includes a mode of operation that can be specified for the efficient generation of a part program with consideration for varied degrees of both user competency and part or workpiece complexity.

5.6 Architecture and Framework Dynamics

The goals of computer integrated manufacturing [39] would require a system that can be configured to meet specific user needs. These needs become apparent in the support of company specific methods, standards, and numerical control machining environments. This user configuration of the generative numerical control framework will be given the nomenclature here of "customization." The introduction of manufacturing technology and process into a part programming environment is the catalyst for developing this customization activity. The generative NC solution developed in the research has the goal to provide production part programs. Therefore, the machining technology used to generate the programs must represent the specific shop environment.

Traditional part programming techniques do not integrate the machining process technology into their solution technique [16,19]. Technology is applied to the part programming task through the interactive user's system entries. These techniques concentrate on geometric definition and guidance of toolpath. As such, these techniques can provide a general solution to part programming without system integrated manufacturing technology.

Tailoring the solution of generative numerical control has been investigated in this research. The requirement for this need to customize the framework can be characterized through three distinct perspectives [37] as presented in Figure 61 on page 157. These perspectives are machine shop capacity, shop organization, and new technology approach. These perspectives can be found singularly and as multiple contributors to the tailoring requirement.

---

[37] The author's machining industry survey notes and observations leading to empirical data formulation of tailoring the solution with manufacturing technology.

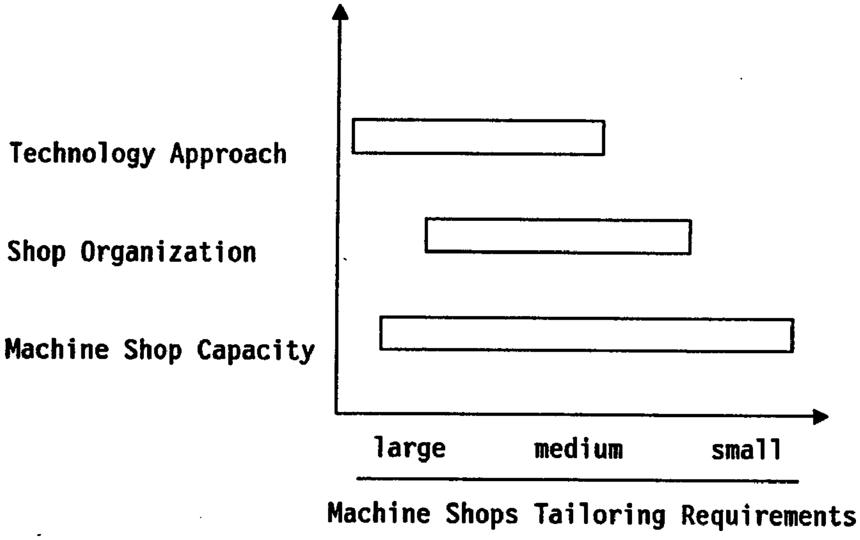

Figure 61. Tailoring the solution requirement perspectives in industry.

Machine Shop Capacity: This requirement for tailoring the generative NC solution is based primarily on two elements of machining shop capacity. These elements are the *machining hardware* and *machining processing* as shown in Figure 62.

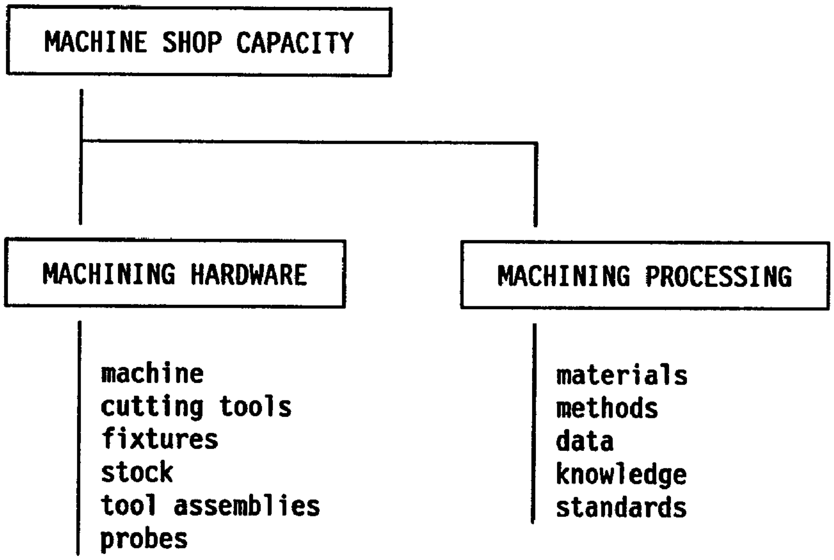

Figure 62. Shop capacity considerations for tailoring the framework.

The *machining hardware* configuration initiates a requirement for tailoring the solution due to physical constraints on part programming. Specific machines, cutting tools, fixture components, and such must be introduced and available to the generative system. A detailed comparison was undertaken of tool holders between two major tooling manufacturers. Each manufacturer provides a standard length V40 connection to the machine spindle. This comparison determined that the physical differences of the "V40 tool holder" standard specification is capable of causing collisions. These collisions are possible if substitution is arbitrarily made without regard for manufacturer source and program clearance specifications. The machine shop capacity requirements for customizing an implementation of the framework are also influenced by the shop's *machining processing* capabilities. Machining processing capability is reduced to the type of materials processed and the methods used for processing. This perspective is contributed to by the data, knowledge, and standards employed for this production processing.

Machine shop capacity perspective is readily deduced from machining differences between two industries. The analogy selected here could be a machine shop capacity difference between a U.S. aerospace manufacturer and a German printing press manufacturer. Both product lines are worldwide leaders in their respective industries in the continued search and introduction of leading edge technology. However, the very basic differences in product functions and manufacturing define dissimilar machining processing capacity.

Machine Shop Capacity defines a requirement for tailoring the generative numerical control solution. This perspective of user need for customization will be substantiated by the basic difference in each machine shop.

Shop Organization: This requirement for tailoring the generative NC system introduces the perspective of *shop organization* or layout. The two philosophies most often encountered by the author's survey of machining facilities are *hardware* based and *commodity* based organization as shown in Figure 63. Although this philosophy of shop organization may be manifest in the shop layout, the organizational impacts toward a generative NC solution are much more extensive. The part programming organization tends to emulate this physical makeup of the machining shop. Tool management services are also a contributor to this perspective. In industry, both of these philosophies of shop organization are commonplace. Production machining may be organized by commodity based structure. At the same plant, tool and model services would emulate the hardware based philosophy.

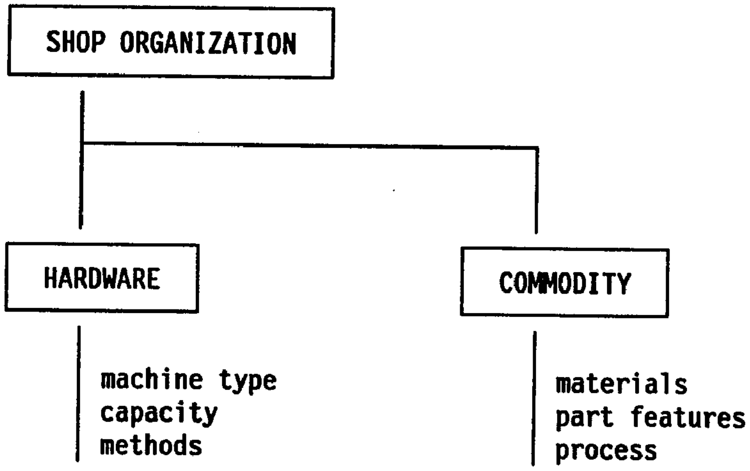

Figure 63. Shop organization considerations for tailoring the framework.

The *hardware* based shop organization presents a part programming environment that provides service to a machine or group of machines. These machines will typically have a singular logical definition with varied physical specifications. That is, a group of single spindle vertical mills with automatic tool changers are treated as a logical machining unit in the plant. However, each machine must be known for the generation of an optimal part program.

The *commodity* based shop organization will typically configure dissimilar machines that each address a portion of the manufacturing requirements of a product line. This concept is referred to as a machining cell [15,21]. Part programming requirements on the generative NC framework would require customizing for this machining cell based on materials, part features, and process.

Technology Approach: This perspective of a requirement for the tailoring of the generative NC solution is a statement of company culture with respect to the introduction of new technology. The two philosophies observed by the author can be defined as *production priority* and *pilot priority* as presented in Figure 64.

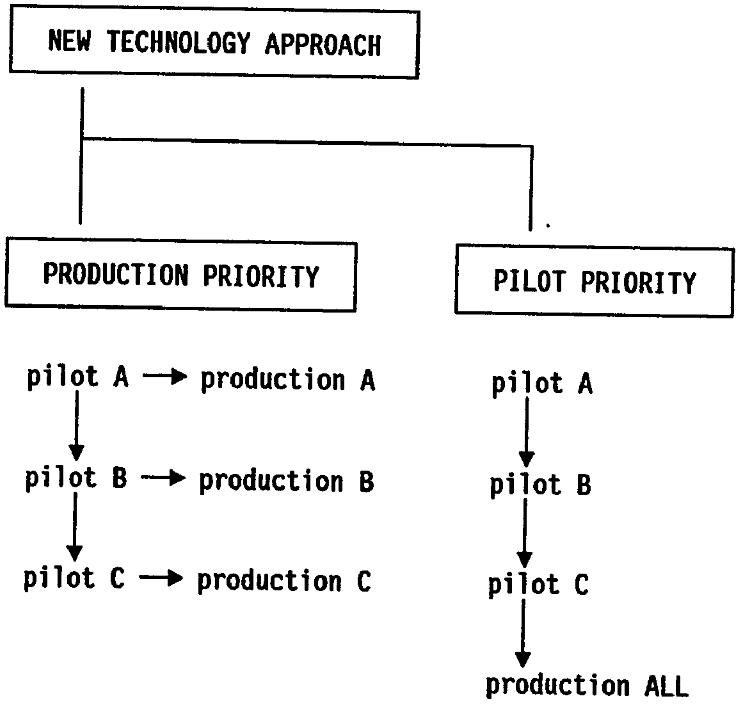

Figure 64. Technology approach considerations for tailoring the framework.

The production priority and pilot priority philosophy both require the customization of a solution for generative NC. The efficiency of this tailoring will be determined by an early and precise determination of which philosophy is driving a company.

Production priority is an approach to new technology that clarifies the contribution of that technology by pilot activity related to a specific production segment. The company's commitment for production to the segment covered by the pilot quickly follows. A second pilot embraces another segment of manufacturing processing. A successful pilot in this area produces another production implementation. The requirement here is to tailor the solution to each of these logical segments of the company's overall machining processing.

The *pilot priority* culture clarifies contribution to the business through a series of pilot applications before any production is committed. The production commitment would be a global decision at the overall business level. Tailoring a solution to this culture would involve the generative numerical control framework to be a larger server of production part programming functions. This approach seems to not take advantage of an evolutionary path available through the framework to grow a solution. The isolated pilots must be systematically integrated after the production commitment has been taken.

5.6.1 Tailoring the Solution

The requirement for tailoring the solution proposed by the generative numerical control part programming has been developed. Weeks and Dickinson [74] confirm the need to customize a system. They identify the need for an "open system" as an essential and fundamental component of an automated manufacturing system. The next question posed to this research effort is clear. Where in the framework is "tailoring the solution" available?

5.6.2 A Static Architecture

The framework of generative NC is developed from a set of *system components* and an *architecture*. These system components have been further segmented into *manufacturing data, manufacturing knowledge, numerical control database, CAD/CAM system services*, and *part programming functions.*

Tailoring the solution for generative numerical control is accomplished through the development of the framework without input or change to the architecture. The architecture is designed to be static. This architecture provides for tailoring and function evolution or growth. This statement is testimony to the architecture establishing an absolute contribution to industrial production part programming. The customization of the solution is reserved for interaction with the system components defined for generative NC. This requirement is also provided in the processing control externalized from the architecture proper.

5.6.3 A Dynamic Framework

The framework of generative numerical control is dynamic in the sense of providing a "tailoring the solution" capability to the built-time user. This user has a different role than the run-time user previously developed. Build-time user definition and skills requirements are developed as a related element of the generative NC framework research.

The framework incorporates a tailoring of the solution through interaction with the *system components*. Tailoring of the solution can require adding, deleting or modifying these fundamental parts of the generative NC framework. Figure 65 on page 164 uses the system topology perspective of these system components to overlay the effects of tailoring the solution. For presentation purposes the tailoring as shown in the figure represents a net growth available in each component of the framework.

The build-time user task is defined to tailor a generative NC system with specific data, knowledge, and processing. This task is focused on the tasks of application administration and application programming. These activities are most often found shared between two persons in a manufacturing organization. These users are typically referred to as the generative NC *Master-User* and *Programmer*. The post installation workload of an implementation of the generative numerical framework will be skewed toward the master-user.

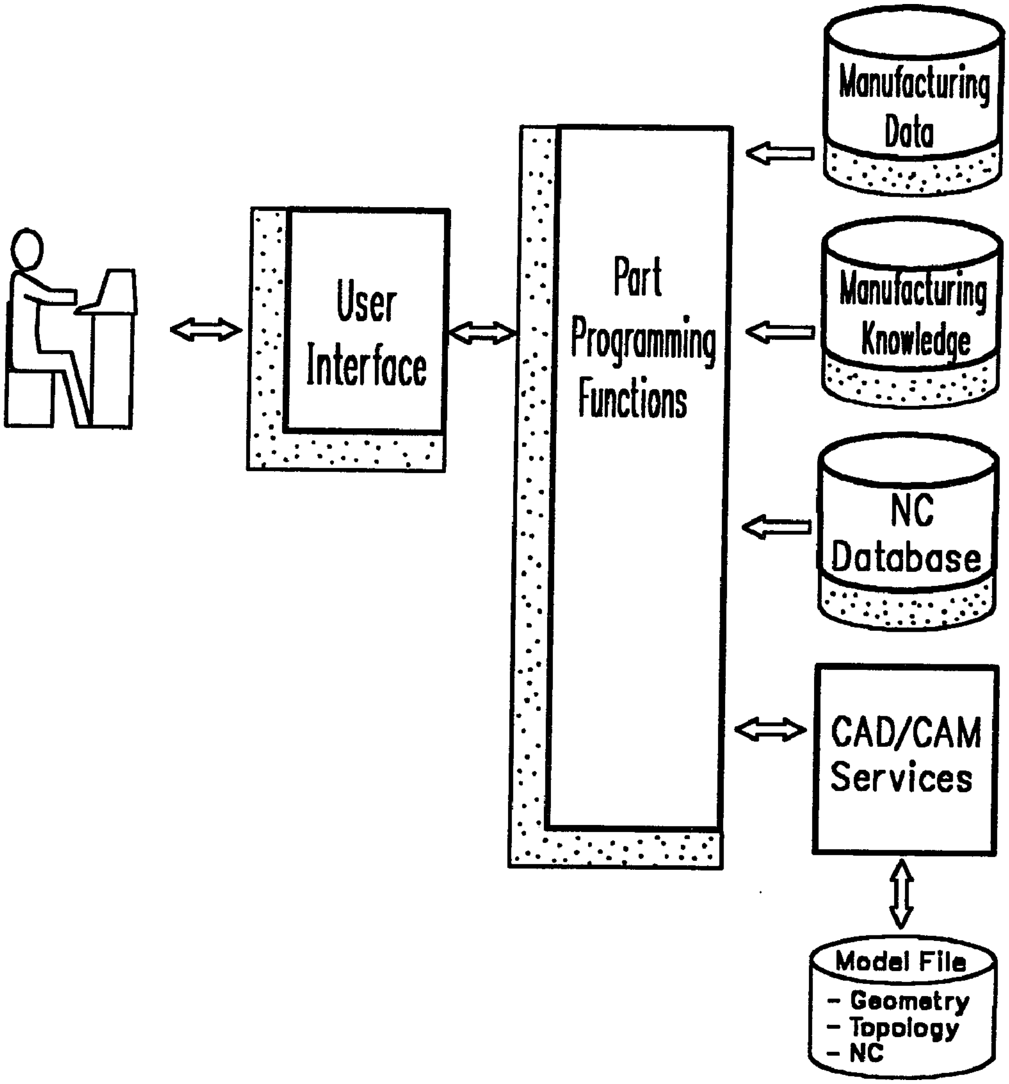

Figure 65. Framework dynamics through component tailoring.

5.6.3.1 Generative Numerical Control Master-User

This user's skills are employed in defining and maintaining an installation's generative NC methodology, data, knowledge, and processing. The master-user will possess skills as shown in Figure 66 to be employed in the customization on the generative NC solution. This definition is included here in the effort to identify the extent of customization available within the framework development.

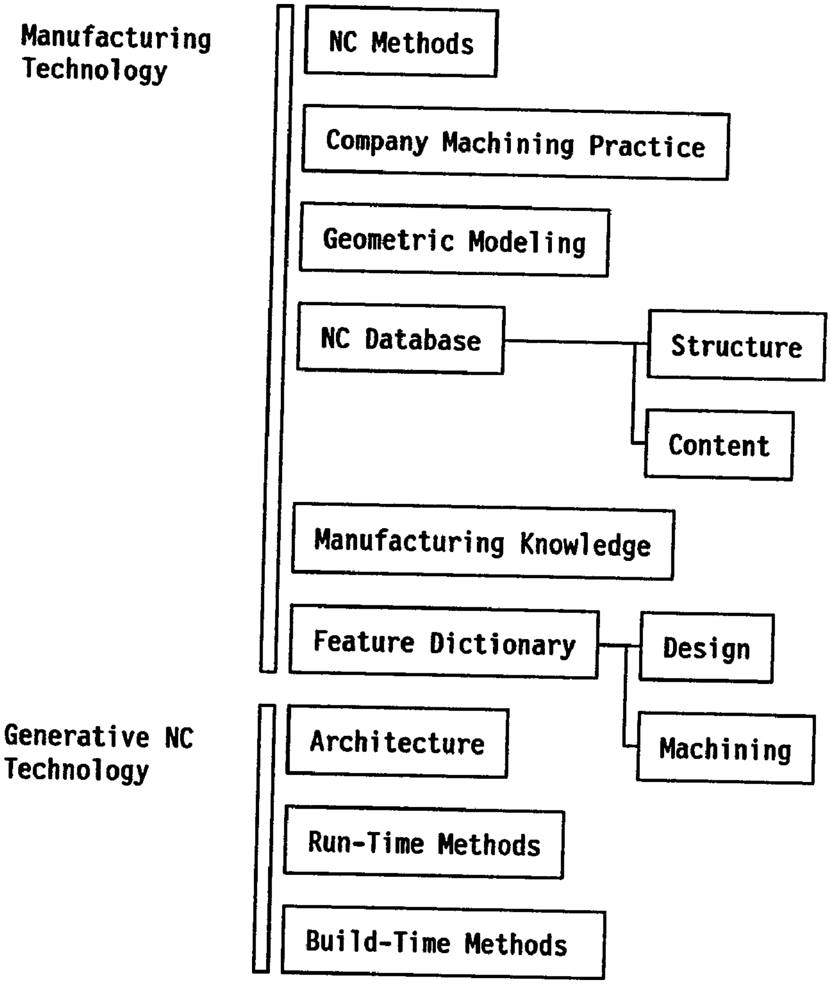

Figure 66. Generative NC master-user skill requirements.

5.6.3.2 Generative Numerical Control Programmer

This user's skills are employed in the programming requirements for installation and additional program tailoring of a generative numerical control system. The skills necessary for customization are mapped directly from the *system components* defined in the development of the framework. The *generative NC programmer* approaches the customization task through the "tools" available through each system component.

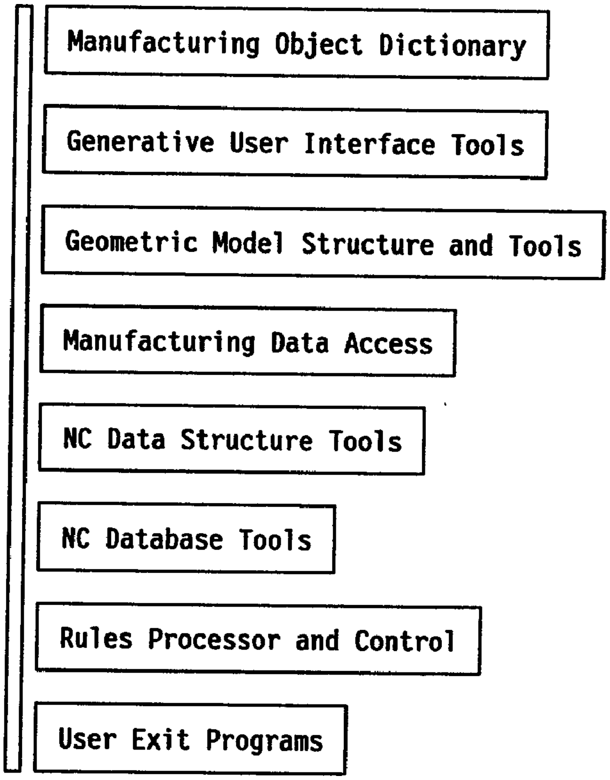

Figure 67. Generative NC programmer skill requirements.

6.0 Generative Part Programming Functions

This chapter of the research documentation formalizes a set of functions to complete the definition of the generative numerical control framework. This *system component* organizes the research results into the information essential to providing a solution for part programmers, NC engineers, and toolmakers. This system component of generative NC also extends the depth of this automated system framework. The effort expended in this area of the research was to identify a set of functions for an automated part programming system. This set of capabilities proposes a comprehensive and generative part programming solution.

