COMPUTER AIDED PROCESS PLANNING: THE IMPLEMENTATION AND EVOLUTION OF CAPP SYSTEMS

by

Ronald W. Wilms

B.S., Mechanical Engineering,
Kansas State University, 1981

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTERS OF SCIENCE

Department of Industrial Emgineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

Approved by:

Grosh

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

"Process Planning covers a number of functions in most firms, the main ones being: component routing, method description, time generation, standards creation/ maintenance, and NC programming. Each of these involves the preparation of documentation that is used in the instruction of the people involved in manufacture; in other words, the definition of how to make things." (Blore, D., 1984)

The major function of process planning is, the creation or modification of plans on "how to make things"; sounds fairly simple! However, the creation of plans requires a substantial amount of detailed knowledge. This includes: current information of the facilities within the company and often outside the company (sub-contractors, vendors, etc.), the physical capabilities of all the machines in the plant(s), types of tooling and fixtures available, production rates, and tolerance requirements. Also required is a knowledge of manufacturing methods, how things are made, how they can be held and moved, and what should be done first.

The planner must assimilate all of this information and knowledge to create the process plan consisting of: routing documents that show the different processes through which the component passes, a methods sheet showing the detailed method of manufacture at individual operations, the sequence of cutting, the machine speeds/feeds, specific tooling and specialized processes.

Obviously, the creation of a process plan requires a great deal of information and knowledge. This "information base" problem is amplified as new technology emerges or machines and processes become obsolete, break down, or are no longer available. The highly dynamic nature of manufacturing today requires constant changes and evolution of the product lines manufactured. Process planning encounters another significant problem; that of the high clerical content in most of the functions. It is estimated that between 45% and 80% (Granville, C.S., 1986) of process planning is of a clerical nature. That means that no less than half of the engineers time is spent performing clerical tasks.

The clerical nature of the records problem becomes compounded when there is a lack of maintenance and updating of the records. Often times existing plans are simply modified or parts of one used. Consequently, when references are made to these records the application of the information in them can result in poor or even inaccurate process plans. Sometimes, simply finding the existing process plans is difficult. This is especially true when previous process planners retire or leave the company.

Finally, process planning suffers from inconsistency. Each process planner has a different knowledge or data base. Each remembers different plans to modify or different machining processes to accomplish the same task. The end result is a variety of different process plans for the same or similar parts.

Most often referenced in literature is an example of following nature. Four process planners are asked how they would implement the drilling of a 40mm hole.

1) Drill 35mm, Drill 40mm
2) Drill 20mm, Drill 38mm, Bore 40mm
3) Drill 40mm
4) Drill 39mm, Bore 40mm

Each of these is very feasible. Planner #2 had experience in an industry which required close tolerances while planner #3 was less precise. This demonstrates the effect of the planners' background or "database". Nearly every piece of literature that discusses inconsistencies of process planning uses the saying "ask ten process planners how to make a part and you will have ten different process plans."

THE EVOLUTION OF CAPP

Advances in computer technology have entered manufacturing in the form of Numerical Control (NC) machines and more recently, Computer Aided Design (CAD). The use of an extensive database, a large amount of clerical manipulation, and a method of consistently producing process plans makes Computer Aided Process Planning (CAPP) the next logical step for the manufacturer. It is sometimes referred to as simply Automated Process Planning (APP).

Process Planning was performed manually by all manufacturers well into the 70's. The problems of: 1) retiring process planners and a resulting loss of "expertise", 2) time consuming and error prone clerical work, and 3) inconsistent and duplicate process plans were, and in many cases, still are prevalent. In the early 1970's, computer database storage capabilities and computational powers started becoming available, affordable, and to a great extent, merely practical. Industry has moved slowly into CAPP since this time.

Five stages of Process Planning development have been distinguished by Frank A. Logan, president of Logan Ltd. (Natick, MA). His company is a leader in advanced Artificial Intelligence CAPP type systems and the founder of the advanced LOCAM process planning system. (Logan, F.A., 1986) These levels vary in their degree of sophistication and contain a significant amount of overlap when compared to existing systems.