The set of functions contained herein defines the detailed capability of a generative numerical control solution. This contribution to current research expands the knowledge of the automated system approaches documented in the literature. Ghosh et al [19] describes a compete system for two-dimensional cutting cells. The contribution from his work is defined from his "global space approach" with treatment of processing planning at the cell level. There is no consideration in his work for the functions that define the detailed automation necessary for his "generative process planning strategy." Weeks and Dickinson [74] make their contribution to the literature by proposing the essential and fundamental "concepts" for an automated production system that is focused to the manufacturing of sheet metal parts. The Weeks and Dickinson concepts do not define the functional level of such a system. Brown and Gyorog [4] propose their generative system for dimensional inspection planning. This work conceptually defines four "system components" and nine "activities." However, the definition of the system concludes with their definition of the activities in their "inspection planner procedure."

Houten [27] uses a very modular approach to his generative system. His presentation defines each module in this parallel processing formulation. Machining methods selection, tool selection and cutting conditions are most applicable to the research presented in this section of the dissertation. Of the three modules, the cutting conditions presentation provides the most insight into the detailed capabilities.

Let us return to the development of the :generative part programming functions after having considered the literature. The set of functions proposed herein furthers the current definition of an automated and generative numerical control part programming solution by one additional level. This set of functions will also serve as a definition to structure a comprehensive evaluation of any emerging implementations of generative NC part programming technology. This research will not approach the specification of the algorithms that will supply each of the defined function. This task is reserved for framework implementation.

The approach used to organize and present the results of this research regarding the generative part programming functions is to first reference the *processing modules* as developed in Chapter 5. These generative numerical control processes will be elaborated into the *manufacturing activities* that comprise them. Ultimately, the *generative part programming functions* will be defined through a structured decomposition of these manufacturing activities. This approach is identified in Figure 68.

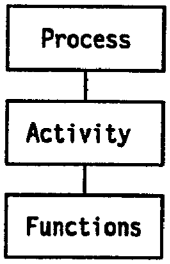

Figure 68. Research approach to generative part programming function definition.

However, this approach to the definition of the part programming functions for generative NC does not adhere to such a strict hierarchical categorization. This approach is modified in Figure 69. This expanded research approach indicates that a series of *activities* provide a better expression of each *generative process module* in terms of taxonomies of manufacturing solutions. These *activities* each retain an iterative option as represented the figure. The *functions* within each activity may also be classified as activity functions "A1" or shared functions "S1." This classification of functions will be of greater importance during an implementation phase. These considerations for implementation will provide efficient handling of shared functions through focused programming efforts. However, the task in this work is to communicate the functional requirements for each *generative processing module.*

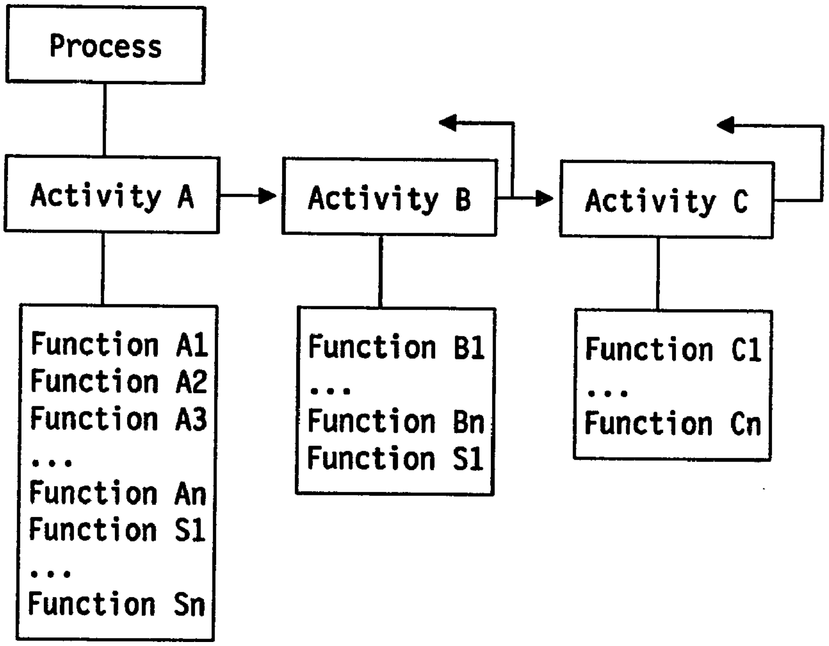

Figure 69. Expanded research approach.

The processing modules developed in the *architecture* are represented in Figure 70. The identification of the *activities* with the associated development of the included *functions* will be undertaken in the sequential order that the process modules appear in the architecture.

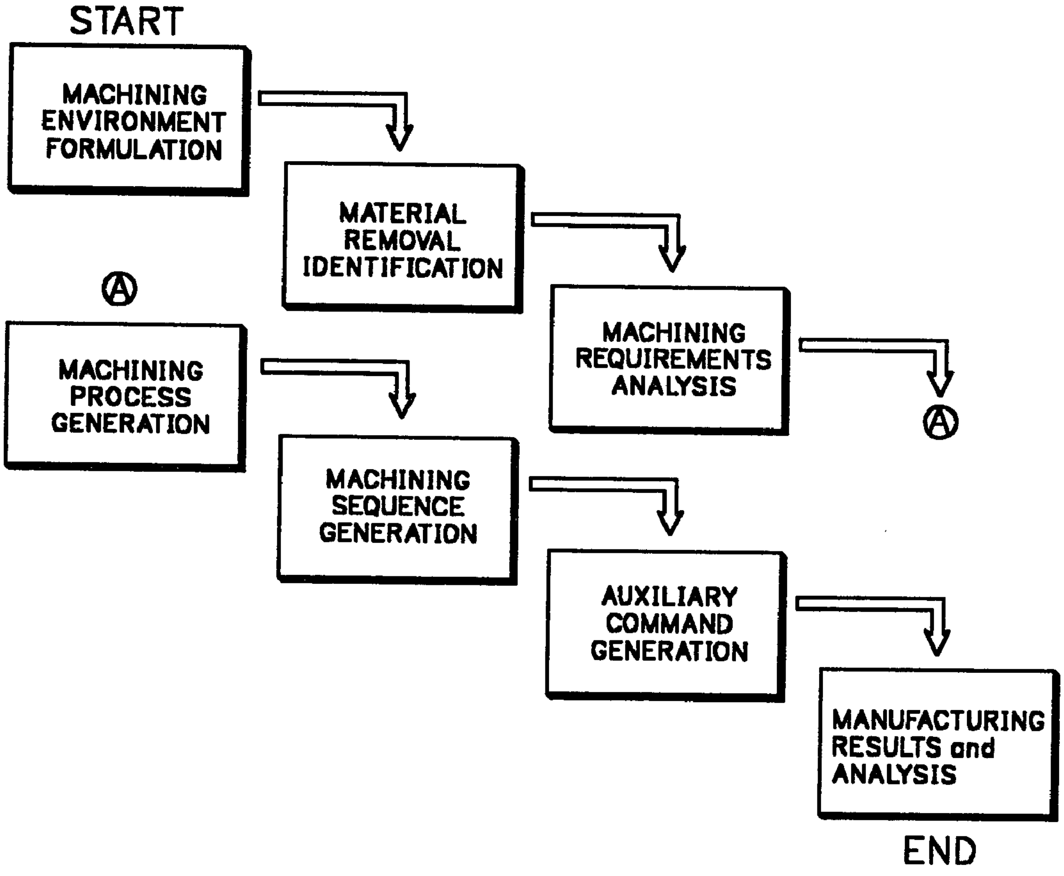

Figure 70. Generative NC processing modules.

6.1 Machining Environment Formulation

This process as developed for generative numerical control contains two activities. These activities are *Application Data Collection* and *NC Workspace Creation.*

6.1.1 Application Data Collection

This *activity* provides the initial user interface to the system and the point of entry or validation of default values of all data required from the user. This activity provides the user functions as presented in Figure 71 on page 172. These functions are available in the form of panels that employ the following interaction methods in dialogue:

- KEY information to the panel:
  - Part programming session output file name (default).
  - Setup dimensions for fixture on machine table.
- SELECT from system supplied alternatives data:
  - Model files for Part, Stock, and Fixture (optional)
  - Database window - desired *NC Database*
  - Know-how window - desired *Mfg. Data* and *Mfg. Knowledge*
- VERIFICATION of defaults values:
  - system processing - algorithm control parameters
  - system processing - application control parameters

The point should be made here that this activity is the unique and the only required user interface for this generative system. This is consistent with the goal to enable a fully automatic mode of operation. All required data is extracted from the user within this initial activity.

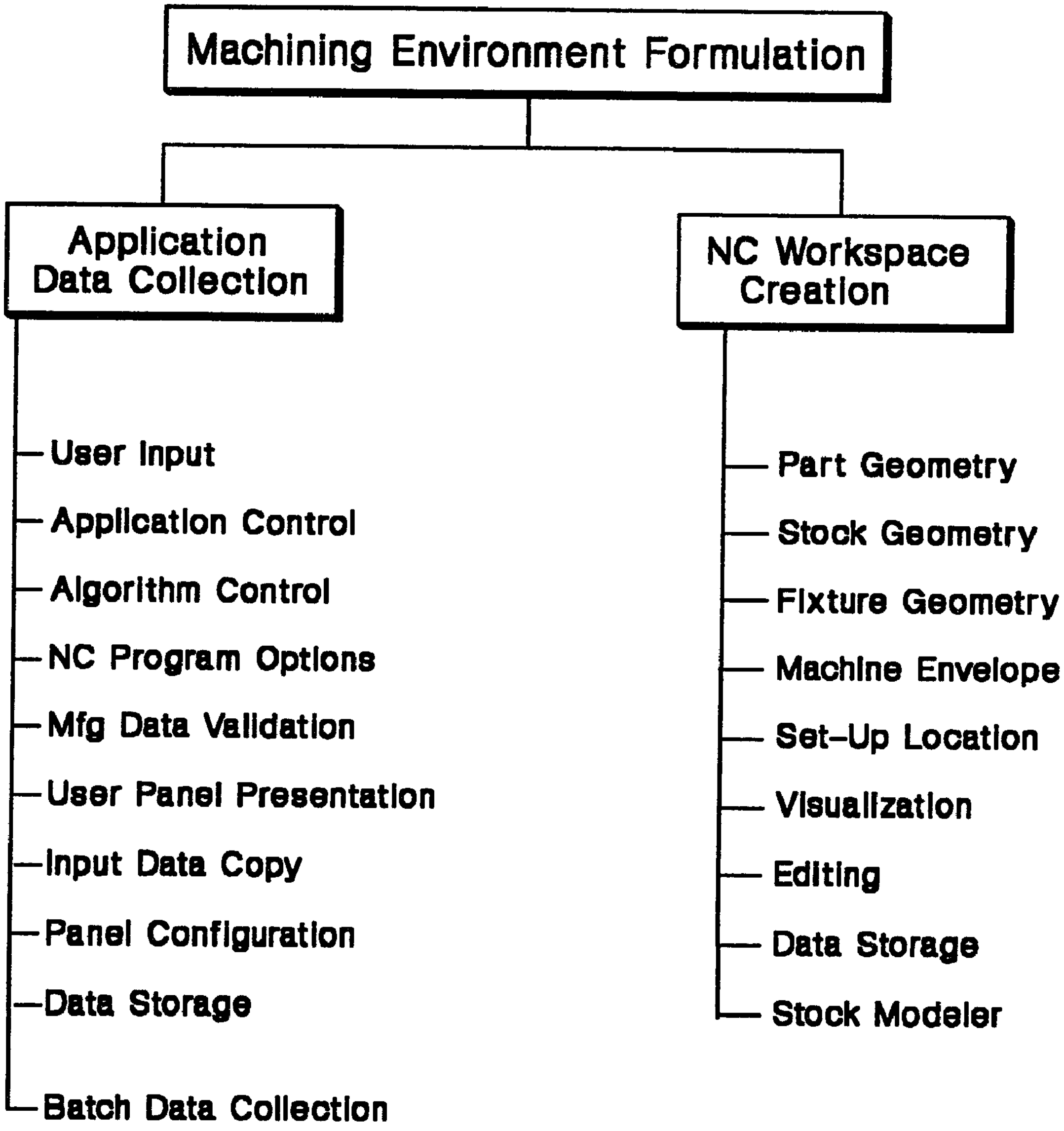

Figure 71. Machining Environment Formulation function composition.

This input can be alternatively defined to a declarative file. This file will use the same content and format as the interactive *Application Data Collection* activity and will be substituted to start execution of an implementation of the system. The reader should consult the *architecture controlling concepts* of *machining environment data* and *off-line data preparation* for a complete explanation.

The hierarchy of processing or algorithm use for an execution is set in this activity as defined for Application Data Collection. The functions that provide this capability are *application control* and *algorithm control.*

6.1.2 NC Workspace Creation

This *activity* provides the function to create the physical representation of the part programming workspace. This activity is accomplished by combining the part, stock, fixture, and machining envelope. Solids modeling is the mechanism employed for each model accessed. Part and stock models are mandatory within the framework established for generative numerical control. The stock may be staged in a model, model file, or constructed in a system session. The fixture model may also be available in a preplan file as required by the user. This NC workspace element is defined in whole or as a partial fixture definition. A partial definition would include only the geometry that is significant to operational path planning. This element may be implemented as a model containing the specific selection and placement of the locators, support, and clamping mechanisms. This element of the numerical control workspace is not mandatory. The fixture is included in the NC workspace as a element to enable comprehensive spatial analysis resulting in mathematically generated collision free toolpaths. The user must accept the responsibility of fixture placement if he optionally chooses not to specify a fixture model to be used in the generative part program creation.

The machining envelope is constructed from the machines *manufacturing database*. This function uses the machines "axis travel limits" and definition of one or more "dead zones" to construct the volume that is accessible by the face of the machine's spindle. This element is then assembled with part, stock, and fixture based on the user defined *setup dimensions* Additional NC workspace functions support the machining environment formulation requirements for *visualization, editing,* and *data storage* within the programmed system implementation.

The data collected and created in the *Machining Environment Formulation* process is written by the *CAD/CAM services* to the *part programming output file*. Processing errors found return *messages* to the generative user interface and are written to a *system error file*.

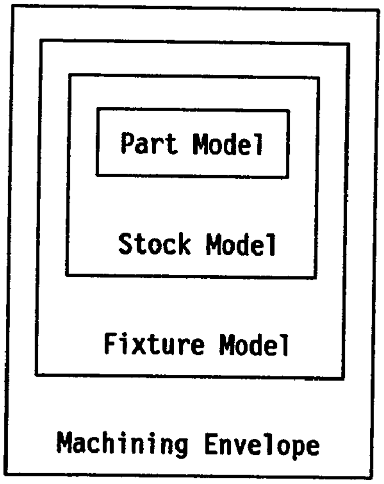

Figure 72. Conceptual schematic for NC Workspace Creation.

6.2 Material Removal Identification

This process is achieved in the framework for generative numerical control through the contribution of its five activities. *Feature Transfer, Part-In-Stock Features, Machining Accessibility, Machining Feature Topology* and *Planner's Window* are the activities that comprise the *Material Removal Identification* process. These activities are presented in Figure 73 on page 176 and Figure 74 on page 179.

6.2.1 Feature Transfer

This *activity* includes the functions that access the part specification in feature and geometric form. The initial transfer begins with access to the features data structure. The next step is to provide a transfer or mapping to installed machining feature objects. The part features definition must be mapped to a machining semantic with the result being an instance of a machining feature. This basic data format is exactly the same as the CAD/CAM feature. The basic data structure is given manufacturing application definition with the assignment of manufacturing contextual meaning. The associated attribute and tolerance data is transferred to the machining feature by respective function.

The features and tolerance handling implementation is modularized for migration to solids and feature based solids standards as they become production implementation ready [38]. The architecture developed for generative numerical control directs the majority of this burden to the host CAD/CAM System.

---

[38] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features - Volume 2," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995.

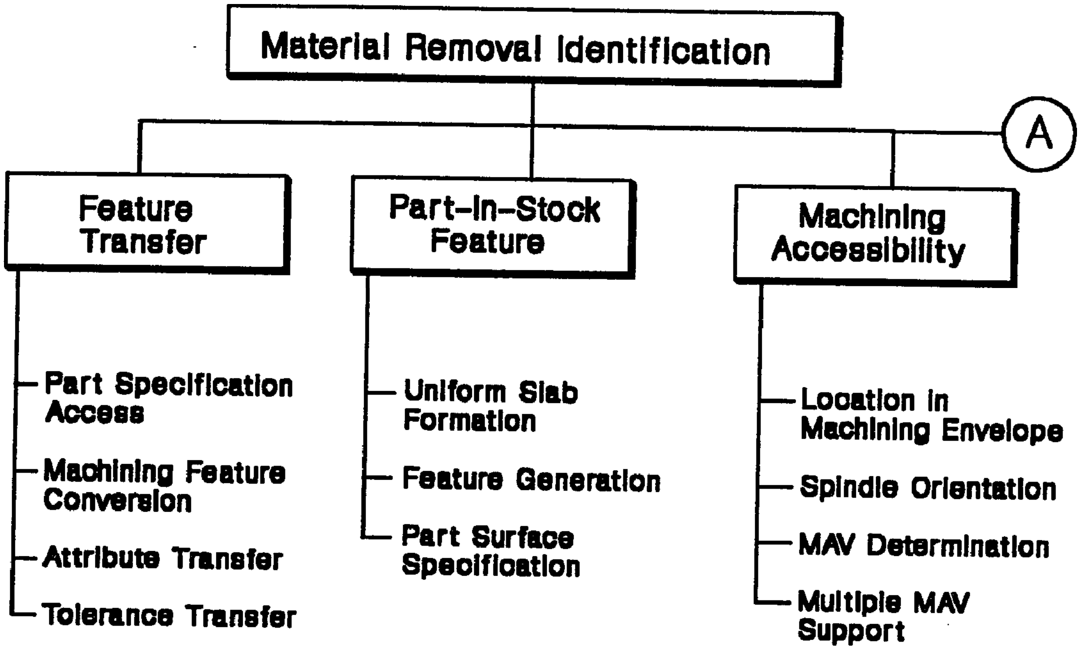

Figure 73. Material Removal Identification function composition - part 1.

This exposure is managed through the use of basic application programming interface callable subroutines. These routines provide a standard interface to the feature object definition and provide read, write, delete, and modify capability within an implementation of the framework. Evolution of the feature object and access to it can be managed by the CAD/CAM services and updating the basic application programming interface.

6.2.2 Part-in-Stock Features

This *activity* provides the functions to quantify the machining requirement outside of the part and inside of the stock. This volume is quantified by the Boolean intersection of the stock and the part solids. These functions include spatial enumeration techniques with emphasis on the formulation of uniform thickness slabs or planar face machining features [39]. The part surface specification is maintained with anticipation of roughing to finishing selective processing.

6.2.3 Machining Accessibility

This *activity* provides the initial application analysis as to the physical machinability of the part given the environment constraints as defined by the user.

- Machine table
- Setup information
- Machining envelope
- Fixture accessibility
- Spindle orientation
- Travel axis limits
- Table position limits

---

[39] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, p 61.

This analysis draws on the physical constraints of the machining envelope and the setup or location of the part on the machining table. The functions included in this activity are the analysis of all features on the part within the addressable volume of the machine spindle. This function is furthered with the analysis of spindle orientation. Some of these functions will increase in complexity and will be more applicable to the rotary axis capability of the given machine. Features are then reviewed to determine the possible spindle approach or approaches. This data is recorded as the possible machining approach vector(s) [40].

The machining accessibility activity is completed with a selection of a single machining approach vector. The application would provide basic manufacturing data to make a choice. Multiple MAV's can be supported throughout the application with alternative analysis being undertaken after more process information has been generated. This utility of the multiple machining approach vector function is determined by specific evaluation of the part programming requirements.

Multiple machining approach vectors can be used within the detailed plan formulation to provide alternatives of a more suitable process or efficiency of machining. The example here would be a choice to create a particular part surface with a toolpath trajectory using the side of the cutting tool or to rotate the part and engage the end of the cutting tool. This rotation would present an alternative machining approach vector to the spindle and typically provide more control and better results within the part program. However, auxiliary commands that would position the part through the rotary axis must be employed.

---

[40] Machining Approach Vector (MAV) is the direction defined on a machining feature which will be aligned with the machine spindle to generate a detailed machining plan.

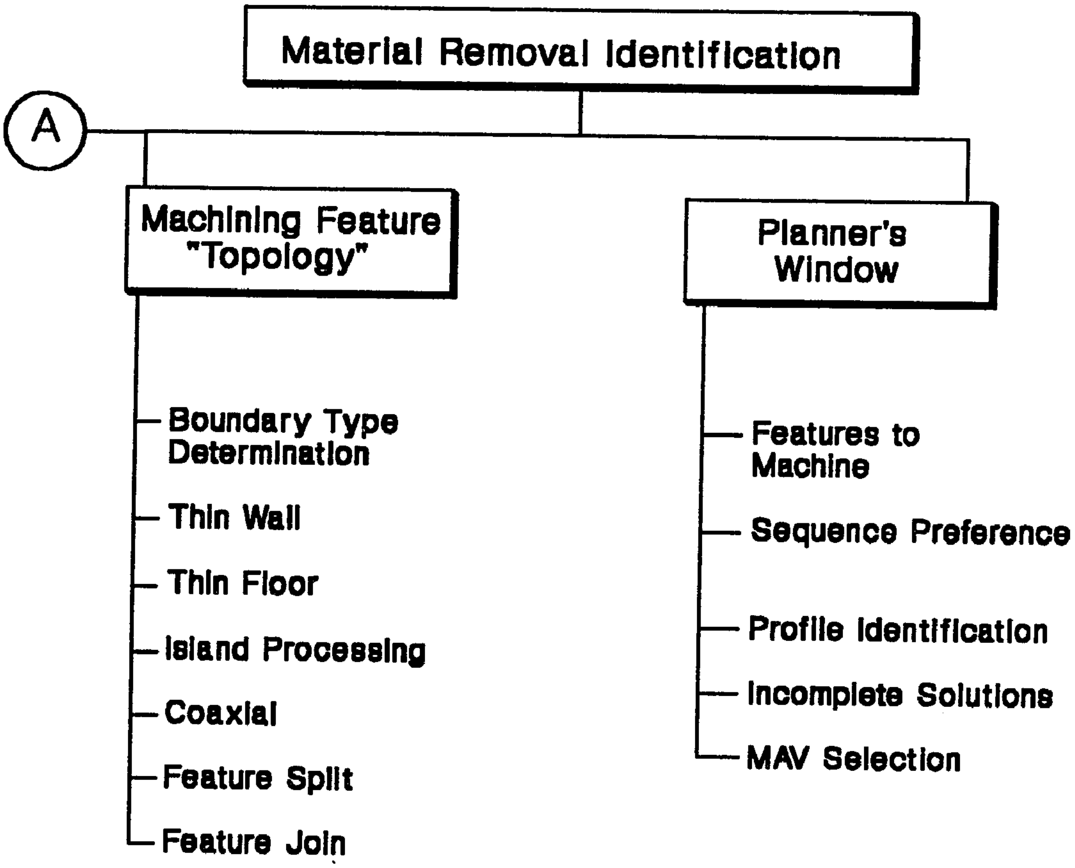

Figure 74. Material Removal Identification function composition - part 2.