This same concept of varying levels (although different than Logan's) of development is used here to show the advances in process planning. This paper discusses the types of CAPP systems available today, their evolution, and the problems of implementation. Each stage of development is described and referenced to a discussion of an existing system in use today.

The primary bases for all of the existing CAPP type systems require some use of group technology in the form of part classification and coding (GT/CC). Each system varies in the methodology of GT/CC use. This topic is an important aspect of CAPP, and therefore, is discussed in the following section.

GT/CC

There are many definitions of Group Technology (GT) and they are continuously changing as the scope of GT changes. Much of the original work in this field started in the Soviet Union during WW II when "like" machines were grouped together and moved east to avoid capture by the Germans. One of the first major publications on the subject was by the Russian, Mitrinov, in 1959. However, group technology/classification and coding systems intended specifically for design and manufacturing are a relatively recent development.

This early, but sound, concept of group technology evolved as it moved from Asia into western civilization. V.B. Solja (nationality not mentioned in reference) defined GT in a broader sense. "Group-Technology is the realization that many problems are similar and that, by grouping together similar problems, a single solution can be found to a set of problems, thus saving time and effort." (Halevi, 1980, p. 77) And, in "Engineering," (1968) Group-Technology was defined as "... the technique of identifying and bringing together related or similar parts in a production process in order to utilize the inherent economy of flow production methods." (Halevi, 1980, p. 77)

Conventional machine shops generally have similar machines or types of machining operations grouped together. However, if parts are to be manufactured utilizing the advantages of group technology, the physical layout of the plant must be realigned. The similar design attributes of a group or family of parts also requires similar processes and manufacturing sequences. Thus, the machines can be arranged in a production line by common machining operation sequences.

This flow line type of operation is most commonly found in the form of cells or U-shaped production lines. The machines in the cell may vary significantly, ie: from a lathe to a vertical mill, but are grouped together for a general machining sequence. In this manner, the benefits of mass production are realized by combining several small batches of similar parts, thus making a larger and more economical production lot. Additionally, this method increases manufacturing efficiency by reducing the number of moves and the distance materials must be transported.

As can be seen, the idea of GT has changed since that originally defined by the Russian, Mitrinov. It is no longer a grouping of similar machines, but rather, a grouping of similar production sequences. In today's manufactuirng terms, group technology is defined as "...a technique for manufacturing small to medium lot sized batches or parts of similar process, of somewhat dissimilar material, geometry and size, which are produced in a committed small cell of machines which have been grouped together physically, specifically tooled and scheduled as a unit." (Rembold et al, 1985)

Group Technology and Classification Coding (GT/CC) are key elements for the successful implementation of any CAPP system. The task of classification and coding is mentioned here second. However, in many instances, this important task is performed before the plant layout is changed. Parts must be classified according to appropriate characteristics and a meaningful code assigned. It then becomes a relatively simple matter to use the code to retrieve or group parts according to similar characteristics or manufacturing sequences.

The problem: choosing an appropriate set of characteristics and a good scheme so that the needs of all users of the system are served. (Schaffer, 1981) This includes design engineers, planning/control, manufacturing/tooling, management, etc. A design engineer may want the code to describe specific features of the part, whereas a process planner may want the code to describe the process or routing of the part.

Classification is the procedure of arranging items into groups according to some principle or system whereby like things are brought together by virtue of their similarities, and then separated by specific differences. A code can be a system of symbols in which numbers or letters, or a combination of numbers and letters are given a certain meaning.

There is no universal classification and coding system that can be directly applied to all Group Technology/Computer Aided Process Planning (GT/CAPP) systems. Most GT/CAPP approaches have been implemented with GT/CC systems developed for the specific needs of an organization; or, existing classification and coding systems have been adapted for a specific purpose. In fact, most commercially available schemes provide a means for tailoring them to the unique needs and conditions of the user.

There are many types of GT/CC systems. Each fall into a variety of categories, such as functional or descriptive, qualitative or quantitative criteria, design-oriented or production-oriented, hierarchical or chain-like (discrete) structure, separate codes vs composite codes, long vs short codes, etc. However, in most cases, each system uses combinations of these features in one way or another, thus making it difficult to compare the systems.

Whether it is a so-called universal or tailor-made system, it should be adapted and modified to meet specific needs and requirements of the company. It is therefore necessary to perform a comparative evaluation of the currently available systems and evaluate them based on the needs of the company. It may be determined that an entirely new classification/coding system needs to be developed.

The three classification codes that are most prevalent today are monocodes, polycodes or hybrid classification/coding systems. A monocode system is similar to a hierarchical tree structure and is probably the oldest scheme. The value of the first digit position identifies the highest level group. The second digit divides that group into smaller groups based on a set of discriminating characteristics. The remaining digits continue to divide the previous set in a similar manner. During decoding, the code number must be read from left to right, understanding each digit to discover where to go on the information tree.

The poloycode classification system views the entire population of parts to be classified and includes a list of questions about each part's characteristics or attributes. The answers to questions are recorded in a consistent order and the results fitted into code digit values. The major distinction between a polycode and a monocode is that in a polycode, the interpretation of a character in a given position is independent of any other digit. (Schaffer, G., 1981)

Most industrial coding systems use a hybrid construction that combines the best features of both monocodes and polycodes. To reduce the length of a strict polycode, the first digit of such a system may split the population into appropriate subgroups, as in a monocode structure. Then each subgroup can have its own polycode structure. Thus, within each of these shorter polycodes, the digits are independent of each other. Such an arrangement makes the coding system appropriate for design retrieval while also serving many manufacturing needs. An example of such a polycode is used in the MICLASS coding program described in Appendix A on MICAPP.

Why all this concern about coding and classification? "Before any of the group technology systems can be used, thousands of parts must be coded by shape, dimension, tolerance, surface finish, chemistry, production requirements and other criteria. The task of taming decades of manufacturing in a cohesive data base often proves daunting" (Stix, G., 1984).

In Styx's article Computers Accelerate Manufacturing, [Computer magazine, December 15, 1984] several company's cost of coding and classification alone are described.

Richard Wambach, coordinator for the U.S. Apparatus Division of Eastman Kodak says it took a team of eight (design and manufacturing engineers, software/hardware specialists, and clerks) two years to establish a fully coded parts database for their 125,000 parts. This was a substantial investment (over $1 million) of money and time. Says Wambach "But the avoidance of redundant design pays for itself five times over each year for the one time initial investment. It's the kind of project that doesn't pay off in three years. You don't see results until the system's online."

In another case of Landus Tool Division of Litton Industries (Waynboro, PA), Vice President of Operations James C. Harris says the division originally hoped to complete a group technology project within 8 months. "We've run over that by more than double. Don't let anybody tell you it's going to be fast and inexpensive." he says. The project was estimated to cost $250,000; it has already exceeded that figure by a factor of 3 and will probably reach the $1 million mark by completion.

These two examples are hard facts that must be faced when starting a classification and coding task. These costs are upfront before the implementation of the CAPP program. So what are these problems with coding and classification? Each author has his own list of major problems. Granville's (1986) list is consistent with most process planners.

1) Code system can be inaccurate. In the code, "exact values" of part attributes are not stored. The part's exact attribute value is fitted into one of several discrete ranges for that value. If a user wishes to retrieve an exact value, all the other values in that range are also retrieved, and the user can get too much information. If the code number becomes too long and therefore seeks a very specific match, often no information will be found meeting the retrieval criteria.

2) Code systems are inflexible and often cannot accommodate new technologies and changes in the product lines. The existing code system may not capture a new product that is twice the length of the old produot because the new product length exceeds the largest value for the length digit. The user is asked to determine a classification scheme today which may be inappropriate tomorrow. This is the "crystal ball" method for organizing data.

3) Code numbers must be applied correctly and consistently. The rules for coding must be followed consistently and with exact discipline or the data collected will be inaccurate.

4) Code numbers are not "user friendly" and must be supported by extensive user training. When coders leave a company, training new coders can be extremely difficult. The problem of retiring process planners was a major advantage of a CAPP system, however, this point still shows the necessity of experience.