6.2.4 Machining Feature Topology

This *activity* investigates the position of each feature with respect to the part. This activity also reviews the anatomy or structure of each machining feature and records the specific findings in program processable form.

These findings contribute to the definition of the machining feature object that is used for *Machining Process Generator* and *Machining Sequence Generator.*

These functions include a boundary type determination. This function provides information on which boundary elements are machinable [41] and not machinable [42]. Thin wall and thin floor conditions are determined through investigation of feature adjacency. Islands are determined by investigation of a positive solid included in material removal volume [43].

The functions of this activity also identify and record coaxial conditions that may be found within the machining feature set. The example here is the hole feature and the countersink feature. This information would be used within the detailed planner to efficiently generate a process for the multiple features at the level of machining capability provided by *manufacturing data and knowledge.*

This activity also provides the functions to interrogate each feature within the context of geometric machining efficiency. These functions are sensitive to the coupled machining operations available in the numerical control processor. The analysis done is to determine if a logical split or joining of the machining features would provide a more regular [44] material removal identification.

---

[41] Machinable Boundary Element is an attribute that can be attached to each element of the basic curve that is used to define a machining feature. This attribute designates the element as not creating a part surface and must be machined through.

[42] Not-Machinable Boundary Element is an attribute that can be attached to each element of the basic curve that is used to define a machining feature. This attribute designates the element as creating a part surface and must not be machined through.

[43] Material Removal Volume (MRV) is the volume of material to be removed by generative numerical control processes as defined by the geometric component of a machining feature.

[44] Regular Machining Removal Volumes would be constructed by feature split or join functions to create MRV's that more closely map to volumes that are addressed by the NC processor commands.

The execution of the algorithms that will need to be assembled to provide the functions as defined for Machining Feature Topology represent a significant portion of computing resources for the generative numerical control part programming overall process. Therefore it is important that this function be invoked as required for each part programming problem. The hierarchy of processing available will be a direct contributor to the processing efficiency and performance that is realized in this activity. This processing hierarchy is set in the Application Data Collection activity.

6.2.5 Planner's Window

The automated activities of *material removal identification* are then opened to the part programmer or NC engineer through the use of the planner's window. The planner's window provides an optional user interface where the user can supply additional point of use data to the functions that are quantifying a part's material removal requirements.

The planner has functions that allows him/her to deviate from the application philosophy of complete machining within the constrains outlined by the machining environment. The planner can identify individual machining features to machine or reserve for machining by a following part program. The planner will also need the function to optionally add any specific machining sequence preference he would want to see between two machining features. This information is recorded as precedence and will be used to augment the application generated precedence data that will be used to generate a machining sequence for the part operation.

The planner's window will also include the function to add profile machining feature selections. These features could be specified as both open and closed contours. These contours could typically be referenced as outside profiles [45].

The planner's window becomes complete with functions that support the investigation of application results accomplished in this aspect of the detail planning process. The incomplete transfer from design feature to machining feature is one such investigation. The working machining approach vector selected by the application can also be investigated and selected with this activity. This utilization of the planner's window would provide for user interactive extrapolation of the programmed capability that have been implemented to handle Material Removal Identification.

An initial implementation of this framework may elect to implement the planner's window to provide all of the functions of Material Removal Identification. This interactive solution could be chosen to postpone the detailed handling of part geometry and capability provided in the CAD/CAM system services. The author proposes that this should be a temporary solution of an implementation for generative numerical control.

---

[45] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, p 40.

6.3 Machining Requirements Analysis

This process transforms a set of qualified machining features into an ordered set of machining feature objects. These objects will be presented to *Machining Process Generation.* The activities that provide this transformation are *Machining Feature Classification* and *Feature Geometry Analysis.* These activities are represented in Figure 75 on page 185

6.3.1 Machining Feature Classification

This *activity* provides a classification of the machining features for more efficient detail planning. This classification provides a sorting capability of machining requirements prior to *Machining Process Generation* The functions that are included in this sorting related activity are machining feature type, relative size, machining approach vector, alternative machining approach vectors, and possible patterns.

Patterns are found by matching attributes with distinctive processing for both axial and milling pattern cases. The function of patterns is added through the aspects of this research that are concerned with software processing efficiency. Zhang et al [78] identify the "pattern" contribution to a sequence planner. Application is made for patterns in this research to also improve the efficiency of process generation. The axial pattern functions classify hole type machining features that have the same machining specifications. This criteria is typically applied at a level of "allowance" that is provided by user control. Pattern functions would be important to classify singular or groups of milling type machining features. The milling pattern recognition presents a different level of problem to the classification software. Feature structure patterns will reduce the milling pattern classification to that as found with axial features.

The current feature definition activity of PDES [67] appears to be concentrating on "regular" geometric patterns. This approach is also found in the CAM-I [5] work. Typically, a "compound feature" is applied to this situation. Additional research on the topic of representing a random placement of "like" features in a machining semantic would contribute to feature efforts. This work would also relieve each application from internal definition and management of this feature occurrence.

6.3.2 Feature Geometry Analysis

This *activity* provides functions to analyze the detailed geometry of each machining feature. The functions calculate the geometric attributes that constrain the detailed plan options available to each machining feature. These attributes are:

- **Logical Machining Direction** is an element (line) described in the CAD/CAM system that specifies the direction of tool motion for "back and forth" toolpaths.
- **Machining Requirement Minimum Radius** is the minimum radius-of-curvature in a machining feature boundary that create a constraint on the finishing *tool requirement.*
- **Minimum Tool Channel Width** is a dimension found by analyzing the machining feature boundaries and inner domains boundaries that identifies the maximum diameter constraint on the finishing *tool requirement.*
- **Maximum Tool Channel Width** is a dimension found by analyzing the machining feature boundaries and inner domains boundaries that identifies the maximum diameter constraint on the roughing *tool requirement.*

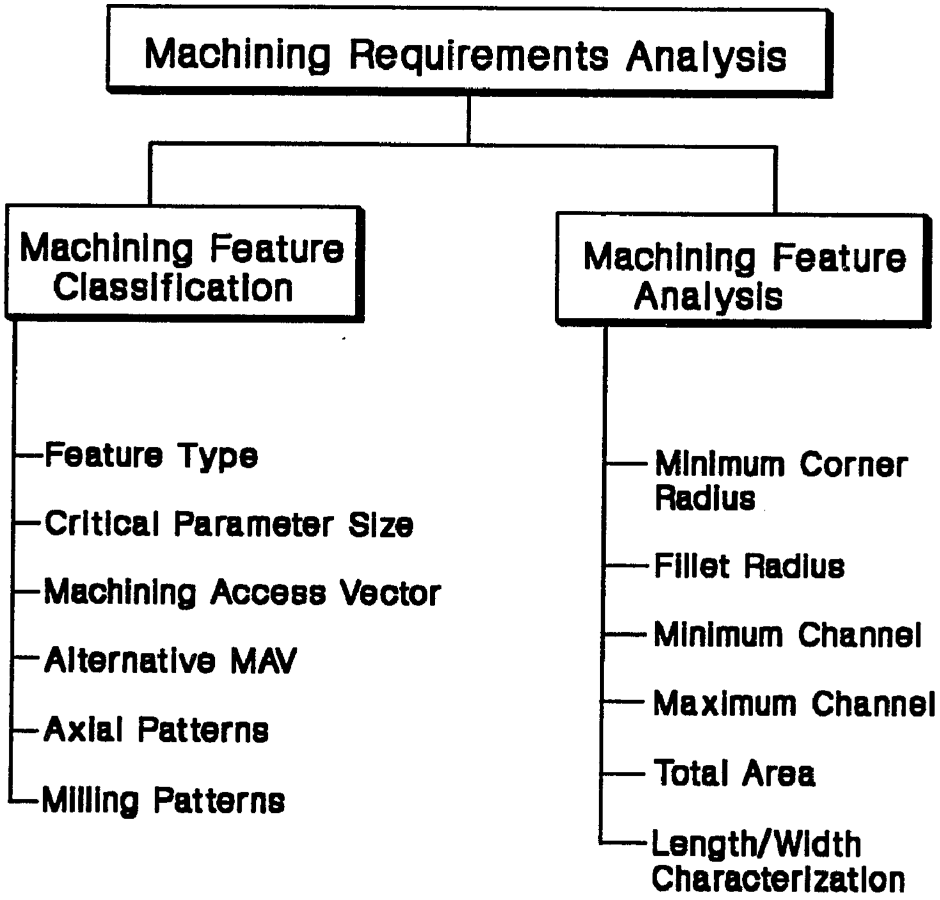

Figure 75. Machining Requirements Analysis function composition.

- **Machining Requirement Area** is the total area of the machining requirement in the machining feature. This attribute is used in *roughing strategy* considerations.
- **Machining Requirement Profile Length** is the total length of all the boundary elements of the machining requirement in the machining feature.

The *Feature Geometry Analysis* functions are seeded with an extraction of the basic curve of each machining feature. This curve is typically discretized for numerical control machining purposes. That is, the curve is decomposed into a series of linear and circular segments in accordance with a maximum allowable deviation from the original curve. This discretization is consistent with the mechanical motion control available to machine tools. The data that has been created to qualify a machining feature is now employed within the functions of *Feature Geometry Analysis.* The *boundary elements types* and *machining approach vector* would be called by these functions and used in these calculations to provide Feature Geometry Analysis data. This data completes the machining feature object definition that will be available to *Machining Process Generation.*

6.4 Machining Process Generation

This process is defined by the development of functions for three activities. These activities are *Machining Operations, Cutting Tool Selection,* and *Machining Operation* Area. Each machining feature object is brought to this process in the order in which it is available from *Machining Feature Classification.* Each machining feature engages all activities in Figure 76 on page 189 before the next machining feature is selected for Machining Process Generation.

6.4.1 Machining Operations

The functions included in this *activity* are assembled to provide an unambiguous definition of a machining process made up of a sequential assembly of machining operation(s). This activity is the generative heart of this numerical control part programming technology. The machining operations are generated from very unique and discrete data that have no inherent machining process form. This data is available in first principal form. The sum total of this data as generated for a machining feature establishes the contextual meaning of each *machining operation* within a *machining process.* This approach should be noted in contrast with the many proposals from the literature. Arikan and Totuk [1] have selected machining process at design time. This is done through their "design by machining operations" system definition. Gu and Zhang [23] also imply a direct link between machining features and the associated operation. Although the focus of their work is object definitions for process planning, this approach can be viewed as a limitation when transferred to the operational planning and NC part program generation problem. The strong link between *machining feature* and the *machining process* is also apparent in both the Generative Machining [10,65] and Manufacturing Analyst [10] approaches. Khoshnevis and Tan [38] indicate a variation of this approach by linking the feature type to a "basic group of processes."

In contrast with these approaches in the literature, Houten [27] confirms this author's contention that a more flexible approach to machining process generation is to build from "elemental machining know-how" as applied to a specific machining feature instance. This concept itself can clarify the variation in the computer aided process planning problem approaches with that of operational planning and NC part programming generation. The machining process "selection" problem is transformed to a *machining process generation* when a tight coupling of the NC processor capabilities is introduced to the solution. This approach allows for the optimization of processes at their time of generation. The most flexibility to operational planning is then achievable. This is opposed to a somewhat disjoint handling of "optimization" by Bala and Chang [2] and also by Ghosh et al [19]. The tight coupling of the "NC processor" with *machining process generation* is clearly a different approach from the work of Arikan and Totuk [1] This separate handling for the "NC processor" is also identified in the literature by Gu and Zhang [21] Once again these examples cited in the literature identify how the process planning system "conceptualized" a link to NC part program creation. An operational planning approach with tight coupling of the target *NC processor system component* offers greater efficiency and optimization in the generated part programs.

The functions of *Machining Operation* employ a distinct process flow for axial and milling type features. The process flows are unique and structured specifically for the technological differences of the associated machining processes. The axial and milling distinct process flows are supported by the function algorithms that differentiate the machining process basic dissimilarities. The resultant implementation can be two processing flows and sets of shared functions that efficiently address the uniqueness of the general holemaking and milling processes. This type of process flow also capitalizes on the underlying similarities with a shared set of functions for both the axial and milling processing.

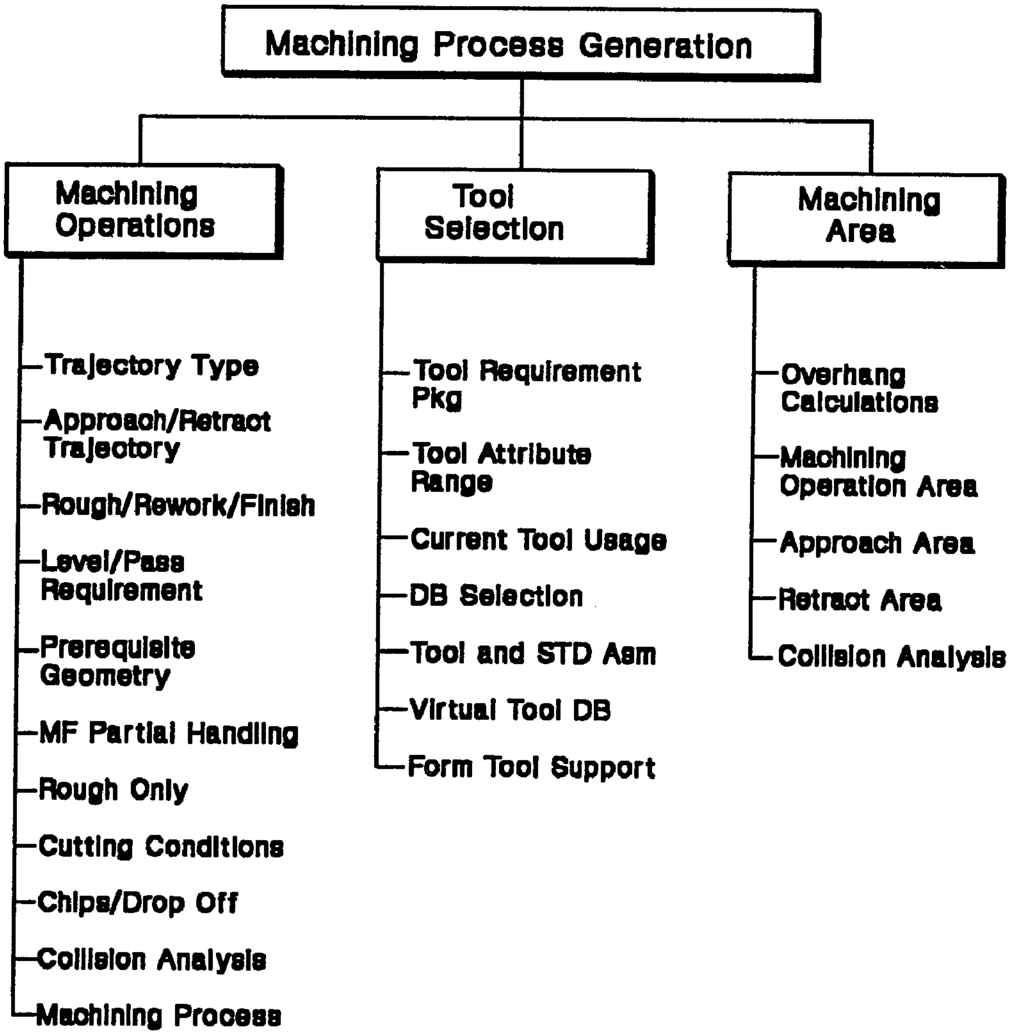

Figure 76. Machining Process Generation function composition.

Trajectory type and approach/retract trajectories are an important foundation of many machining operations. These are generated from a finite list of capability available in the *CAD/CAM services NC processor* and supported in the machine shop. These trajectories are applied to the *material removal volume* to determine the requirements for multiple machining level [46] or machining pass [47] control. Sensitivity to geometry conditions or prerequisite geometry forms enables an optimal application of cutting conditions data generated for each selected trajectory. These cutting conditions are computed knowing the exact amount of material planned for the cutting tool and the *dynamically modeled workpiece.* The computations are based on rate-of-material removal formulae.

The machining environment formulated for a particular part program will require additional functions to support *partial machining handling* of certain machining features. This situation will cause a deviation from the system philosophy of "complete machining." This deviation may be due to lack of manufacturing data and knowledge implemented in the system or simply an *user preference.* User preference can best be exemplified here through a "rough only" instruction given during *machining environment formulation* Similar functions and reasoning is applied for handling workpiece portion drop-off as a deviation from a reduction-to-chips machining philosophy.

The machining operation is then completed through functions that provide handling for *machining stages* within the machining operation as bundled by the selected NC processor. *Technology based machining operations* can be consulted for the details of machining stage bundling. These functions generate the operation parameters for roughing, semi-finishing or rework, and finishing.

---

[46] Machining Level is referred to from the axial perspective of the cutting tool used in a machining operation. A level would be defined by an allowable axial depth-of-cut.

[47] Machining Pass is referred to from the radial perspective of the cutting tool used in a machining operation. A pass would be defined by an allowable radial depth-of-cut or step over amount.

6.4.2 Tool Selection

This *activity* includes the functions that support the tool selection portion of Machining Process Generation. These functions package a machining operation's *tool requirement.* This requirement is prepared with possible tool attribute ranges. These ranges are formulated with a specific minimum and maximum values that frame the preferred attribute value. This "range concept" more closely balances the internal program handling of the data with the physical and technical bounding on a machining process. The available tool inventory is also a factor that promotes cutter selection by individual attribute ranges with preferred values.

The functions of tool selection include a review of the current tools already selected in a part program. This review has the objective function of providing an existing tool that satisfies the new tool requirement. If a current tool does not satisfy the requirement, the activity proceeds with an additional call to the data base. The tool requirement is always saved for *process analysis, error diagnostics,* and *NC database content management.* Tool selection will provide functions to select the individual tool components or a standard tool assembly [48]. *Special tools* configuration is supported by a virtual database tool [49] entry. The form tool identification and utilization within an NC processor is the point of further research. The form tool has geometrical attributes that do not conform to a class definition as defined for the *NC Database.* Trajectory point-of-contact identification and communication with the toolpaths calculator is a topic not addressed by this research.

---

[48] Standard Tool Assembly is defined in the manufacturing database as a specific set of components (cutter and holder) with a defined set length. This standard tool assembly is available from a database query that would normally pull only the tool component description.

[49] Virtual Database Tool is a tool with attributes defined within a part programming session by the user. This tool conforms in class definition and attribute set to a current entry to the manufacturing database. This capability is typically employed when no tool can be found and the user accepts responsibility for cutter grind to the specifications used for toolpaths calculation.

6.4.3 Machining Operation Area

This *activity* provides the functions to identify the specific geometry that a cutting tool will engage based on the machining operation defined. These functions provide their service to the system through a close coupling with data available from *CAD/CAM services*. Utilization of these mathematical services as available from the *NC processor* will produce consistent results for all the machining operations employed with this type of application. Consistency is achieved between the toolpath calculator's "boundary geometry" computation and *machining area*. The functions that are provided will compute the machining area with contributions made by tool placement for complete machining. Additional functions compute the *tool approach area* and the *tool retract area*. The *machining operation area* is generated by performing a Boolean addition on each of the three contours (machining area, tool approach area and tool retract area) to form a singular contour.

The dynamically modeled workpiece can be built using extruded contours at the machining operation level. These extrusions are formed as translational sweeping primitives [50] with the "base set" being the contour defining the *machining operation area* and the "sweep direction" being the *machining approach vector*. The solution works well for the domain of prismatic part machining as specified for this research. This solution builds the *dynamically modeled workpiece* to the material removal precision available from the *NC processor*. The translational sweeping primitives approach also simplifies the proposal by Wang [73]. Wang employs "tool swept volumes" to build his "intermediate part geometry." This approach can be computationally more expensive as it includes continuous certification of the NC processor output that is not required by prismatic parts machining operations.

---

[50] Mantyla, M., An Introduction to Solid Modeling Rockville, Maryland: Computer Science Press, 1988, pp. 96-97.

6.5 Machining Sequence Generation

This process is accomplished through the functions available in three activities as shown in Figure 77 on page 194. *Geometry Access, Operation Sequence* and *Tool Assembly Selection* are these activities. This process generates the sequence in which machining operation will be approached to manufacture the part. Tool assemblies are also selected within the activities defined for this process.

6.5.1 Geometry Access

This *activity* provides the functions to place each machining operation for each machining feature in the workpiece and to determine the geometric accessibility by the machine spindle. *Geometry Access* considers the context of each machining operation with respect to the final part surfaces and stock material. A resultant network of *geometric precedence links* are generated that will define the geometric access of each machining operation. These functions are gathering additional information to be used in the sequence activity. Vandenbrande and Requicha [72] introduce "accessibility" in their work toward feature recognition. The same concept applies in the geometric preparations for the sequencing problem. Delbressine et al [15] addresses part geometry through their notion of "precedence based on reachability." This setup planning technique focuses on the spindle reachibility to part features.

The processing available in the functions for axial and milling machining operations are unique for this activity in machining operation sequencing. The nesting of machining operations is taken advantage of here to reduce the number of calculations necessary to determine the overall geometry precedence of the part programming problem.

The workpiece is managed to ensure that part gouging or excessive stock engagement is not presented to the machine spindle. The *geometric access* activity is relying on the *machining details* as specified in the fully qualified *machining operations* that have been generated for each machining feature. Collision analysis is the basis of formulating the geometry access within *Machining Sequence Generation.*

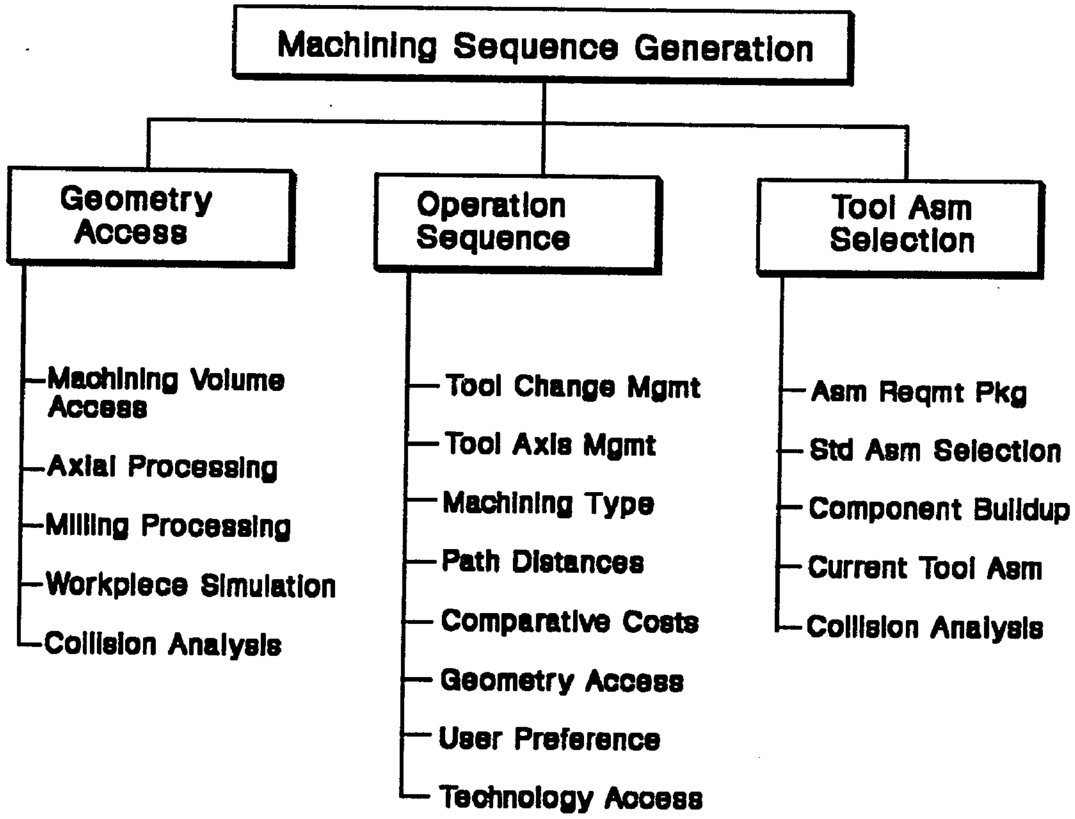

Figure 77. Machining Sequence Generation function composition.

6.5.2 Operation Sequence

The functions that are included in this *activity* are formulated to consider *geometry access or precedence.* Zhang et al [78] also address the concept of geometry access in the machining operation sequencing problem. However, their approach considers the machining feature and treats all machining operations within that feature as a "machining macro." The approach proposed by this author is to treat every *machining operation* within each *machining feature* as an object for the sequence problem. This proposal seems to provide a more pragmatic approach when each machining operation emulates the concepts of *technology based machining operation* and has a unique tool. In this approach the machining feature is primarily made available as a "macro view" of the included machining operations. This permits a logical simplification of the geometric computation associated with this activity. The functions of this activity also consider the *user precedence* that was optionally added between two machining features. Also, the *technological precedence* that has been generated as part of *machining process generation* will be the key information handled in the functions that comprise operation sequence.

Precedence (geometry, user, and process) has only provided bounding for part of the operation sequence activity. Shop practice and methods will be supported with functions to provide additional guidance to complete a machining sequence. Zhang et al [78] also introduce "good machining practice" to their solution. Their approach supports the formalization in this research of shop practice and methods know-how in the sequencing solution. Tool change management and tool axis management are functions that provide constraints on the total number of respective changes undertaken by a particular machining sequence. The function to handle the machining type will be based on shop practice.

Shop practice can be exemplified as a rule for drill and tap operations to be performed last in the sequence. This may be desirable from one machine shop's perspective to promote minimum build up of machining debris in threaded holes.

The *operation sequencing* functions are then approached with a quantitative perspective with calculations performed to generate additional sequence related data. This quantitative approach is typically undertaken to differentiate between several next candidates when all the previous constraints have been applied. Path distance to the next candidate is one such function. The use of this function will manage the overall length of the path specified in the final part program.

A comparative costs function will assign costs to next candidates in terms of the amount of time necessary to execute all required program elements. A typical candidate may require a tool change and a tool axis change (table rotation) after some additional path movement elements are specified. Each program element is assigned a cost based on implemented *manufacturing data* to quantify the total cost to the next candidate in the machining operation sequence. The candidate with the least total cost is used as the next sequence element. The reader can consult the *architecture controlling concept* of *comparative cost sequencing* for a complete explanation.

6.5.3 Tool Assembly Selection

This *activity* includes the functions that support the identification of the tool assembly requirement based on the machining sequence generated for the part programming problem. Sequenced machining operations of a part provide the information necessary to generate a tool assembly requirement. The sequence information has been used to build a solid model of the manufacturing plan using Destructive Solids Geometry (DSG) technique and the sweep primitives for each machining operation. The technique is used in a goal achievement manner to validate complete machining of the part beginning with the stock definition. Vandenbrande and Requicha [72] use this DSG technique in a similar manner with feature primitives to validate complete featurization of the part. The *sequence model* provides stock and part surface data about the in-process workpiece. This data is used to extract tool set length requirements, holder collision constraints, and tool flute length modification requirements.

The *tool assembly requirement* is packaged after formulation of the set length data. This data is calculated in context of the workpiece configuration at each entry in the machining sequence. Physical connectivity at the tool assembly to the machine spindle is managed with information derived from the machines catalog of the *NC Database*. A tool assembly can be selected from the database that contains both standard tool assemblies and tool assembly components. The components will be *dynamically built* using the data available regarding connectivity of each shank-to-pocket physical attachment. A given tool crib for the machine shop may be organized to support standard tool assemblies, build assemblies from components, or both philosophies. The functions provide for the *Tool Assembly Selection* activity to support this manufacturing technology variability of tool crib organization.

The functions of this activity include a review of current tool assemblies that have been previously defined for this part. This review may determine additional utilization by the *tool assembly requirement*. Upon finding no similarities, the requirement will be passed to the database to select or construct a new assembly that conforms to each attribute within the tool assembly requirement. Iterative calls to this function would be made with additional information on existing collisions caused by a database entry. The candidate list is refined with each iterative call to specify one tool assembly for a given machining operation. The selection of a tool assembly validates a candidate machining operation and completes its processing by the *Machining Operation Sequence* activities.

6.6 Auxiliary Command Generation

This process is comprised of two activities. *Tool Transition Elements* and *Auxiliary Commands* provide the functions to this process. This module is used to complete the part program by adding the toolpath elements that are required between each sequenced machining operation. The auxiliary commands required in the part program are also added by this process. These commands are provided to the part program in the specific machine tool post processor syntax. These activities are presented in Figure 78 on page 199. The *architecture controlling concept* of *machining state variable controller processor* can be consulted for the detailed treatment of the "control mechanism" for this activity.

6.6.1 Tool Transition Elements

This *activity* provides functions to complete the automatic path planning for elements of the part program in support of the planned and *sequenced machining operations*. These program transition elements are required to move the tool in a collision free path in accordance with the axis control data of the selected machine. The path elements are program generated and based on "current" location and the "next" location. Path control data is read from the machines *NC database* and *application control* inputs. These *transition path elements* are added to each part program to control spindle movement for the commands that specify:

- Program Start to First Machining Operation,
- Path Elements Between Two Machining Operations,
- Path Elements To/From Tool Change,
- Path Elements To Support Tool Axis Change (ROTABLE), and
- Last Machining Operation to Program End.

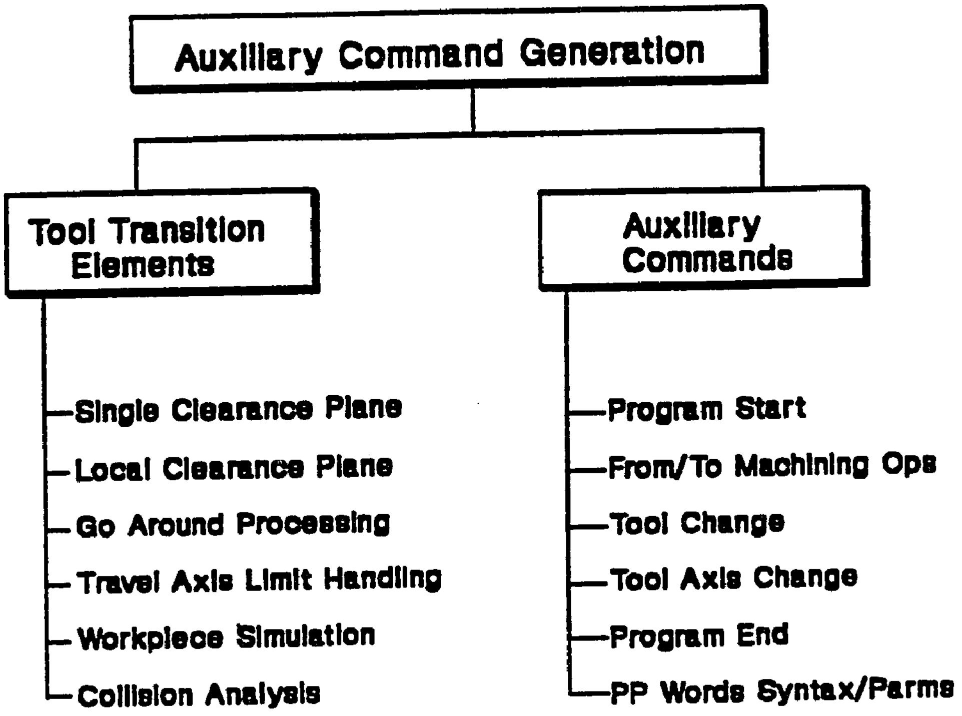

Figure 78. Auxiliary Command Generation function composition.

The functions that comprise this activity are supported by a *transition element processor* that is invoked with control of multiple levels-of-processing. These processing levels provide the functions of a single clearance or "safe plane" for the motion of these transition paths. The activity also handles local or multiple clearance planes for "go over" transition philosophy processing. The highest level of function will use "go around" techniques when a path over obstacles is not possible in the specific machining operation. Constraints that require this processing level could be part size, placement in the available machining envelope, machine axis travel limits, tool assembly length, or fixture components. These functions all make use of system functions for *machine axis travel limit processor*, *dynamic workpiece* configuration at each *sequenced machining operation*, and the *collision detection with analysis processor*.

6.6.2 Auxiliary Commands

A part will need to be completed by adding auxiliary commands to the part program. This *activity* completes the numerical control command stream that a specified machine's controller is expecting when downloading NCdata. The required commands can be in the form of sets of syntax with parameters that must be evaluated and placed in the program. These commands create explicit commands for:

- Program Start and End Commands,
- Tool Changes
- Rotary Table Changes,
- Cycle Statements, and
- Tool Axis Changes.

6.7 Manufacturing Results and Analysis

The functions of generative NC part programming are completed in that all the operational planning data and part program elements required by a specific *CAD/CAM services NC processor* are generated. However, this data has been generated by automated programs that implement the functions defined for the each process module. Access to this data as the manufacturing results of the system execution and analysis of these results are system definition "boiler plate issues." Therefore, the generative NC process concludes with the activities of *Manufacturing Results* and *Manufacturing Results Analysis*. These activities introduce the interactive functions to access the results of the generative NC process. This generative system user interface allows the run-time user to query the in-process information that was system computed and and mathematically verified throughout the process modules. These activities are presented in Figure 80 on page 203.

6.7.1 Manufacturing Results

This *activity* is comprised of the functions that provide output from an implementation of generative numerical control. These functions provide the *NC commands* that result from the detailed operational planning. The resultant *workpiece geometry* is generated by the functions supporting this resultant NC command manufacturing data. The activity also provides functions to support a *tool list processor* for the part program.

The *routing processor* functions will scan the manufacturing data available and assemble it according to the implementation defined user requirements. All these output are to be available in file and print formats.

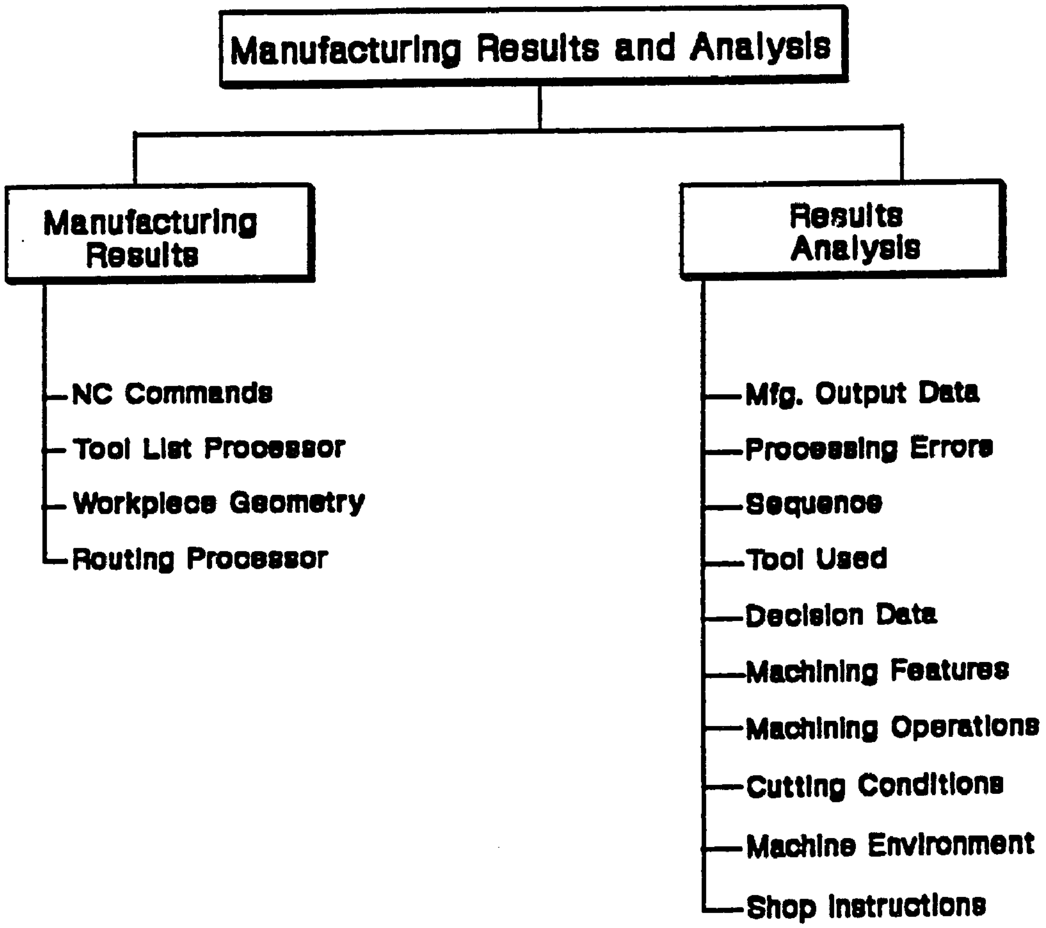

Figure 80. Manufacturing Results and Analysis function composition.

This workpiece is constructed by the application as each machining operation is planned and removed from the original stock. The workpiece can be retrieved at a state following any of the machining operations. The form of the workpiece model is consistent with the stock definition so that it can be defined as the stock object for subsequent processing requirements on this geometry.

6.7.2 Manufacturing Results Analysis

The results of the generative numerical control system are interactively opened by the user with the functions provided in the *Manufacturing Results Analysis* activity. This activity represents the final set of functions as defined in the framework. The data generated during an execution of the system can be found through the *generative system user interface* built and placed for this purpose.

The functions that supports this activity include access to the manufacturing results of the automated system execution. However, this notion goes much further based on the *system components* and *architecture* that comprise the framework for generative numerical control. The functions are expanded to include system *processing errors* reporting and system diagnostics. This activity grows in importance in the movement from interactive graphics programming systems to generative part programming systems. The level of automation available requires that all aspects of the detailed operational planning process be available to the user. The detailed operational plan is interrogated by functions to identify *sequence information, tool where used, machining features*, and *machining operations*. Detailed analysis of the *machining operation* provides the user with low-level manufacturing control data that generates the CAD/CAM services NC processor commands.

7.0 The Example Part

This chapter is included in the presentation of this research in an effort to highlight the contribution made by the framework for generative numerical control part programming as applied to a typical part configuration. The part geometry targeted by this example was developed as an activity to define a benchmark for NC part programming. The results of this activity are found in the literature by Aber and Glemser [52]. This reference contends that a geometric configuration be formulated whose purpose is the "creation of a demonstration part that is used with a developing numerical control application." This work includes the design, manufacturing planning, and fabrication requirements of a set of geometry that is proposed to be a demonstration for a developing numerical control application. Similar efforts have organized and developed a set of geometry to be used for the demonstration and benchmark examples of CAD based numerical control applications. The CEFE Arbeitsgruppe 23 [53] effort has developed such a reference geometry to be used for this purpose by European manufacturers and system development efforts. A review of their part's machining requirements can be categorized as a series of differentiated hole making and pocket machining exercises. The CAM-I efforts also use reference geometry. One of these geometries ANC TEST 101-M[54] from their Advanced Numerical Control Program (ANC) incorporates a similar area of features as the example part.

---

[52] Aber, D.M., and Glemser, R.G., "The Design, Manufacturing Planning, and Fabrication Requirements of a Part for the Demonstration of a Developing Numerical Control Application," Joint Report Series: Manufacturing Systems Engineering Program, Center for Design and Manufacturing Innovation, and NET Ben Franklin Advanced Technology Center, Technical Report #87-001, 1987, pg 35.

[53] "CEFE Arbeitsgruppe 23: CAD/NC - Referentzteil," CEFE CAD/CAM Entwicklungsgesellschaft, 1991.

[54] Butterfield, W., Green, M., Scott, D., Stoker, W., "Part Features for Process Planning," Computer Aided Manufacturing International, C-86-PPP-01, 1988, pp. 9-48-61.

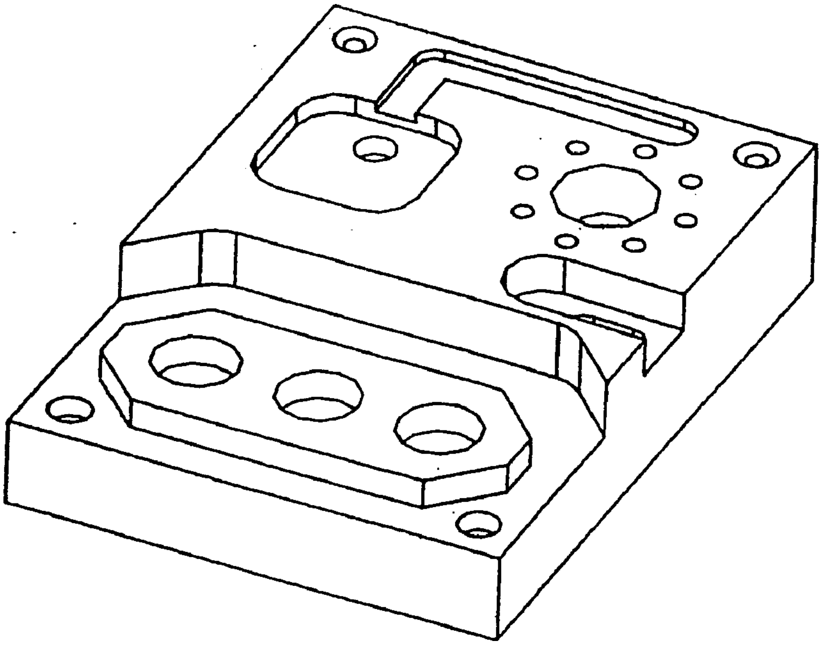

Figure 81. Aber and Glemser reference NC geometry.

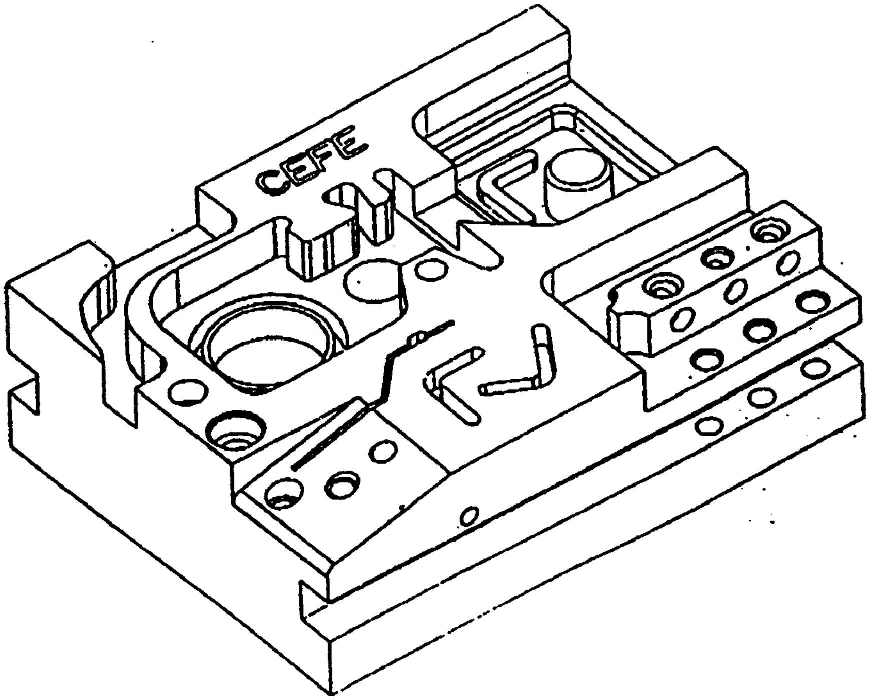

Figure 82. CEFE - CAD/Numerical control reference geometry.