5) Code numbers are not transparent. The user cannot readily identify the type of part he's looking for by reading the code number. Not many people can remember the meaning of a 30 digit number.

6) Use of code numbers keeps other personnel, such as designers and purchasing personnel, from easily using the data in the system.

In summary, classification and coding is a vital element for a CAPP system. It requires substantial time and money for it's establishment and maintenance. Even the best designed scheme will have instances when a part will not conform.

The CAPP systems require a classification and coding scheme, a large manufacturing database, and for optimal utilization, the use of manufacturing group technology. So what sets the programs apart?

THE STAGES OF CAPP DEVELOPMENT

STAGE 1, Traditional. The traditional method of Process Planning is the manual method. This is done by using the planner's memory as data storage and retrieval and his clerical skills for the manual production of the actual plans. This method, as the topic of this paper would suggest, is not an automated system.

STAGE 2, Computerization. The second stage of Computer Aided Process Planning is considered to be any computer assisted process planning systems which does not have generative capability. The process plans are selected from existing plans or modifications there of. It is primarily a computer "organization" system.

Computer Aided Manufacturing-International (CAM-I), an organization comprised of companies concerned with manufacturing technology, was charged with exploring the feasibility of a Computer Automated Process Planning system in 1972. In 1974, the first commercially available system was CAM-I's CAPP program. A description of this system is described in a later section. This is a variant system based on group technology techniques.

Variant process planning consists of entering existing part process plans into a computer database. Each part is classified by the user and a code number assigned. The code number supplies a description of dimensions or part characteristics. When a new process plan is required, the existing ones are systematically searched by the computer for a match or near match based on the coding scheme. Either the existing plan is used or it is "varied" to fit the purpose; thus, the term variant process planning.

STAGE 3, Interactive. In this stage, a series of questions are interactivly answered by the planner and the classification code constructed by the computer. The system then automatically selects appropriate keywords and associated parameters to drive the manufacturing logic. These systems are considered either a constructive or advanced variant type of CAPP. Time estimating, methods planning, and time/cost standard sub-routines are also available using interactive questions. The primary advantages of this type of system are the diverse variety of parts that can be entered and the small amount of manufacturing logic required.

The MIPLAN program, found in Appendix A, is an example of this stage of evolution. This system is a variant type process planner with multiple subroutines provided for additional process planning activities. Among the most important is the MICLASS GT/CC system. Questions are answered through an interactive system which automatically derives a code number. Advances and upgrades in this program have made it overlap and also classifiable in later stages as a generative type of system. Similar in capabilities is the CUTPLAN/CUTTECH system produced by Metcut Research Associates Inc. (Cincinnati, OH), (Zdeblich, 1987) The ICAPP system, discussed in Appendix B, is another example of a variant/generative system.

STAGE 4, Semi-Automatic. The previous two stages have used extensive manually operated interactive functions to generate the classification code number and subsequent routing sequences. In addition, feature descriptions have been developed based on operations or elements of operations.

Stage 4 uses advanced logic so that the computer automatically selects a series of features that are used for coding and development of the entire process plan. In these systems, the classification and coding module is an essential subroutine that provides consistent classification and coding. The system then determines or "generates" a process plan using the manufacturing logic database.

This stage of development is referred to as generative process planning. Two examples of programs with these capabilities are presented in this paper. First, is GECAPP, developed by the General Electric Corporation. It is summarized in more detail in Appendix C. The second example of stage 4 development is called GENPLAN. A summary of which is found in Appendix D. It was developed by Lockheed-Georgia, one of the first users of CAM-I's CAPP system. The specialization for Lockheed-Georgia's specific use led to the development of a more advanced, generative type of CAPP system.

At this level of development, the programs provide a substantial number of sub-systems that allow for such things as automatic cost estimating, setting of time standards, accounting information etc. Additionally, the database has grown significantly to incorporate such items and vendor drawings, customer account information, and automated regeneration of existing plans to make use of new technology.

STAGE 5, Automatic. Fully automatic process planning is the progressive expansion of Stage 4 coding where the automatic coding system contains all required manufacturing information. This level is NOT completely practical nor possible with present technology. In the not too distant future, the part drawings will be made on a CAD system, the process planner program invoked, the part automatically coded and the process plan produced.