The primary differences between the CEFE geometry and that of the Aber and Glemser part is a ramp milling exercise and the length of the resultant NC program. The requirement for a positioning axis or additional setup was present in the Aber and Glemser configuration for affixing a machined marking [55] to the demonstration part.

Having considered these efforts as cited from the literature, the Aber and Glemser part is selected with focus on a portion of the demonstration geometry. The specific example used here will maintain the structure as referenced while modifying the content. This content modification is introduced in an effort to keep focus on the contribution of the framework developed in this research without opening for debate the proposed manufacturing method and machining processes.

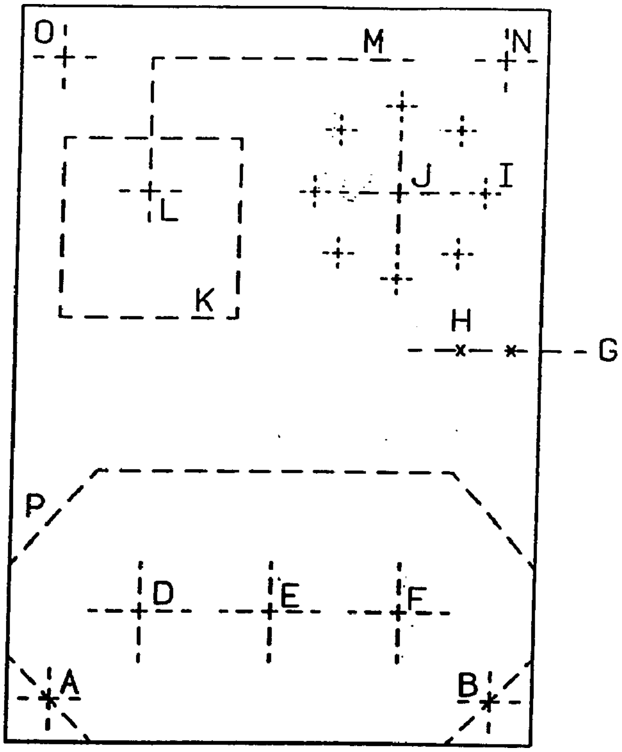

Figure 83. Aber and Glemser reference part feature's layout.

---

[55] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, pg 50.

The portion of the geometry chosen has machining requirements that specify both hole making and pocket machining. Area IJ of the referenced part [56] has been identified for this purpose. Figure 83 on page 207 presents area IJ in the context of the demonstration part. This area provides an opportunity to highlight the contribution of the developed framework while maintaining a focus on the research core.

7.1 The Geometry

The "I" area is composed of a set of two feature hole types as shown in Figure 84.

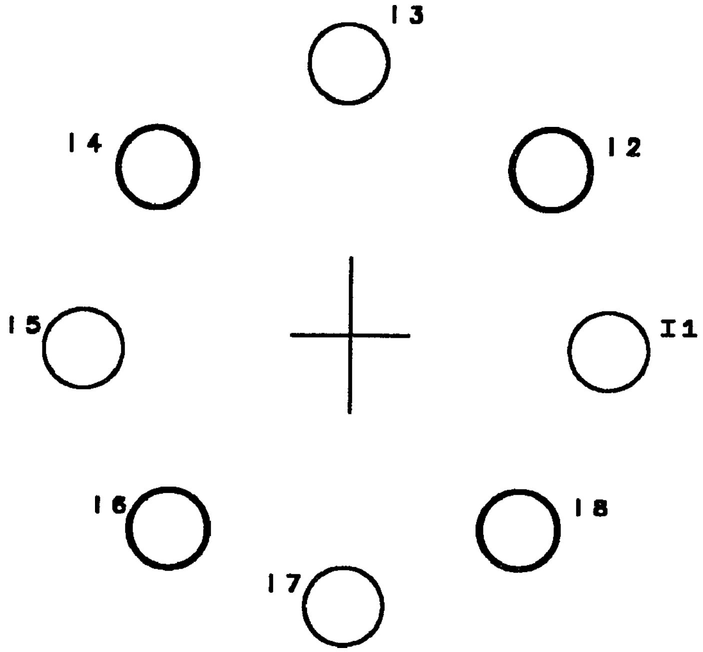

Figure 84. I Area - detailed and labeled.

---

[56] Aber, D.M., and Glemser, R.G., "The Design, Manufacturing Planning, and Fabrication Requirements of a Part for the Demonstration of a Developing Numerical Control Application," Joint Report Series: Manufacturing Systems Engineering Program, Center for Design and Manufacturing Innovation, and NET Ben Franklin Advanced Technology Center, Technical Report #87-001, 1987, pg 8.

Each feature will have attributes as an instance of a hole type class [57] as noted in Table 2. This abridged class is proposed to provide focus to the research core.

HOLE Feature - Material Removal Volume:

- DIAMETER of the Hole
- DEPTH of the Hole

Tolerance:

- LOCATION of the Hole
- FORM of the Hole

Technological Attributes

- BLIND bottom condition

Table 2. Dimensional summary of I hole features.

Name	Class	Diameter	Depth	Location Tolerance	Form Tolerance	Bottom Condition
I1	Hole	6.35 mm	3.0 mm	Normal	Loose	Flat
I2	Hole	6.35 mm	8.5 mm	Tight to I8	Tight	Flat
I3	Hole	6.35 mm	3.0 mm	Normal	Loose	Flat
I4	Hole	6.35 mm	8.5 mm	Normal	Tight	Flat
I5	Hole	6.35 mm	3.0 mm	Normal	Loose	Flat
I6	Hole	6.35 mm	8.5 mm	Normal	Tight	Flat
I7	Hole	6.35 mm	3.0 mm	Normal	Loose	Flat
I8	Hole	6.35 mm	8.5 mm	Reference I2	Tight	Flat

---

[57] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, pp 29,46.

The "J" area has been modeled as a circular pocket. An internal boss or island is associated with this pocket as shown in the Figure 85. This geometry is modeled by attributes as an instance of a pocket and boss feature class [58].

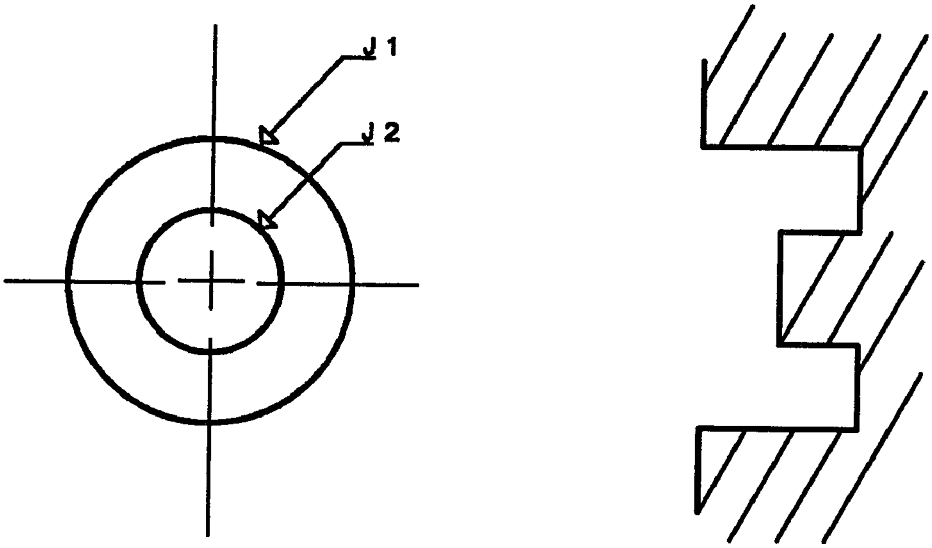

Figure 85. J Area - detailed and labeled.

POCKET Feature - Material Removal Volume

- DIAMETER of the Pocket
- DEPTH of the Pocket

Tolerance:

- LOCATION of the Pocket
- FORM of the Pocket

Technological Attributes

- FLAT bottom condition

---

[58] "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," ISO/WD 10303-224 Industrial Automation Systems and Integration - Product Data Representation and Exchange, 1995, pp 21,22,43,63.

The boss feature attributes are:

BOSS Feature - Material Removal Volume

- DIAMETER of the Boss
- DEPTH of the Boss

Tolerance:

- LOCATION of the Boss
- FORM of the Boss

Technological Attributes

- FLAT top condition

Table 3 provides the dimensional summary of the machining requirements for the "J" area.

Table 3. Dimensional summary for J pocket and boss features.

Name	Class	Diameter	Depth	Location Tolerance	Form Tolerance	Bottom Condition
J1	Pock	20 mm	8.5 mm	Normal	Normal	Flat
J2	Boss	1.75 mm	8.5 mm	Normal	Tight	Flat
J1/J2	NEW pock	20 mm	8.5 mm	Normal	Tight	Flat

The geometry for this example is constructed from the information provided above and modeled as features in the CAD model. This model is displayed as Figure 86 on page 212. This model has ten features that will be used to define machining operations.

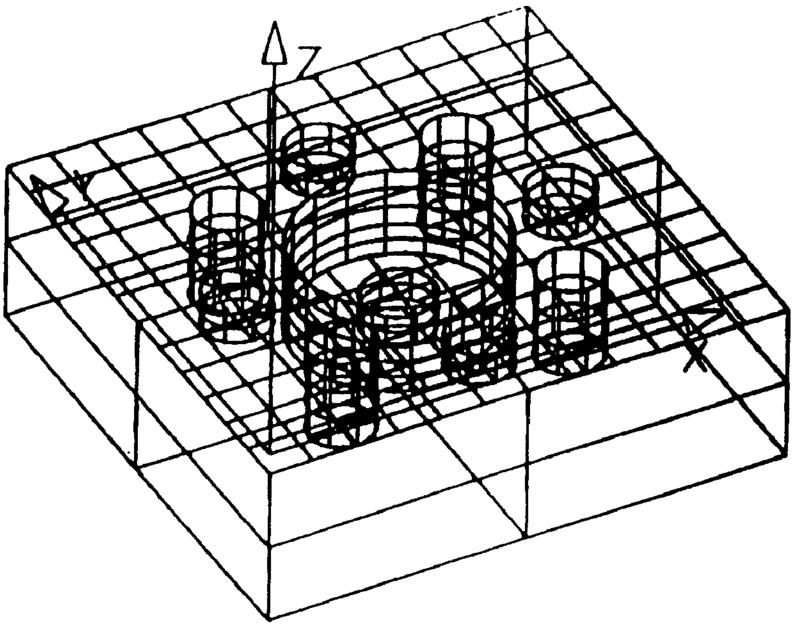

Figure 86. Example part geometry with features.

This model will be read to extract these feature class and instance descriptions through an application programming interface as described in Chapter 4 - System Components Definition, CAD/CAM System Services. This model provides the features data necessary to begin the generative NC process. In the example, this data has been generated to simulate feature-based design. However, feature identification techniques are developing [59,74,75] for use with existing geometry that could also provide the required input for this framework.

7.1.1 The Stock

The stock selected for this example is a material billet prepared from available stock. The stock billet is prepared and sized to be 50 mm in width and 50 mm in length with a thickness of 17 mm.

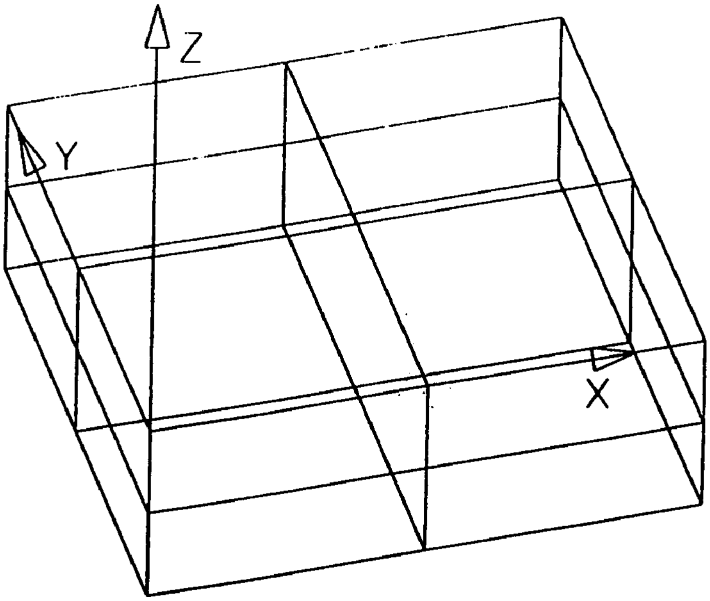

Figure 87. Stock model for the Example part.

The machining requirement for the example geometry has coincident faces with the material billet along the X-axis and Y-axis. The part program would generate a material removal requirement to create the planar part surface along the Z-axis.

7.2 Treatment by the Framework

The treatment by the framework of this example geometry is recorded as the part proceeds through the process of generative numerical control as developed in this research. The *functions* applied to the example geometry are identified to communicate to the reader the processing flow. The narrative is supported by figures to record the evolution of the workpiece from a stock billet to final part.

Attention is given to the manufacturing processes as developed for the framework by this research and summarized below.
This is the sequential processing core of the framework at the level where generative numerical control part programming functions are invoked.

1. Machining Environment Formulation
2. Material Removal Identification
3. Machining Requirements Analysis
4. Machining Process Generation
5. Machining Sequence Generation
6. Auxiliary Command Generation
7. Manufacturing Results and Analysis

The components that have been defined for the framework will be referred to during the example part walkthrough. These components are given context in Figure 88 on page 215 placed here to refresh the reader in the generative numerical control architecture.

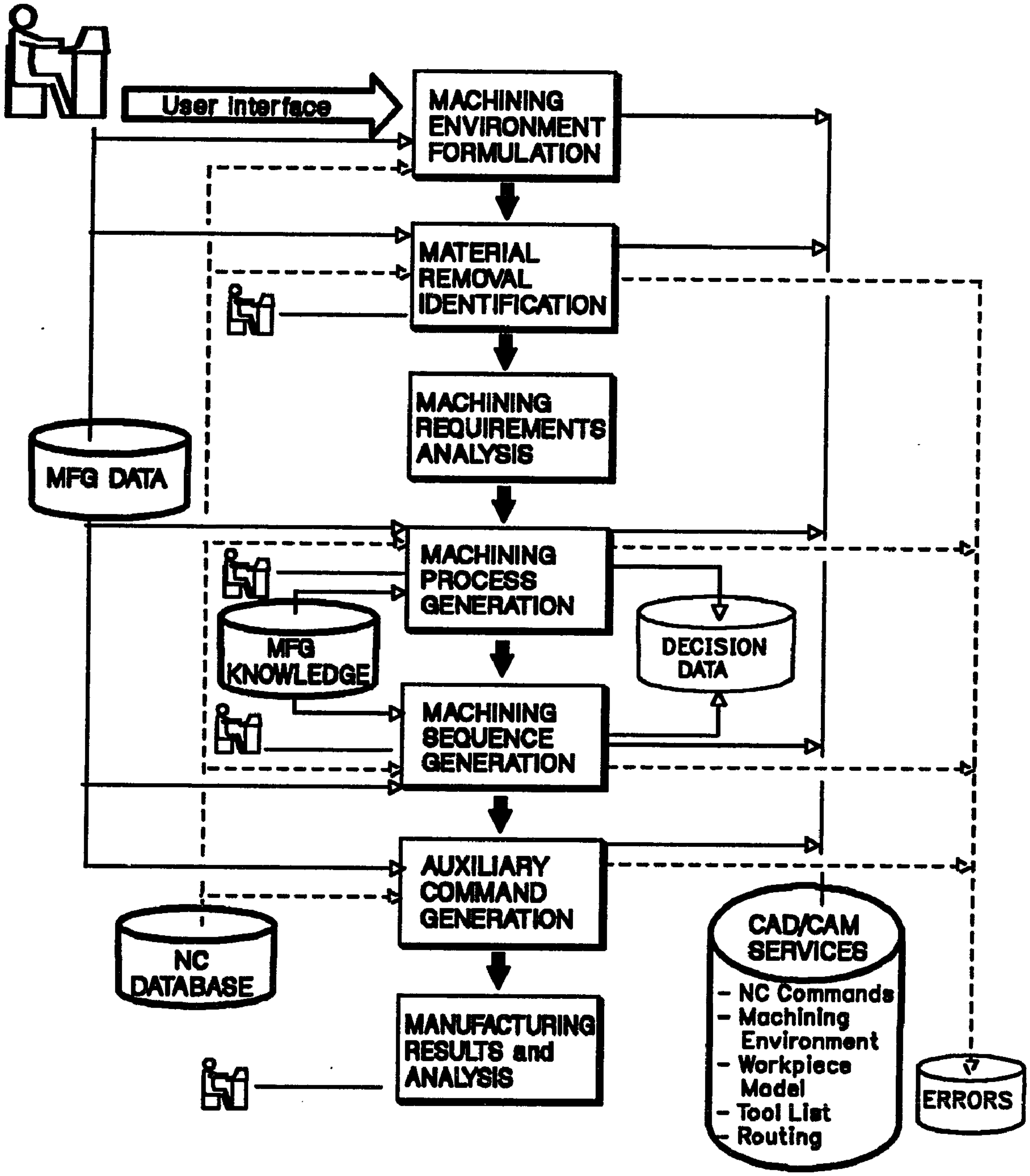

Figure 88. The generative numerical control architecture.

7.2.1 Machining Environment Formulation

This process starts the generative numerical control flow. All required user inputs are collected here for the example part. The physical modeling of the numerical control workspace is completed here.

Application Data Collection: The process begins for the example part with this activity. The part model is selected from the CAD/CAM system file structure. The stock model is likewise selected or built from the part model. A fixture model is not used in this example. Material is defined from manufacturing database by user selection. The machine (DM2400) is keyed or selected in the same manner. The setup dimensions (0,0,78) are keyed at this time. The user has entered all data necessary to create a part program.

The reader will note that no "similar to part 1234" data has been entered by the user or is used by the application. The user can access optional environment data that has been collected at application build-time. This data controls the application execution and provides the options that will be used by the imbedded algorithms that supply *part programming functions.*

NC Workspace Creation: The machining environment data collected from the user, supplied as build-time manufacturing process data or retrieved from the manufacturing database, is used to create the NC workspace. Parameters belonging to the machine (axis travel limits and dead zones) are used to build the machining envelope. The part and stock for this example are placed in the machining envelope using the setup dimensions.

7.2.2 Material Removal Identification

The example part is now found in the NC workspace. The part begins a flow through the sequence of processes to build a machining plan and part program output. The first process applied to the example part will identify all material removal requirements.

Feature Transfer: The example part feature definition must be mapped to a machining semantic for use in this material removal type application. Each design feature is analyzed and mapped to a machining feature form. This activity treats each attribute for the machining feature class. This function is provided by attribute mapping tables and methods. The result is a transfer of geometric, tolerance, and technological data from design features to machining features.

Part-In-Stock Features: The example part is placed in the context of the stock billet. In this context, material is found outside of the part definition and inside of the stock boundary. This function identifies the need to quantify these "additional" material removal requirements. This material is quantified as a material removal requirement with a singular uniform thickness slab placed on the top of the upper part surface. This material is generated as a slab type planar face machining feature 50 mm by 50 mm with a thickness of 2 mm.

Machining Accessibility: The features of the example part are analyzed to determine if they can be machined in the machining environment formulated for this part program. This analysis includes the *part-in-stock feature* (slab type feature) previously identified. The functions that are invoked in the programmed code that implements the framework are based on spindle accessibility (location and orientation) of each material removal requirement. The NC database is consulted for allowable spindle orientation, axis travel limits, and dead zones in the *machining envelope*.

A machining approach vector is established for each feature. The machining approach vector for Hole I1 is displayed in Figure 89.

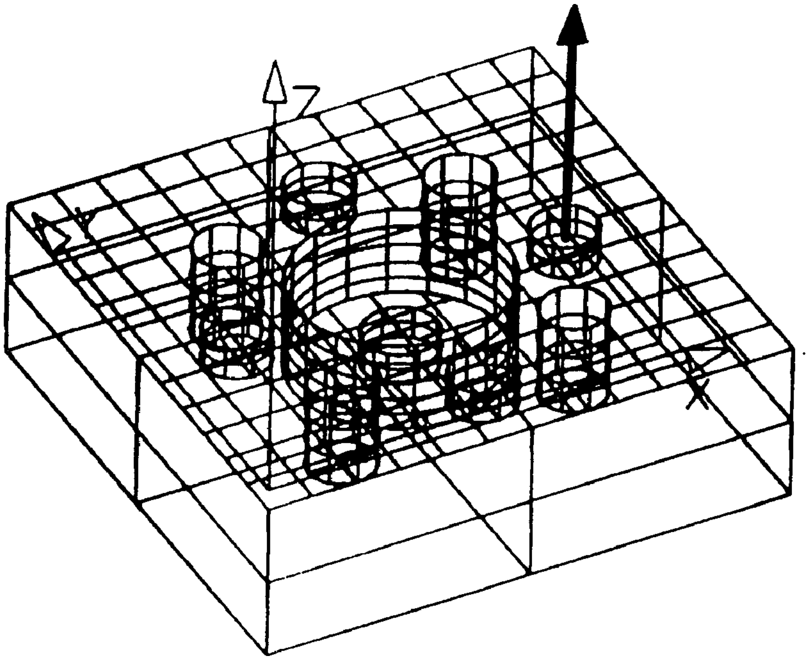

Figure 89. Machining approach vector for Hole I1.

Machining Feature Topology: The example part is further analyzed to determine the existing context of the feature in the part and any possible relationships with other features that should be taken into account during the planning activities on this feature. In the example part, the framework specifies a system labeling of boundary elements to determine if they are part surfaces or they must be machined through. Other functions within this activity would be used on more complex models to identify efficient material removal for that part.

Planner's Window: The example part will not use this activity. Generally, the functions available here are reserved for user interactive extrapolation of programmed capabilities to handle the tasks of Material Removal Identification.

7.2.3 Machining Requirements Analysis

The example part enters the activities of *Machining Feature Classification* and *Feature Geometry Analysis* with a set of qualified machining features.

- Pocket J1 and Boss J2
- Hole I1
- Hole I2
- Hole I3
- Hole I4
- Hole I5
- Hole I6
- Hole I7
- Hole I8
- Slab S1

Machining Feature Classification: Each machining feature is classified to provide an order of presentation to the algorithms that comprise the functions specified for *Machining Process Generation* The classification is accomplished by a systematic sorting based on feature type, relative size, machining approach vector, and apparent patterns.

The sorting strategy for the DM2400 with no tool changer orders machining features from smaller to larger relative size. Same size features are ordered by largest depth. Therefore, the holes are ordered first, then pocket, then slab. The example part has the machining approach vectors for each machining feature in the same direction. A pattern (depth = 8.5 mm) is found in holes I2, I4, I6, and I8.

A second pattern (depth = 3.0 mm) is found in holes I1, I3, I5, and I7. The contribution of this activity is not to be confused with *Machining Sequence Generation* that will be determined for the machining operations planned for the example part.

1. Hole I2 - Pattern 1
2. Hole I4 - Pattern 1
3. Hole I6 - Pattern 1
4. Hole I8 - Pattern 1
5. Hole I1 - Pattern 2
6. Hole I3 - Pattern 2
7. Hole I5 - Pattern 2
8. Hole I7 - Pattern 2
9. Pocket J1 and Boss J2
10. Slab S1

Feature Geometry Analysis: The example part's machining features are each analyzed to determine a set of geometric attributes that are used in the generation of a machining process. These additional parameters are stored with the machining feature objects. Algorithms are invoked to calculate:

- logical machining direction,
- machining requirement minimum radius,
- minimum tool channel width(s),
- maximum tool channel width(s),
- machining requirement area, and
- machining requirement profile length.

7.2.4 Machining Process Generation

This process is iterated for each machining feature presented in the order determined by the *Machining Feature Classification* activity.

Machining Operations: The pattern function has reduced the scope of the problem presented to this activity. The machining processes for ten individual machining features are generated by four iterations of the functions.

1. Holes (I2,I4,I6,I8) - Pattern 1 - Drilling/Deephole
2. Holes (I1,I3,I5,I7) - Pattern 2 - Drilling
3. Pocket and Boss - Pocket Milling
4. Slab - Face Milling

Tool Selection: The *Tool Selection* activity for Hole I2 is begun by the packaging of a tool requirement. This requirement can constrain each attribute of the tool definition. For the purpose of this example, the attributes for diameter, flute length, tooth material, and number of flutes are constrained. This tool requirement for I2 is presented in Table 4.

Table 4. Tool Requirement for Hole - I2.

Attribute	Preferred	Minimum	Maximum
Tool Type	End_mill	...	...
Diameter	6.35 mm	6 mm	6.35 mm
Flute Length	26 mm	20 mm	38 mm
Tool Material	HSS	HSS	HSS
Number of Flutes	2	2	2

The flute length has been calculated from the machining feature depth with appropriate allowance for tool tip geometry, and breakthrough as required. The tool requirement is packaged for a query to the manufacturing database. The query will return one tool that best fits the requirement. The tool returned is shown in Table 5.

Table 5. Tool Found for Hole - I2 - EH2I002500Y012500L.

Attribute	Value
Diameter	6.35 mm
Flute Length	31.75 mm
Tool Material	HSS
Number of Flutes	2
Corner Radius	0
Composition	one_piece

Machining Operation Area: The exact area that will be engaged by the tool must be validated so that the machining process defined will not cause collision with the part surface. In the example part, each machining operation requires this validation. The data is retrieved from the CAD/CAM services NC processor in three components. These components are the material removal tool path area, tool approach area, and tool retract area. The *Machining Operation Area* is constructed by using each of these area and the selected tool information. This object is then used to check collision with the final part. A collision would invoke *functions* to iterate to a collision free process or mark a feature as not machinable.

7.2.5 Machining Sequence Generation

The example part enters the *Machining Sequence Generation* process with a machining process linked to each machining feature. Each machining process is comprised of an ordered set of machining operations. Each machining feature has a tool defined. There is no sequence data explicitly specified by the user within the total set of machining operations for the example part.

Geometry Access: The example part enters this *activity* with the slab being found as requiring machining before every other feature. This geometric access is recorded as *precedence*. No additional precedence links are generated for this activity. This data is recorded and made available to the *Operation Sequence* activity.

Operation Sequence: The example part now enters the *Operation Sequence* activity. Here, *precedence* information is used with imbedded sequence knowledge to generate a machining operation sequence specific to the example part. To begin the example sequence, *precedence* is consulted. The Slab - S1 machining operation is selected by the system first. There is no strong link to S1 so *precedence* cannot offer the next sequence candidate. Therefore, *sequencing knowledge* is consulted. The manufacturing know-how guides possible candidate selection to milling operations before drilling. This results in Pocket J being selected and added to the sequence. No more milling operations remain so the program moves to drilling type operations. The program selects I1 as the start point. I2 is selected next. I2 has a *strong precedence link* formulated from the tight tolerance information between I2 and I8. Therefore, I8 must follow I2 without any other operations between.

A minimum path distance rule is invoked giving I7 as the next candidate. I6, I5, I4, and I3 follow the same selection criteria. The results of sequencing are found below:

1. Slab S1
2. Hole I1
3. Hole I2
4. Hole I8
5. Hole I7
6. Hole I6
7. Hole I5
8. Hole I4
9. Hole I3
10. Pocket J1 and Boss J2

This sequence of machining operations can be executed using the solid stock model to verify completeness of the part program. This function is provided by the *workpiece modeler.* The function is approached at the material removal volume level. Each volume is constructed from programmed calls to the NC processor machining area calculations. The *workpiece modeler* results for the complete *Machining Sequence Generation* is shown in Figure 90 on page 225 Individual machining operation tool path can be visualized through the user interface placed for this activity by calling NC processor tool path calculation services. The centerline of the tool path is then displayed to the user.

The result of *Machining Sequence Generation* has produced a detailed process plan that has determined the machining operations and sequence that will transform the stock billet into a machined solid model. A complete numerical control part program will also need to determine *transition tool paths* motions between each machining operation and *auxiliary commands.*

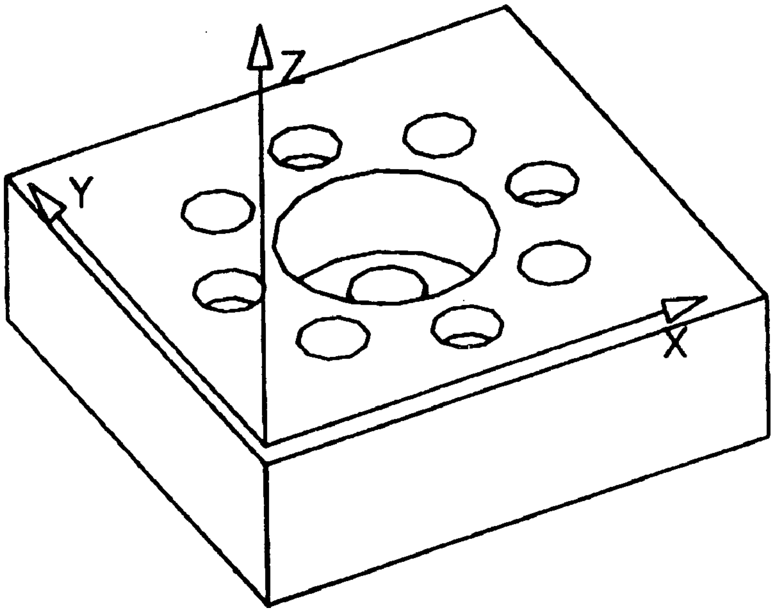

Figure 90. Machining Sequence Generation - Example part verification.

Tool Assembly Selection: For the example part the flute length requirement is not affected by "stacked" machining operations. It therefore remains as 31.75 mm with the selected tool remaining EH2I002500Y012500L. The tool holder that will connect with the shank of this tool is found by making a query to the manufacturing database using information about the spindle connectivity requirement as found with the DM2400 machine . This assembly is mathematically checked for collisions throughout the sequence. If collisions had been detected, another tool assemble requirement would be generated. This new query would have updated collision constraints data as extracted from the previous collision failures.

7.2.6 Auxiliary Command Generation

This process begins with information previously determined for the tool path for each site of material removal about the stock, the sequence of machining operations, and each tool assembly. The *Tool Transition Elements* and *Auxiliary Commands* required to complete a part program are generated by this process for the example part.

Tool Transition Elements: The first move from the machine "home" position to the first point of machining is added in the exact syntax required for the DM2400 machine. This machine requires a FROM statement. Likewise, the tool paths between each machining operation are added. A single clearance plane for the part has been defined in the start up data. Each *transition tool path* will therefore retract to z-clear then xy translate to next location with z-approach. Also, to/from tool change and to/from rotary axis changes would be added at this time. The last *transition tool path* generated for the example part will be a move to PROGRAM END. The DM2400 requires a move to "home" position.

Auxiliary Commands: The example part machined on the DM2400 does not require many auxiliary commands to control efficient machining.

Program generation for the example part is now complete. All required NC commands for the CAD/CAM system NC processor have been created. This data flows to the *Manufacturing Results and Analysis* activities as the final process supported by the framework for generative numerical control.

7.2.7 Manufacturing Results and Analysis

All tool paths and machine control commands have been generated in the context of the CAD/CAM system NC processor. This final process allows the user access to the results and provides insight into how the results were generated.

Manufacturing Results: The *Manufacturing Results* of the generative NC system as defined by this research are found with four characteristics. Results are available for *manufacturing output data*, *decision data*, *error data*, and *machining environment data.*

The primary *manufacturing output data* result is the *NC Commands* For example part purposes, these results are translated by the CAD/CAM services NC processor and presented in both APT source file and NC data form. The APT Source was created by CAD/CAM NC Processor utility that translates the internal form NC Commands to a standard output type such as Automatically Programmed Tool (APT). This APT source output for the example part is found in Appendix A.

The APT Source file is used with a post processor created to translate this standard into NC data expected by the machine tool controller. This example part numerical control data output for the DM2400 is presented in Figure 91 on page 228. It has been created using the post processor for the DM2400 machine. A review will show a direct correlation with the APT Source file. This is primarily due to the scope of the example and the machine capabilities. Greater differences in the level of these outputs would be identified if more controller cycle function was called for in the original NC Commands.

```
000 START MM 01
001 SETUP
002 SPINDLE   ON
003 FR XYZ = 76.0
004 GO X  000.000
005    Y  000.000
006    Z  000.438
007 GO Z  006.000
008 GO X -004.175
009    Y  002.540
010 GO Z -002.000
011 FR XYZ = 30.0
012 GO X  054.175
013 GO Y  008.155
014 GO X -004.175
015 GO Y  013.770
016 GO X  054.175
017 GO Y  019.385
018 GO X -004.175
019 GO Y  025.000
020 GO X  054.175
021 GO Y  030.615
022 GO X -004.175
023 GO Y  036.230
024 GO X  054.175
025 GO Y  041.845
026 GO X -004.175
027 GO Y  047.460
028 GO X  054.175
029 FR XYZ = 76.0
030 GO Z  001.000
031 GO X  040.588
032    Y  025.000
033 FR XYZ = 30.0
034 GO Z -005.000
035 FR XYZ = 20.0
036 GO Z  001.000
037 FR XYZ = 76.0
038 GO X  036.022
039    Y  036.022
040 FR XYZ = 30.0
041 GO Z -007.000
042 FR XYZ = 20.0
043 GO Z  001.000
044 FR XYZ = 76.0
045 GO Z -007.000
046 FR XYZ = 30.0
047 GO Z -010.500
048 FR XYZ = 20.0
049 GO Z  001.000
050 FR XYZ = 76.0
051 GO Y  013.978
052 FR XYZ = 30.0
053 GO Z -007.000
054 FR XYZ = 20.0
055 GO Z  001.000
056 FR XYZ = 76.0
057 GO Z -007.000
058 FR XYZ = 30.0
059 GO Z -010.500
060 FR XYZ = 20.0
061 GO Z  001.000
062 FR XYZ = 76.0
063 GO X  025.000
064    Y  009.413
065 FR XYZ = 30.0
066 GO Z -005.000
067 FR XYZ = 20.0
068 GO Z  001.000
069 FR XYZ = 76.0
070 GO X  013.978
071    Y  013.978
072 FR XYZ = 30.0
073 GO Z -007.000
074 FR XYZ = 20.0
075 GO Z  001.000
076 FR XYZ = 76.0
077 GO Z -007.000
078 FR XYZ = 30.0
079 GO Z -010.500
080 FR XYZ = 20.0
081 GO Z  001.000
082 FR XYZ = 76.0
083 GO X  009.413
084    Y  025.000
085 FR XYZ = 30.0
086 GO Z -005.000
087 FR XYZ = 20.0
088 GO Z  001.000
089 FR XYZ = 76.0
090 GO X  013.978
091    Y  036.022
092 FR XYZ = 30.0
093 GO Z -007.000
094 FR XYZ = 20.0
095 GO Z  001.000
096 FR XYZ = 76.0
097 GO Z -007.000
098 FR XYZ = 30.0
099 GO Z -010.500
100 FR XYZ = 20.0
101 GO Z  001.000
102 FR XYZ = 76.0
103 GO X  025.000
104    Y  040.588
105 FR XYZ = 30.0
106 GO Z -005.000
107 FR XYZ = 20.0
108 GO Z  001.000
109 FR XYZ = 76.0
110 GO Y  024.525
111 GO Z -004.750
112 FR XYZ = 20.0
113 ZERO AT
114    X  025.000
115    Y  025.000
116 GR A360.000
117 >REF COODS
118 GO Y  018.175
119 ZERO AT
120    X  025.000
121    Y  025.000
122 GR A360.000
123 >REF COODS
124 FR XYZ = 30.0
125 GO Z  005.000
126 FR XYZ = 76.0
127 GO Y  024.525
128 FR XYZ =  5.0
129 GO Z -007.500
130 ZERO AT
131    X  025.000
132    Y  025.000
133 GR A360.000
134 >REF COODS
135 GO Y  018.175
136 ZERO AT
137    X  025.000
138    Y  025.000
139 GR A360.000
140 >REF COODS
141 FR XYZ = 76.0
142 GO Z -001.500
143 GO Y  031.675
144 GO Z -010.500
145 FR XYZ =  5.0
146 ZERO AT
147    X  025.000
148    Y  025.000
149 GR A-360.000
150 >REF COODS
151 GO Y  031.825
152 ZERO AT
153    X  025.000
154    Y  025.000
155 GR A180.000
156 >REF COODS
157 ZERO AT
158    X  025.000
159    Y  025.000
160 GR A180.000
161 >REF COODS
162 FR XYZ = 76.0
163 GO Z  004.000
164 GO X  000.000
165    Y  000.000
166 GO Z  000.438
167 SPINDLE  OFF
168 END NEWPART
```

Figure 91. Manufacturing Results - Numerical control data for DM2400.

8.0 Recommendations for Further Study

This section is included to identify and propose further development of this research. The framework for generative NC part programming identified in this research would serve as the foundation for leading-edge solutions to many other computer integrated manufacturing problems. Further study and understanding of the contribution made herein will identify yet additional opportunities.

The reasoning that is used to approach this section is quite straight-forward. However, this approach may elude the strict practice and now intuitive "top down approach" in which we have been schooled. The top down approach [59] "...improves the orderliness and quality of system development." However, "bottom up" reasoning continues the development of the generative numerical control framework as approached with further study. This framework forms the nucleus of a manufacturing system, whose breadth can be expanded through further study.

The top down approach has been employed within this research project. The framework simplicity and inherent quality is the result of this approach. Within the framework, the lowest level of manufacturing data has been structured into a contribution to the manufacturing operations planning and NC part program generation process. This lower level of manufacturing data should form a constraint on further study topics. The effectiveness of the total manufacturing solution, combining the framework and further study topics, will be determined by the simplicity of data exchange and required processing. Therefore, detailed manufacturing data and knowledge defined in the framework must be applied in the solution of these further study efforts.

---

[59] Sempreviro, P., Systems Analysis: Definition Process and Design, Chicago: Science Research Associates, Inc., 1982, pg 81.

8.1 Further Study Approach

The framework of generative NC part programming proposes components, assembles the components in an architecture, and identifies the functions on this architecture for an industrial production level manufacturing tool. The key element in further study will be to use this detailed data and transform it to serving higher level manufacturing decision based and related applications. The detail data is where the bulk of manufacturing knowledge exists for the methods and practices of detailed parts manufacturing. It is critical that the investment taken to validate (use of standard data) or implement shop specific data be protected. Thus, a bottom up approach to further study will ensure that these methods are exercised and no "doors closed" on detailed data or know-how. This data and know-how residing at the lowest level or bottom within an implementation of the framework.

Further study on the findings of this research is proposed in the areas of manufacturing design advisor machine tool selection, tool/fixture design, and additional manufacturing process support (eg. sheet metal, EDM, lathe, ...). Figure 92 positions the further study topics and this current research. Each of these areas of study will require unique skills to pursue. Also, the findings in each area will provide a differing level of contribution to an overall computer integrated manufacturing solution. The approach suggested for further study is to investigate current research on each topic within the framework presented herein. This result could be guided with industrial production requirements for greatest utility in today's business environment. The manufacturing design advisor is introduced by Case [6]. He contends that a process modeling capability for the designer will result in design manufacturability. This advice could be based on the same manufacturing data and know-how used to generate the NC part program. Joseph and Davies [34] comment on the subject of manufacturing design advisor.

Their comment[60] proposes that the solution to the operational planning and NC part programming problem can be applied to guide designers to a manufacturable result within the design stage. This is the role of a manufacturing design advisor.

> Through computer aids in the form of Computer Aided Design (CAD) systems and numerically controlled (NC) machines are being used in industry to reduce the lead time, there is still unnecessary interaction between the different functions. To integrate properly, the designer should be able to access the manufacturing knowledge so that manufacturing difficulties can be considered at the design stage.

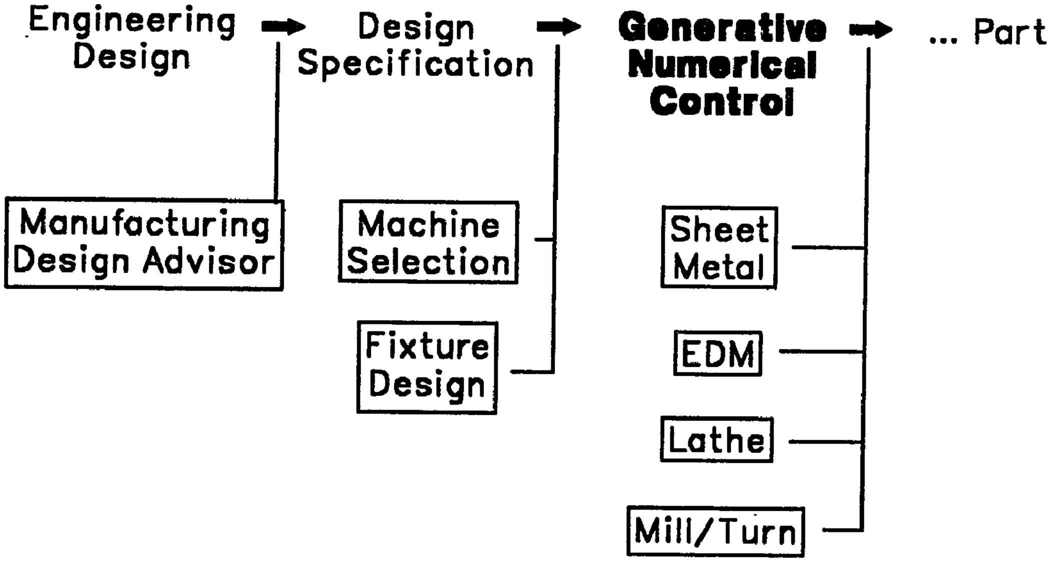

Figure 92. Further study topics and current research.

---

[60] Joseph, A.T., Davies, B.J., "EXCAP - An Expert Process Planning System for Turned Components," First International Conference on Expert Planning Systems, IEE Conference, 1990, p 130.

The tool and fixture design problem is introduced by Prombanpong [61]. Current approaches to the problem are identified in this treatment. The approach chosen integrates the part's process plan with fixture design considerations. The model provided by this approach could be modified to use the data and know-how of the framework for generative NC. The formulation of this module may provide additional insight on the optimal placement of this object in the functions supporting the generative architecture. Currently, the fixture is defined as part of *machining environment formulation* in the beginning of the overall process. Consideration can be given to identification of this object later in the process after *machining process generation* This placement could allow the current *functions* to provide additional information to the fixture design problem.

The application of the framework for generative NC to additional manufacturing processes (eg. sheet metal, EDM, lathe) is a topic for further study. In the literature, Tisza [71] presents an approach to the sheet metal forming. This treatment employs a knowledge based system structuring of a variant planning processing approach. Ghosh [19] examines the process planning problem related to strategy formulation in two-dimensional cutting cell. He proposes a global planning approach to the available capabilities of the cell to be a solution for the two-dimensional cutting of sheet metal. Jian [33] provides an approach to electric discharge machining [EDM]. This work is an interactive formulation for processing planning based on three planning modules. The result of the system is to store the plan with a part number key for later retrieval. Matsumura [53] approaches the problem of operational planning for turning with an adaptive control type solution. However, he keeps the planner on-line during the first phase of machining to sample practical machining of the workpiece. He then proposes change in the machining parameters and update the theoretic database.