Technology, thus far, has used expert systems to interact with all of these various functions. Systems approaching this level are the LOCAM system, described in Appendix E, or the CMPP system, described in Appendix F.

Another system being developed by the CimTelligence Corporation for Northrop Aircraft Division, is called Intellicapp. It is sometimes referred to as an Artificial Intelligent CAPP (AICAPP) system. It is a GT/CAPP based system that links various functions together by an "expert system". It captures exact values for classification attributes and expresses them in common words to the user. In addition, IntelliCapp utilizes a natural language and voice interface. This type of system is currently on the leading edge of technology.

IMPLEMENTATION OF CAPP

Good planning is the key to success for any project of significant magnitude. This is most certainly the case when implementing a CAPP system. "I didn't plan to fail, I failed to plan." (Granville, 1986) The CAPP systems are very difficult to justify using standard accounting payback procedures, let alone implement. Planning is an essential step as the Automation of Process Planning requires a long term commitment to Computer Integrated Manufacturing (CIM). Thus, there must be a solid commitment of time and money by management.

The benefits of consistent plans, faster and more accurate quotations, less errors/scrap, are all possible if a CAPP system is properly planned and implemented. Even with good planning, high cost overruns, lots of manhours and two to three times the "anticipated" completion times are common among the users of CAPP systems. (Stix, 1984) Once management support is given, it is imperative to continually keep them informed along each step so that the obstacles incurred can be overcome.

The first step is to clearly define the scope of the project. What do you hope to gain? What are your needs, now and in the future? Examination of the existing method(s) is a good place to start. A bit of brain storming and research can help for the development of your "wish list" - those things that would be nice to have but may not be necessary or justifiable at this time. Adding onto hardware or software capabilities at a later date may not be practical or possible.

A former colleague of mine is a computer programmer with an engineering background and over 15 years of experience. He indicated the biggest problem with engineers in computer programming, is that they see a portion of the completed system and say "that is really great". In the same breath they invariably add, "wouldn't it be neat if we could add this or that." My point is, that adding capabilities to programs or hardware when they were not designed for is often difficult, time consuming, and can be expensive. Simply, you need to plan ahead.

So what are the problems you are trying to cure?

- Inaccurate Information
- Incomplete Information
- Duplicated Plans
- Inconsistent Plans
- No method set up for: Cost Analysis
  Time Standards
  Tooling Inventory
- High Cost of: Engineering Time
  Machinist idle time
  Scrap and Rework
  Missing deadlines

These are but a few of the common ones mentioned. While considering the scope of the project and the problems of the present system, don't forget to think about the different groups of people in the plant that the new CAPP system will effect. Not only the engineers doing the process plans, but also the draftsman, design engineers, the clerical staff, the accounting department, the sales people for determining quotes, and especially, the upper management.

Defining the broad scope of the project is an important first step. The next step, by no means less important, concerns the various technical considerations. The heart of a computer automated system is of course, a computer. Both the hardware and software are extremely important aspects. These must be considered together to form a complete system.

Experience was gained first hand while attempting to install and use CAM-I's "generic" CAPP program at Kansas State University. This is discussed in detail in the next section. Let it suffice to say here, that, continually changing hardware and software on the university computing system prevented the same program that was operational in 1981 from being readily usable only five years later. The current rapid advances in electronic/computer technology and system software can be a major problem and deserves significant consideration.

Hardware, silicon chips and copper wire, are considered here first. Several questions need to be answered.

- What does the company currently have?
- Will it have the storage capacity and CPU time available for running your CAPP system?
- Do you want a dedicated system?
- Does the system undergo periodic upgrades?
- Will the CAPP program you want operate on the existing system?

The next consideration is the software or CAPP program. In some instances, the choice of programs may dictate the choice of hardware or vise versa. There are a wide variety of programs in use today. Due to the sure massiveness of these systems, I recommend that commercially available packages be explored first. Following is a partial list of available programs found while research was conducted for this paper.

LOCAM by Prime Computer Inc. (Natick, MA) a generative expert system for generic use.