The machine selection problem is presented in context of problem definition, proposed approach with the current research contributions, and a summary of probable findings. The probable findings are presented to direct the further study as goals for related research.

8.2 Machine Selection Problem Definition

The machine selection problem [31] specifies the machine tool or machines required to manufacture a given part specification. The importance of this NC part programming related application will increase as overall shop knowledge based on experience declines. This need also increases as more complex NC equipment replaces single purpose manufacturing machines. Decentralization of production facilities and an increase of in-house part manufacturing (formerly supplier based part manufacturing) will further emphasize the urgency of providing solutions that fully use the available shop capacity efficiently. The considerations in finding a solution to the machine selection problem are presented below.

- Business requirement
    - Volume of parts required, lead time for manufacturing
- Machining Requirement
    - Part geometry
    - Material type
    - Stock form
    - Fixture type
    - Tolerance specification
        - Size, form, position, orientation
- Process Requirements
    - Cutter selection
- Machines Available
    - Shop capacity
    - Current loading
    - Alternative analysis

This problem definition for machine selection is presented graphically in Figure 93. The relationship to the result of the current research is highlighted in the this figure. These result is provided by an implementation of the generative NC framework. This machining operations planning solution is the generalized "machining requirements" highlighted core in this figure. The activity of extracting the machining requirement from a part design would utilize framework contributions. However, this utilization would occur at a higher level than that of machining operations planning. This further study addresses the availability of this higher-level information from the generative numerical control framework.

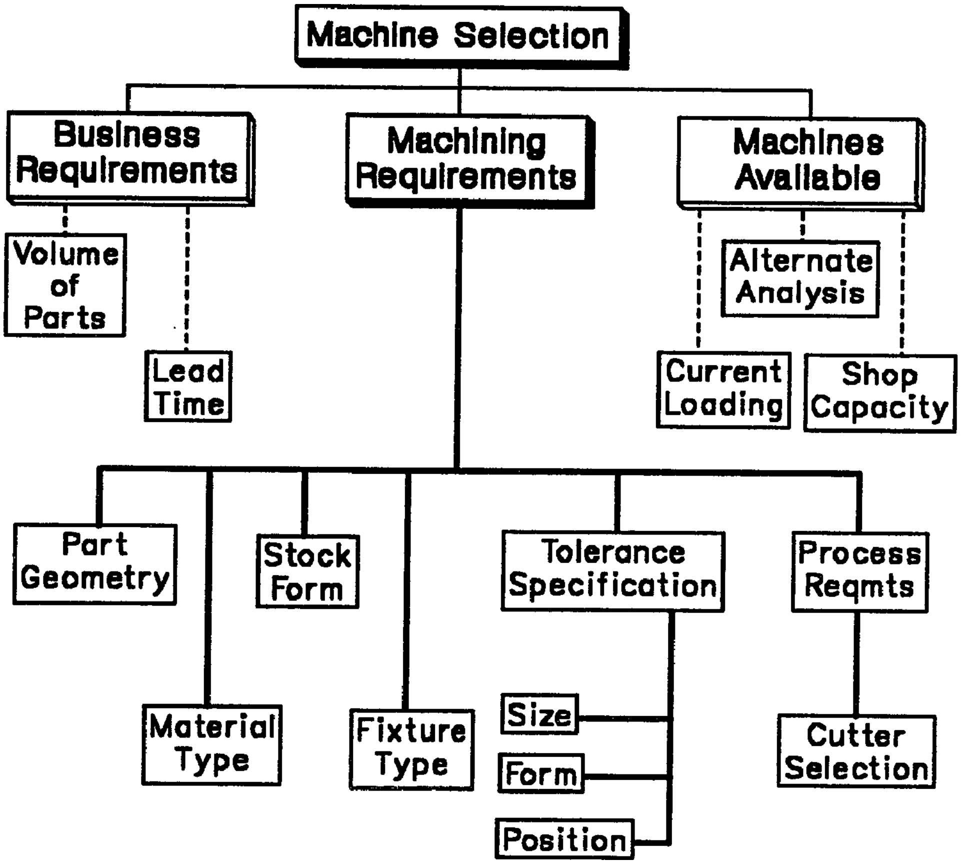

Figure 93. Machine selection problem definition.

8.2.1 Machine Selection Proposed Approach

The problem of machine selection can be approached in the framework for NC part program generation. Further study to be undertaken will concentrate on how to provide on aggregation of the detailed data and know-how required by machining operations planning. This higher-level information will be derived from an analysis made within the context of machining requirements on lower level material removal volumes.

The machines available for selection are also a consideration in the further study area. The representation of the current configuration of the shop or manufacturing capacity within the shop will be a contribution to this solution. The machine representation will identify classes of objects with similar machining attributes. These objects are similar in both their physical and logical configuration. The classes can then be used in an exhaustive comparison of machining requirements with each unique machine instance. The machine class representation will need to support comparison priority assignments to handle the variation of shop practice, asset loading, and scheduled equipment maintenance.

The business requirements of manufacturing that are considerations in the machine selection problem include topics like volume of parts required and lead time available for manufacturing. This information will constrain the pool of candidate machine tools through the analysis of availability and schedule.

The problem of machine selection can now be reduced to a set of syntax analysis problems. The machine representations for the machining selection problem will be tested to determine to what extent the business requirements and machining requirements can be achieved. This approach will produce a series of set/subset identifications that must be analyzed in the context of machines available for efficient production.

8.2.2 Machine Selection Proposed Findings

The proposed findings of this further study topic would be the definition of a processor to be executed prior to approaching the machining operation planning problem. This processor would be a machine selection module. The processor would expand the domain of the solution provided by the framework for NC part program generation. The machine selection module would complement and support the detailed manufacturing data and manufacturing know-how representation of shop methods and practices required for the low-level numerical control programming task. These proposed findings would identify a different perspective or semantic to the data and know-how that supports the machining operational planning problem. This semantic provides an inference from this machining operations planning problem to higher level process planning problem. This semantic is applied to the low-level manufacturing data and manufacturing knowledge. The result is that the low-level data is used in aggregate form to solve the higher level problem. Finding and applying this semantic is ideally an implementation of computer integrated manufacturing. This research result could integrate available resources from the system defined by the framework for generative NC part programming to approach the solution of machine selection at the processing plan level.

9.0 Conclusions

The results of this research are providing two perspectives as relating to the problem of numerical control part programming for industrial production machining. In the first perspective, the framework for generative numerical control is developed. This framework represents the research objective by providing an original and significant contribution to the knowledge of this field. In the second perspective, a set of new concepts is developed. These concepts guide the reduction-to-practice of the manufacturing technology imbedded in the framework as a solution for NC parts programming. This reduction-to-practice is the element of the research effort that promotes the introduction of this solution as a next generation industrial production numerical control part programming technique.

9.1 The Framework

The framework for generative numerical control has three elements. These elements are:

1. the definition of generative numerical control system components,
2. an architecture that assembles these components, and
3. the specification of part programming functions to be delivered by this architecture.

This framework is configured from components that provide inputs and receive outputs from a sequential processing core. This configuration provides an architected solution that manages both NC part program quality and programming methods productivity. The generative NC part programming functions are then overlayed onto this architecture to provide a comprehensive definition for generative numerical control manufacturing technology.

9.1.1 Generative Numerical Control System Components

The framework is constructed from fundamental building blocks. These building blocks are developed in this research as the definition of system components. This research proposes that a generative NC system is comprised of six primary components.

1. User Interface
2. Manufacturing Process Data
3. Manufacturing Knowledge
4. Manufacturing Database
5. Part Programming Functions
6. CAD/CAM System Services

These components are selected from the available technologies. The characteristics unique to generative NC for each of the system components are developed in this research. These components are assembled in the architecture that provides a context for realizing their individual contributions to this technology. As determined by this research, placement and connectivity of these components are as important as their unique form and content. Placement and connectivity within the framework is developed as the controlling architecture of generative numerical control part programming technology.

9.1.2 The Architecture

The architecture for generative NC is developed to be implemented in an automatic and generative system. This architecture can be implemented as a turn key solution. This type of parts programming solution is configurable with a minimum of specific machining shop effort. This style of implementation handles machining know-how as imbedded application data. This part programming solution is robust with respect to coverage the *generative part programming functions*. However, the delivered solution is generated using *basic machining technology* as the source of machining know-how. The site has chosen not to invest in tailoring the system components to be specific to their machining shop. This style of implementation is enabled through the architecture controlling concept of *knowledge implementation with turnkey alternatives.*

The framework research results also enable extensive tailoring to a machine shop's specific methods and practices. This configuration integrates user knowledge with the system *basic manufacturing technology*. This alternative style of implementation employs off-line knowledge management principles to represent user manufacturing knowledge that provides the source of machining shop specific know-how. This user machining know how is partitioned from the system's *basic machining technology* through the architecture controlling concept of *manufacturing know-how hierarchy.*

The architecture also provides the fundamental structure, that when implemented in a solid modeler, provides a mathematically established collision free tool path and completed part program. This contribution to part programming for industrial production machining is enabled through the incremental growth in quality formed by a stepwise progression through the defined generative NC part programming functions.

The quality of the part program is commensurate with the accuracy and completeness of the manufacturing data and knowledge used to specify the particular solution domain.

This architecture is developed to improve the productivity of manufacturing engineers and part programmers. This concept is provided through the integration of a company's standard methods and manufacturing practices with system generated tool path and part programming technology. The result is an increase in both part program quality and the associated programming methods productivity. Each part program is generated based on a shop's machining capabilities. The machines, tooling, processes, and associated machinability data available and specific to the machine shop guides machining process generation. The adaptability and scalability of this solution to a particular machine shop's requirements are provided by system customizing functions applied at each point where machining technology is introduced into the system. The architecture also provides for utilization of evolving technology by offering incremental growth, modification, or replacement of *functions* that facilitate process or processing decisions throughout the system. This system evolution support is necessary to comply with the requirements set forth for technology that enables agile manufacturing [61].

A user interface is designed and placed in this architecture for obtaining and analyzing the results of this automatic NC part program generation. This automatic mode of operation can be augmented with user expertise and additional knowledge integrated throughout optional user interfaces. These optional user dialogues are placed with specific connectivity to the core processing modules as formalized by the architecture.

---

[61] Miller, W., Preiss, K., Thompson, G., "A Model of Customer Supplier Relationships for Agile Manufacturing," Agility Forum, 1994.

9.1.3 Numerical Control Part Programming Functions

The set of part programming functions developed in this research has established the breadth and depth of the architecture. This is the level of research contribution with which the NC part programmer, NC engineer, and NC toolmaker can most readily identify. These potential users are dependent on the functions of a system to complete today's part programming assignment. This contribution to current research expands the definition of the generative part programming solution as documented in the literature. This research has lead to the specification of a minimum set of functions that define generative NC part programming. These functions specify the required and detailed capability of a generative NC technology solution. This contribution has developed the functional "blueprint" for this technology.

This set of functions will also serve as the definition to structure an analysis and evaluation of emerging implementations of the technology. In other words, this contribution serves as a "benchmark" for generative numerical control part programming solutions. This application of the research results is important in an emerging technology where instances of current research and commercialization of that research have compromised the basic definitions. The terms most frequently applied, without sufficient evidence that system functions enable the universally accepted definitions, are "automatic" and "generative." The application of the set of functions defined in this research to the evaluation of automatic and generative efforts will directly identify the level of both automation and generative planning incorporated in that solution.

In industry, this set of functions configured through the architecture can propose a NC part programming paradigm shift. This shift directs focus to "process" from tool path definition. This process paradigm also achieves synergistic results between the functions defined for generative NC. Acceptance of this process paradigm denotes a longer term commitment to manufacturing planning and NC part programming technology. This commitment is to a set of functions overlayed on an architecture that collectively provides an open system. This type of commitment differs considerably from that of traditional procurement decisions for NC packages that are guided only by available "application features and functions" paradigm.

9.1.4 Framework Contribution Summary

This research contributes to the current knowledge of generative NC part programming and its system definition. The contributions of this research from the perspective of the development of a framework for generative NC are tabulated in Table 6.

Table 6. Research result contributions.

Research Result	Contributions - Framework for Generative NC
System Components Definition	Implement company manufacturing standard methods and practices
	Reduces in solution system maintenance
	Promotes standardized machining approaches
Architecture	Automates operational planning process
	Automates part programming process
	Provides Generative programming approach
	Balances available skill and competency
	Promotes resource scheduling efficiency
	Promotes synergy among programming functions
Functions Definition	Defines generative NC technology
	Bounds problem of generative numerical control
	Improves process overall quality
	Identifies benchmark content

9.2 Reduction-to-Practice of a Technology

The concepts that enable the reduction-to-practice of the generative system framework developed in this research are presented here. This contribution provides further insight into the requirements that guide a successful implementation of this technology as proposed by this research.

9.2.1 Automatic Processing

The framework is designed to be invoked as a fully automatic process. Therefore, the architecture is developed to provide an automatic parts programming solution. Object declarations and process flow are controlled by the architecture to ensure data quality and completeness within each processing module. The processing structure has been formulated to place these module boundaries for the containment of logical *process, activity,* and *function* groupings. This process flow then maximizes its′ application for implementation as an automatic process by minimizing the complexity of inter-module dialogue variation. The exchange between modules is highly structured to provide for automatic processing. This structure also enables efficient assignment and consumption of computing resources.

9.2.2 Standard Components

This framework is established within the environment provided by an industrial production solid modeler that is imbedded in a CAD/CAM system. This environment manages the relative size of the technology proposed for generation of the detailed machining plan and numerical control part program. Physical size of the system solution is minimized through the engineered utilization of CAD/CAM system services as developed in this research. The CAD/CAM services definition highlights the unique requirements necessary for a production implementation of the framework for generative numerical control.

9.2.3 Objects and Data Structures

The complexity of the data handling tasks associated with next generation parts programming technology is managed within the CAD/CAM environment. The components developed in the research each contribute with a defined set of objects and object manipulation capabilities. These definitions eliminate replication of basic CAD/CAM and NC processor services. However, the availability of these object definitions is constrained to the CAD/CAM system as designed and implemented for next generation manufacturing application development. Where this concept can be obtained, system implementation efforts are focused on the parts programming process. The alternative is to build and maintain a seemly elaborate scheme of objects and the associated data structures for generative NC technology. The merits of this alternative was not identified in this research.

9.2.4 Open Manufacturing Technology

The generative NC technology has a well constructed dependency on machining technology in the form of manufacturing data, knowledge, and processing. This solution specifies each location of manufacturing technology as open for installation definable to the basic framework structure. The concept provides for a dynamic system to be configured through a combination of *basic machining technology* and company machining standards and practices. This concept also creates user installation responsibility for the accuracy and completeness of the manufacturing data, knowledge, and processing being applied to generate their parts programs. This concept reformulates the traditional customer and supplier "after-the-sale" maintenance. This relationship typically concentrates customer effort on problem identification. The supplier concentrates on problem cause determination. Open manufacturing technology provides that a portion of the solution is built by the customer. Likewise, the customer accepts a portion of problem cause determination.

9.2.5 Configurable Processing

A comprehensive generative NC solution as developed in this research includes breadth and depth of part programming functions processing. Each general machining problem will have varied processing requirements within this system. These requirements will need to be managed for the efficient consumption of available resources. The framework sponsors processing flow configuration enabled through an imbedded algorithm hierarchy. This concept provides for run-time user adaptation of the processing of the generative NC system to meet the anticipates requirements of a specific part type. Trends in run-time processing requirements can be recorded by the system as default processing recommendations.

9.2.6 Technology Reduction-to-Practice Contribution Summary

This research also contributes to the current knowledge of implementing a generative NC part programming system. The contributions of this research from the perspective of the reduction-to-practice concepts of this development of a framework for generative NC are tabulated in Table 7.

Table 7. Research result contributions.

Research Result	Contribution - Technology Reduction-to-Practice
Automatic Processing	Reduced delivery time
	Reduced routine work
	Simulation
Standard Components	Automatic information flow
	No information loss
Object and Data Structures	Improved reproducibility
	Improved precision
Open Manufacturing Technology	Modification different areas
	Iteration different areas
Configurable Processing	Intensive calculations
	Reduced working time
	Iterative processing

9.3 Validation of Research

The framework for generative numerical control has been validated by a series of activities which have followed the author's research. These activities were focused on the implementation of the system definition within the environment provided by a solids modeling based CAD/CAM system. This industrial implementation of the framework originated with the selection of the CATIA system as the provider of CAD/CAM and NC processor services. An implementation team was assembled for these activities and staffed with both engineering and programming expertise. The key contributors of this implementation team are acknowledged at the beginning of this dissertation.

The proof-of-concept implementation of this research was to prototype the system definition with available technologies. CATIA Version 3.2.5 was used for this purpose. This level of code required considerable application level developments in feature based modeling and manufacturing data structures to complete the implementation. The rules processor selected was Knowledge Tool ®. This activity resulted in an automatic and generative next generation part programming technique. This activity was closely followed by an implementation with the new capabilities of CATIA Version 4. This development was able to take advantage of an imbedded features processor, rules processor, and numerical control technological based processor. This form of the implementation provided the means to validate my research results in the manufacturing industry worldwide. The framework for generative NC was then adopted as a strategic direction for the CATIA system. This research's framework provides the basis of solutions requested by major industrial partners. The result being worldwide announce of this technology on September 1995 as the basis for a family of manufacturing products. The publication of the research coincides with this announcement.

® Knowledge Tool is a registered trademark of the IBM Corporation.

10.0 References

1. Arikan, M.A. Sahir, Totuk, O.H., "Design by Using Machining Operations," Annals of the CIRP, Vol 41/1, pp. 185-188, 1992.

2. Bala, M., Chang, T.C., "Automatic Cutter Selection and Optimal Cutter Path Generation for Prismatic Parts," International Journal Production Research, Vol 29 No 11, pp. 2163-2176, November 1991.

3. Bobrow, J.E., "NC Machining Tool Path Generation From CSG Part Representations," Computer Aided Design, Vol 17 No 2, pp. 69-76, 1985.

4. Brown, C.W., Gyorog, D.A., "Generative Inspection Process Planner for Integrated Production," Advances in Integrated Product Design and Manufacturing, American Society of Mechanical Engineers, pp. 151-162, 1990.

5. Butterfield, W., Green, M., Scott, D., Stoker, W., "Part Features for Process Planning," Computer Aided Manufacturing International, C-86-PPP-01, 1988.

6. Case, K., "Using a Design by Features CAD System for Process Capability Modelling," Computer Integrated Manufacturing Systems (UK), Vol 7 No 1, pp. 39-49, 1994.

7. Chan, S.C., Voelcker, H.B., "An Introduction to MPL - A New Machining Process/Programming Language," IEEE International Conference on Robotics and Automation, IEEE, pp. 333-344, 1986.

8. Chang, T.C., "TIPPS - A Totally Integrated Process Planning System," Ph.D. Thesis, Virginia Polytechnic Institute and State University, UMI, 1982.

9. Choi, B.K., Lee, C.S., Hwang, J.S., Jun, C.S., "Compound Surface Modelling and Machining," Computer Aided Design, Vol 20 No 3, pp. 127-136, 1988.

10. Christman, A., "Generative Numerical Control Product Assessment," CIMdata Report 1994.

11. Colding, B.N., "Intelligent Selection of Machining Parameters for Metal Cutting Operations: The Least Expensive Way to Increase Productivity," Robotics and Computer-Integrated Manufacturing, Vol 9 No 4-5, pp. 407-412, August-October 1992.

12. Dassault Systemes, "CATIA Manufacturing Infrastructure - General Information Manual," SH52-1105-15, August 1995.

13. Dassault Systemes, "CATIA 3D Parametric Variational Modeler Interactive Functions Reference Manual," SH52-0625-01, 1993.

14. daSilva, R.E., Wood, K.L. and Beaman, J.J., "Representing and Manipulating Interacting and Interfeature Relationships in Engineering Design for Manufacture," Advances in Design Automation, Vol 23 Part 1, ASME, pp. 1-8, 1990.

15. Delbressine, F.L.M. deGroot, R. van der Wolf, A.C.H., "On the Automatic Generation of Set-Ups Given a Feature-Based Design Representation," Annals of the CIRP, Vol 42/1, pp. 527-530, 1993.

16. El-Midany, T.T., EL-Bay, M.A., "Auto CNC Part Programming through Machinability Data Base System," Computer Applications and Design Abstraction, American Society of Mechanical Engineers, Vol 43, ASME, pp. 1-7, 1992.

17. Ferstenberg, R., Wang, K.K., and Muckstadt, J., "Automatic Generation of Optimized 3-Axis NC Programs Using Boundary Files" IEEE International Conference on Robotics and Automation, IEEE, pp. 325-332, 1986.

18. Gehrke, Dr., Lauscher, J. Dr., "Personal Communication on generative numerical control fundamental approach," Volkswagen AG and IBM, Wolfsburg, Germany, 1994.

19. Ghosh, S., Beitialarrangotia, J., Douglas, S., "Automatic Process Planning Strategy Applied to a Flexible Two-Dimensional Cutting Facility," Journal of Materials Processing Technology, Vol 37 No 1-4, pp. 61-68, 1993.

20. Groover, M.P., Automation, Production Systems, and Computer-Integrated Manufacturing, Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

21. Groover, M.P., Fundamentals of Modern Manufacturing: Materials, Processes, and Systems, Upper Saddle River, New Jersey: Prentice-Hall, 1996.

22. Groover, M.P., Zimmers, E.W., CAD/CAM: Computer-Aided Design and Manufacturing, Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

23. Gu, P., Zhang, Y., "OOPPS: An Object- Oriented Process Planning System," Computers and Industrial Engineering, Vol 6 No 4, pp. 709-731, Pergamon, 1994.

24. Guyder, M.K., "Automatically Optimized NC Tool Path Generation for Machining," United States Patent Number 4907164, 1990.

25. Han, C., Li, J., Ham, I., "Development of an In-House Computer Automated Process Planning System Based on Group Technology Concept," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

26. Hinds, J.K., "Automatic Machining Using Constructive Solid Geometry with Boolean Combinations of Primitives Including Tool Offset to Form a Machining Pattern," United States Patent Number 4618924, 1986.

27. Houten, F.J.A.M. van, PART: A computer Aided Process Planning System, FEBODRUK Enschede, Netherlands, 1991.

28. IAMS, MetCAPP V4 Process Planning Application - Technology Modules Manual, Cincinnati, Ohio: Institute of Advanced Manufacturing Sciences, Inc., 1994.

29. IBM, Automatically Programmed Tool for AIX, GB35-0157-00.

30. IBM, NC Post Processor Generator Customization Guide, SH20-6910-1.

31. Jackson, Steven D., Mittal, Ravie O., "Automatic Generation of 2-Axis Laser-Cutting NC Machine Program and Path Planning from CAD," Computer in Industry, Vol 1 No 2, pp. 223-231, 1993.

32. Jagdale, S.S., Wang, K.K., "An AI Based Generative Process Planning System for Machining Operations," Proceedings: Second International Conference on Industrial and Engineering Application of Artificial Intelligence and Expert Systems, IEA/AIE, pp. 528-535, 1989.

33. Jain, V., Batra, J., Garg, A., "Computer Aided Process Planning (CAPP) for Electric Discharge Machining (EDM)," Journal of Materials Processing Technology, Vol 48 No 1-4, pp. 561-569, 1995.

34. Joseph, A.T., Davies, B.J., "EXCAP - An Expert Process Planning System for Turned Components," First International Conference on Expert Planning Systems, IEE Conference, pp. 130-135, 1990.

35. Joshi, S., Chang, T.C., "Graph Based Heuristics for Recognition of Machining Features from a Solid Model," Computer Aided Design, Vol 20 No 2, pp. 58-66, 1988.

36. Joshi, S., Chang, T.C., Liu, C.R., "Process Planning Formalization In an AI Framework," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

37. Kang, T., Nnajl, B.O., "Feature Representation and Classification for Automatic Process Planning Systems" Journal of Manufacturing Systems, Vol 12 No 2, pp. 133-145, 1993.

38. Khoshnevis, B., Tan, W., "Automated Process Planning for Hole-Making," Manufacturing Review, Vol 8 No 2, ASME, pp. 106-113, 1995.

39. Kochan, D., Editor, CAM Developments in Computer - Integrated Manufacturing, Germany: Springer-Verlag 1986.

40. Koloc, J., "The Influence of the Programming Language on the Productivity and Reliability of Part Programming," Proceedings of the 13th Annual Meeting of Numerical Control Society, 1976.

41. Koncewicz, D., "Numerical Control Programming System NCS," Pr Nauk Institute Cybernetics Tech Politech Wroclaw, Vol 89 No 39, pp. 13-19, 1991.

42. Lawler, B.D., "Artificial Intelligence/Expert Systems - Applications to CAM," NCGA '90 Conference Proceedings, pp. 92-101, 1990.

43. Lehtihet, E.A., Aber, D.M., "The Application of Solid Modeling to Manufacturing Engineering Problems," North American Manufacturing Research Conference XV, 1987.

44. Lee, Y., Chang, T., "CASCAM - An Automated System for Sculptured Surface Cavity Machining," Computers in Industry, pp. 321-342, 1991.

45. Lennon, M.B., Roth, R.N., "A Review of CAD/CAM Solutions for Numerical Control Part Programming," International Mechanical Engineering Congress, Sydney, 1991.

46. Liu, D., "Utilization of Artificial Intelligence in Manufacturing," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

47. Loney, G.C., Ozsoy, T.M., "NC Machining of Free Form Surfaces," Computer Aided Design, Vol 19 No 2, pp. 85-90, 1987.

48. Lu, S. C-Y, "Knowledge Processing Technology for Concurrent Engineering Tasks," Advances in Computer-Integrated Manufacturing - Volume 2, JAI Press Ltd., pp. 195-231, 1993.

49. Mantyla, M., An Introduction to Solid Modeling Rockville, Maryland: Computer Science Press, 1988.

50. Marefat, M., Britanik, J., "A Case-Based Approach for Process Planning," Manufacturing Science and Engineering, Vol 68 No 1, ASME, pp. 3-12, 1994.

51. Marks, P., "Expert System-Based Generative Planning Process," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

52. Matsuda, M., Kimura, F., "Extraction of Machining Features for Milling Data Generation," Computer Applications in Production and Engineering, pp. 153-360, 1991.

53. Matsumura, T., Obikawa, T., Shirakashi, T., Usui, E., "Autonomous Turning Operational Planning with Adaptive Prediction of Tool Wear and Surface Roughness," Journal of Manufacturing Systems, Vol 12 No 3, pp. 253-262, 1993.

54. Mortenson, M.E., Geometric Modeling, New York: John Wiley and Sons, 1985.

55. Milner, D., Vasiliou, V., Computer-Aided Engineering for Manufacture, London: Kogan Page Ltd, 1987.

56. Niebel, B., "Mechanized Process Selection for Planning New Designs," ASME Paper 737, 1965.

57. Oliver, J.H., Wysocki, D.A., Goodman, E.D., "Gouge Detection Algorithm for Sculptured Surface NC Generation," Journal of Engineering for Industry Vol 115 No 1, ASME, pp. 139-144, 1993.

58. Pande, Dr. S.S., Prabhu, B.S., "An Expert System for Automatic Extraction of Machining Features and Tooling Selection for Automats," Computer-Aided Engineering, Vol 7 No 4, pp. 99-103, August 1990.

59. Phillips, R.H., Mouleeswaran, C.B., "A Knowledge-Based Approach to Generative Process Planning," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

60. Pressman, R.S., Williams, J.E., Numerical Control and Computer Aided Manufacturing, New York: John Wiley, 1977.

61. Prombanpong, S., Lewis, R., "Computer-Aided Fixture Design System with Process Planning Integration for Prismatic Parts Manufactured on CNC Machining Centers," ASME Proceedings - International Computers in Engineering, pp. 369-380, 1992.

62. Reester, K.A., "Post Processor Words for GNC Prototype 1," Generative Numerical Control Project, Document Number 91-015-011, September 1991.

63. Rembold, U., Dillmann, R., Editors, Computer Aided Design and Manufacturing Methods and Tools, New York: Springer-Verlag, 1986

64. Rentz, R.E., "RAMP Technology and Intelligent Processing in Small Manufacturing," Federal Conference on Intelligent Processing Equipment, NASA Conference Publication 3138, 1991

65. SDRC, I-DEAS Master Series Product Catalog, 1993.

66. Siores, E., Fang, X.D.,Yao, Y, "Intelligent Design for Manufacturing Employing Knowledge Based Expert Systems," Information Technology for Advanced Manufacturing Systems, IFIP TC5/WG5.3, 1991.

67. Slovensky, L., "Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Form Features," Industrial Automation Systems and Integration - Product Data Representation and Exchange, ISO/WD 10303-224, 1995.

68. Socha, D.H., "Process Planning The Tangible Medium for Group Technology Benefits," CAPP: From Design to Production, Society of Manufacturing Engineers, 1988.

69. South Carolina Research Authority, "Software and Design Document for the Rapid Acquisition of Manufactured Parts (RAMP) Small Mechanical Parts (SMP) Generative Process Planning Environment (GPPE) Version 1.5," Report No. DDR003006-0, April 1992.

70. Ssemakula, M., Sivac, P., "Automatic Generation of NC Part Programs," AUTOFACT '87, pp. 35-48, 1987.

71. Tisza, M., "Expert Systems for Metal Forming," Journal of Materials Processing Technology, Vol 53 No 1-2, pp. 423-432, 1995.

72. Vandenbrande, J.H., Requicha, A.A.G., "Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 15 No 12, pp. 1269-1285, December 1993.

73. Wang, W.P., "Integration of Solid Modeling for Computer Planning," ASME Conference on Computer-Aided/Intelligent Process Planning, pp. 177-187, 1985.

74. Weeks, N.J., Dickenson, S., "Toward an Integrated Parametric, Feature-Based, Sheet Metal Manufacturing Solution," AUTOFACT Conference Proceedings - SME, pp. 47-64, 1992.

75. Wei, Y., Fisher, G., Santos, J.L., "A Concurrent Engineering Design Environment for Generative Process Planning using Knowledge-Based Decisions," ASME Design Technical Conference, 1990.

76. Yeh, C.H., Fischer, G.W., "A Structured Approach to the Automatic Planning of Machining Operations for Rotational Parts Based on Computer Integration of Standard Design and Process Data," The International Journal of Advanced Manufacturing Technology, Vol 6 No 3, pp. 285-298, 1991.

77. Yip-Hoi, D., Ditta, D., "Issues in Computer-Aided Process Planning for Parallel Machine Tools," Advances in Design Automation, Vol 1 Part 1, ASME, pp. 153-161, 1993.

78. Zhang, Y., Nee, A., Fuh, J., "A Hybrid Approach to Computer-Aided Process Planning for Prismatic Parts," Computers in Engineering, Vol 1, ASME, pp. 437-442, 1994.

79. Zhao, Z., Baines, R.W., Blount, G.N., "Definition of Generic Relationships Between Design and Manufacturing Information for Generative Process Planning," Computer Integrated Manufacturing Systems, Vol 6 No 3, August 1993.

Appendix A. Example Part - APT Source File

```
PPRINT EXAMPLE PART
PPRINT
PPRINT MODEL = EXAMP.NCPRISM
PPRINT PART OP = *PO2
PPRINT MACHINE = DYNA2400
$$*CATIA0
$$*AXS35
$$    1.00000      .00000      .00000      .00000
$$     .00000     1.00000      .00000      .00000
$$     .00000      .00000     1.00000      .00000
TLAXIS/ .000000, .000000, 1.000000
PPRINT OPERATION NUMBER: 1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,49.99800
FROM /    .00000,    .00000,    .43800                PT    1
PPRINT OPERATION NUMBER: 2
PPRINT TOOL = FMD_TA1
CUTTER/ 6.350000,  .000000, 3.175000,  .000000,  .000000,$
        .000000, 31.750000
TOOLNO/1,MILL,6.35000
LOADTL/1
PPRINT OPERATION NUMBER: 3
RAPID
GOTO  /    .00000,    .00000,   6.00000               PT    2
RAPID
GOTO  /  -4.17500,   2.54000,   6.00000               PT    3
RAPID
GOTO  /  -4.17500,   2.54000,  -2.00000               PT    4
PPRINT OPERATION NUMBER: 4
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO  /  54.17500,   2.54000,  -2.00000               PT    5
GOTO  /  54.17500,   8.15500,  -2.00000               PT    6
GOTO  /  -4.17500,   8.15500,  -2.00000               PT    7
GOTO  /  -4.17500,  13.77000,  -2.00000               PT    8
GOTO  /  54.17500,  13.77000,  -2.00000               PT    9
GOTO  /  54.17500,  19.38500,  -2.00000               PT   10
GOTO  /  -4.17500,  19.38500,  -2.00000               PT   11
GOTO  /  -4.17500,  25.00000,  -2.00000               PT   12
GOTO  /  54.17500,  25.00000,  -2.00000               PT   13
GOTO  /  54.17500,  30.61500,  -2.00000               PT   14
GOTO  /  -4.17500,  30.61500,  -2.00000               PT   15
GOTO  /  -4.17500,  36.23000,  -2.00000               PT   16
GOTO  /  54.17500,  36.23000,  -2.00000               PT   17
GOTO  /  54.17500,  41.84500,  -2.00000               PT   18
GOTO  /  -4.17500,  41.84500,  -2.00000               PT   19
GOTO  /  -4.17500,  47.46000,  -2.00000               PT   20
GOTO  /  54.17500,  47.46000,  -2.00000               PT   21
PPRINT OPERATION NUMBER: 5
```

```
RAPID
GOTO /  54.17500,  47.46000,   1.00000                PT   22
RAPID
GOTO /  40.58750,  25.00000,   1.00000                PT   23
PPRINT OPERATION NUMBER: 6
PPRINT MACHINE OPERATION = DRILL WITH DWELL/DEL
PPRINT OPERATION NAME    = *MO19
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO /  40.58750,  25.00000,  -5.00000                PT   24
DELAY/2.00000
FEDRAT/MMPM,199.80000
GOTO /  40.58750,  25.00000,   1.00000                PT   25
PPRINT OPERATION NUMBER: 7
RAPID
GOTO /  36.02203,  36.02203,   1.00000                PT   26
PPRINT OPERATION NUMBER: 8
PPRINT MACHINE OPERATION = DRILL DEEPHOLE
PPRINT OPERATION NAME    = *MO16
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO /  36.02203,  36.02203,  -7.00000                PT   27
FEDRAT/MMPM,199.99800
GOTO /  36.02203,  36.02203,   1.00000                PT   28
RAPID
GOTO /  36.02203,  36.02203,  -7.00000                PT   29
FEDRAT/MMPM,300.00000
GOTO /  36.02203,  36.02203, -10.50000                PT   30
FEDRAT/MMPM,199.99800
GOTO /  36.02203,  36.02203,   1.00000                PT   31
PPRINT OPERATION NUMBER: 9
RAPID
GOTO /  36.02203,  13.97797,   1.00000                PT   32
PPRINT OPERATION NUMBER: 10
PPRINT MACHINE OPERATION = DRILL DEEPHOLE
PPRINT OPERATION NAME    = *MO15
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO /  36.02203,  13.97797,  -7.00000                PT   33
FEDRAT/MMPM,199.99800
GOTO /  36.02203,  13.97797,   1.00000                PT   34
RAPID
GOTO /  36.02203,  13.97797,  -7.00000                PT   35
FEDRAT/MMPM,300.00000
GOTO /  36.02203,  13.97797, -10.50000                PT   36
FEDRAT/MMPM,199.99800
GOTO /  36.02203,  13.97797,   1.00000                PT   37
PPRINT OPERATION NUMBER: 11
RAPID
```

```
GOTO  /  25.00000,   9.41250,   1.00000                    PT   38
PPRINT OPERATION NUMBER: 12
PPRINT MACHINE OPERATION = DRILL WITH DWELL/DEL
PPRINT OPERATION NAME    = *MO18
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO  /  25.00000,   9.41250,  -5.00000                    PT   39
DELAY/2.00000
FEDRAT/MMPM,199.80000
GOTO  /  25.00000,   9.41250,   1.00000                    PT   40
PPRINT OPERATION NUMBER: 13
RAPID
GOTO  /  13.97797,  13.97797,   1.00000                    PT   41
PPRINT OPERATION NUMBER: 14
PPRINT MACHINE OPERATION = DRILL DEEPHOLE
PPRINT OPERATION NAME    = *MO14
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO  /  13.97797,  13.97797,  -7.00000                    PT   42
FEDRAT/MMPM,199.99800
GOTO  /  13.97797,  13.97797,   1.00000                    PT   43
RAPID
GOTO  /  13.97797,  13.97797,  -7.00000                    PT   44
FEDRAT/MMPM,300.00000
GOTO  /  13.97797,  13.97797, -10.50000                    PT   45
FEDRAT/MMPM,199.99800
GOTO  /  13.97797,  13.97797,   1.00000                    PT   46
PPRINT OPERATION NUMBER: 15
RAPID
GOTO  /   9.41250,  25.00000,   1.00000                    PT   47
PPRINT OPERATION NUMBER: 16
PPRINT MACHINE OPERATION = DRILL WITH DWELL/DEL
PPRINT OPERATION NAME    = *MO17
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO  /   9.41250,  25.00000,  -5.00000                    PT   48
DELAY/2.00000
FEDRAT/MMPM,199.80000
GOTO  /   9.41250,  25.00000,   1.00000                    PT   49
PPRINT OPERATION NUMBER: 17
RAPID
GOTO  /  13.97797,  36.02203,   1.00000                    PT   50
PPRINT OPERATION NUMBER: 18
PPRINT MACHINE OPERATION = DRILL DEEPHOLE
PPRINT OPERATION NAME    = *MO13
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
```

```
GOTO  /  13.97797,  36.02203,  -7.00000                       PT    51
FEDRAT/MMPM,199.99800
GOTO  /  13.97797,  36.02203,   1.00000                       PT    52
RAPID
GOTO  /  13.97797,  36.02203,  -7.00000                       PT    53
FEDRAT/MMPM,300.00000
GOTO  /  13.97797,  36.02203,  -10.50000                      PT    54
FEDRAT/MMPM,199.99800
GOTO  /  13.97797,  36.02203,   1.00000                       PT    55
PPRINT OPERATION NUMBER: 19
RAPID
GOTO  /  25.00000,  40.58750,   1.00000                       PT    56
PPRINT OPERATION NUMBER: 20
PPRINT MACHINE OPERATION = DRILL WITH DWELL/DEL
PPRINT OPERATION NAME    = *MO20
PPRINT PATTERN NAME      =
PPRINT TOOL ASSEMBLY     = FMD_TA1
SPINDL/RPM,500.00000,CLW
FEDRAT/MMPM,300.00000
GOTO  /  25.00000,  40.58750,  -5.00000                       PT    57
DELAY/2.00000
FEDRAT/MMPM,199.80000
GOTO  /  25.00000,  40.58750,   1.00000                       PT    58
PPRINT OPERATION NUMBER: 21
RAPID
GOTO  /  25.00000,  24.52500,   1.00000                       PT    59
RAPID
GOTO  /  25.00000,  24.52500,  -4.75000                       PT    60
PPRINT OPERATION NUMBER: 22
SPINDL/RPM,500.00000,CLW
INTOL /     .02500
OUTTOL/     .00000
AUTOPS
INDIRV/   1.00000,     .00000,     .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,     -4.75000,$    CIR    1
      .47500),ON,2,INTOF,$
             (LINE/     25.00000,     25.00000,     -4.75000,$
                        25.00000,     24.52500,     -4.75000)
GOTO  /  25.00000,  18.17500,  -4.75000                       PT    61
AUTOPS
INDIRV/   1.00000,     .00000,     .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,     -4.75000,$    CIR    2
      6.82500),ON,2,INTOF,$
             (LINE/     25.00000,     25.00000,     -4.75000,$
                        25.00000,     18.17500,     -4.75000)
FEDRAT/MMPM,300.00000
GOTO  /  25.00000,  18.17500,   5.00000                       PT    62
RAPID
GOTO  /  25.00000,  24.52500,   5.00000                       PT    63
FEDRAT/MMPM,49.99800
GOTO  /  25.00000,  24.52500,  -7.50000                       PT    64
AUTOPS
INDIRV/   1.00000,     .00000,     .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,     -7.50000,$    CIR    3
      .47500),ON,2,INTOF,$
```

```
              (LINE/     25.00000,    25.00000,     -7.50000,$
                            25.00000,    24.52500,     -7.50000)
GOTO  /  25.00000,  18.17500,  -7.50000                              PT    65
AUTOPS
INDIRV/   1.00000,    .00000,    .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,     -7.50000,$    CIR    4
     6.82500),ON,2,INTOF,$
              (LINE/     25.00000,    25.00000,     -7.50000,$
                            25.00000,    18.17500,     -7.50000)
PPRINT OPERATION NUMBER: 23
RAPID
GOTO  /  25.00000,  18.17500,  -1.50000                              PT    66
RAPID
GOTO  /  25.00000,  31.67500,  -1.50000                              PT    67
RAPID
GOTO  /  25.00000,  31.67500, -10.50000                              PT    68
PPRINT OPERATION NUMBER: 24
SPINDL/RPM,500.00000,CLW
AUTOPS
INDIRV/   1.00000,    .00000,    .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,    -10.50000,$    CIR    5
     6.67500),ON,2,INTOF,$
              (LINE/     25.00000,    25.00000,    -10.50000,$
                            25.00000,    31.67500,    -10.50000)
GOTO  /  25.00000,  31.82500, -10.50000                              PT    69
AUTOPS
INDIRV/  -1.00000,    .00000,    .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,    -10.50000,$    CIR    6
     6.82500),ON,(LINE/     25.00000,     25.00000,    -10.50000,$
                            25.00000,    18.17500,    -10.50000)
AUTOPS
INDIRV/   1.00000,    .00000,    .00000
TLON,GOFWD/     (CIRCLE/     25.00000,     25.00000,    -10.50000,$    CIR    7
     6.82500),ON,(LINE/     25.00000,     25.00000,    -10.50000,$
                            25.00000,    31.82500,    -10.50000)
PPRINT OPERATION NUMBER: 25
RAPID
GOTO  /  25.00000,  31.82500,   4.00000                              PT    70
RAPID
GOTO  /     .00000,     .00000,   4.00000                            PT    71
RAPID
GOTO  /     .00000,     .00000,    .43800                            PT    72
FINI
```

Vita

David Aber obtained a Bachelors of Science degree in Mechanical Engineering from Clarkson University. David accepted a position in Product Manufacturing Engineering with International Business Machines Corporation. Industrial assignments have involved CAD/CAM Applications development with both Site and Corporate responsibility relating to product development, design, and manufacturing. He joined Lehigh University where he earned a Master of Science Degree in Manufacturing Systems Engineering.

Upon returning to IBM, David accepted a position as Generative Numerical Control Project Technical Leader. This position provided the industrial environment to continue his numerical control research effort. The generative numerical control project was the result of this effort. Within this project, David held the sole worldwide responsibility for a technically accurate, comprehensive, and complete solution which defines the generative numerical control parts programming technology. This effort was the most significant undertaking of manufacturing technology development in recent years. The contribution made is the definition for next generation numerical control part programming solution and a broad spectrum of manufacturing applications. He is currently a Senior Engineer with worldwide responsibilities in this area of advanced numerical control.

David is a candidate for the degree of Doctor of Philosophy in the field of Industrial Engineering with a major in Manufacturing Engineering.