METCUT Research Association Inc. with CUTPLAN, CUTTECH, AUTOPLAN, MultiCAPP, (Cincinnati, OH) a group of generic, generative CAPP systems.

ICAM by the U.S. Air Force, (USAF Material Lab WPAFB, Dayton, OH), generative system for aerospace parts.

CAPE by Garrett Turbine Engine Co. (Phoenix, AZ), a variant/generative system for jet engines.

APPAS by Purdue University (Lafayette, IN), generative system for machining centers.

ACAPS by Pennsylvania State University (University Park, PA), generative system for turned parts.

EXAPT from West Germany, Interactive system for pressure vessels.

WICAPPS by Westinghouse Defense and Electronics Center (Baltimore, MD) variant system for electronics manufacture.

ICAPP by the University of Manchester Institute of Science and Technology (UMIST), (United Kingdom).

CAPPE by PERA Inc. (Melton Mowbary, England), generative system for NC machining centers.

POPS by the University of Iowa (Iowa City, IA), generic generative expert system.

AUTAP by the Technical University of Aachen (Aachen, West Germany), generative system for NC machining.

Interprogramma' by Software R & D Institute (Sofia, Bulgaria).

RATIBERT and PRODI by the Technical University of Dresden (Dresden, GDR), generative system.

PC compatible CAPP Systems:

LETS-MB by Tipinis Associates Inc. (Cincinnati, OH), for Layout, Estimation, Tooling & Design, and setup for Multiple Spindle Bar Automatics, uses and APPLE II C.

Micro-CAPP and Micro-GEPPS by Pennsylvania State University (University Park, PA), an interactive constructive system for machined parts, uses the Japanese KK-3 coding system, for use with IBM compatibles.

FALK CAPP by Falk Corp. (Milwakee, WI), process plans & NC tape generation for Chucking Machines, uses an APPLE II C.

DREKAL by Massachusetts Institute of Technology (Cambridge, MA), for cutting machinery, uses IBM compatibles.

Group Technology / Classification Coding Programs:

MICLASS by Metcut Associates, xref METCUT.

DCLASS by Bringham Young University (Provo, UT).

MultiClass by ORI, xref MIPLAN program.

SAGT by Purdue University (Lafayette, IN).

KK-3 from Japan.

OPT by Creative Output International (Israel).

CAMAC by the University of Ashton (Birmingham, U.K.).

This list is by no means exhaustive, but it shows the variety of names, organizations, and countries involved with CAPP systems. Notice that a number of these programs are for a specific purpose; ie: machining centers, cylindrical parts, etc. This specialization of the programs significantly reduces the scope, and thus, the amount of machining logic, data base, and programming required to develop an operational system.

One prevailing problem is the proprietary nature of many of the systems in use. They are customized for a single manufacturer and are not transferable or available to others. The problem of the software/hardware transportability was evident in the difficulties incurred during the implementation of CAM-I's CAPP program at Kansas State University. System software, available terminals, and other things of this nature prevent CAPP systems from being more widely used.

An interesting item was noted during the making of the list of available CAPP systems. There were no Japanese programs noted in any of the over 80 articles researched for this paper. The sole exception is the use of the KK-3 coding and classification program. This I surmise is due to one of two things. Either they are not using CAPP systems in Japan, which seems unlikely, or they are doing a fine job of keeping the information to themselves. Any final conclusions are left up to the reader.

After exploring the available CAPP systems, the decision of buying a system versus writing an in-house program has to be made. In either case, a good computer scientist is needed. A computer science colleague of mine once pointed out that writing a program requires a programmer; but getting it to work on existing hardware and software can definitely be a science, thus the need for a computer scientist.

Even a purchased package will require customization for a specific manufacturer. This will either have to come from the vendor, or from within the company. The computer scientist's time must be included when considering the cost of implementing a CAPP system. In addition, it will have long term effects on cost and the time required to get the system up and running and can dramatically effect the resulting benefits.

When considering a program, to purchase or write, a major concern will be the coding and classification system to be used. At several large manufacturers, this portion alone of implementing a CAPP system took nearly two years. An excellent article on the costs, time considerations, and benefits of CAPP is Computers Accelerate Manufacturing by Gary Stix. [Computer Decisions, Sept. 15, 1984, pp 45-58].

The hardware, classification and coding system, and the purchase or in-house development of the program comprises the main considerations. However, there are a few others that should not be left out. First, if a package is purchased, will it work as sold? If not, what and how many modifications are required?

Next, consider the system data storage capabilities. It has been said that a CAPP system is the manipulation of a huge database. Will the system be able to hold all current and future plans? What about future expansion?

Along these same lines, what about other capabilities. Will the hardware interface with present or future CAD systems? Can global changes in the process plans be made as new technology is brought into the plant? Does the system keep a record of changes and revisions of the plans for later tracking?

After each of these topics are evaluated separately, they must all be put back together into a total system and measured against the original scope of the project. Are the hardware and software systems compatible? Does it meet the current needs? Can it be modified to meet realistic future needs?

A number of points have been brought up and most certainly, there are more to consider. Implementing a CAPP system is no easy task. The most important considerations are summarized as follows:

1) Commitment of management to CIM
2) Planning, short and long range
3) Computer hardware and software
4) Time, Cost, and benefits

These central issues are essential for the successful implementation of a CAPP system.

CAM-I'S CAPP

Among the first approaches to a generic Computer Aided Process Planning system is the CAPP system developed under the sponsorship of Computer Aided Manufacturing-International Inc. (CAM-I). This organization is an industry supported manufacturing entity for the development and distribution of manufacturing technology. CAM-I's CAPP is a variant system that was developed in 1973 primarily to demonstrate the feasibility of computer automated process planning.

This system, like all others, has some very specific hardware requirements for implementation. The user's manual for the CAPP program specifies that the system was "designed to operate under a computer time-sharing system or an operating system with multiprogramming capabilities. In the latter case, each CAPP terminal and user must be allocated a certain portion of main computer storage and dedicated central processing unit service. The basic CAPP system is designed for an IBM 360/370 system with TSO (time-sharing option). The display terminal device supported is the Hazeltine, Model 2000". (Modifications are documented for the use of a Hazeltine, 1500 or 1510 series due to the unavailability of the Hazeltine 2000.) The computing system used must include at least one(1) direct access storage device. Deviation from the above hardware configuration will necessitate certain software modification of the CAPP program."

The program was written in FORTRAN 66 with extensive use of characters in integer fields. This design caused difficulty during the implementation of CAM-I's CAPP at Kansas State University. The later versions of FORTRAN (77) specifically designate character fields. In addition, computer cards were the primary means of entering data onto computer systems. For this reason, much of the data entry is set up "like" computer cards and is location and length specific. The program is provided in 3 files, 2 in IBM ASSEMBLER source code and 1 in FORTRAN source code. The FORTRAN source code must be compiled using the level H or H-extended compilers. Either the F or H level assemblers may be use for the assembler language code.

The bases of operation for this program, like most CAPP system, is group technology methods of classifying and coding parts. This CAPP program does not require any specific coding system. It provides for up to a 36 position code identification scheme. The code may be that of the manufacturers exiting system or any other system of choice.

A substantial data base is needed for this and any other CAPP type system. CAM-I's CAPP requires six data

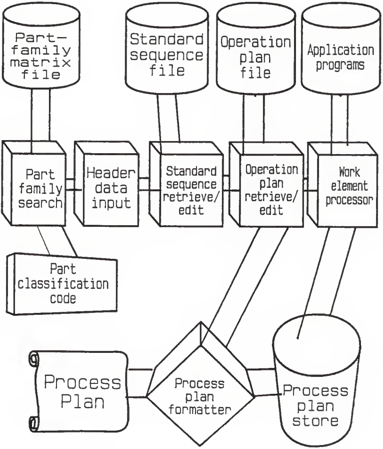


Figure 1. CAM-I's CAPP System

structure files: part-family matrix file, standard sequence file, Operations-Code table, Operation-Plan file, part-family setup file, and process-plan store file. See Figure 1. Each of these files are used for various functions of the program that are accessed through an interactive menu driven system.

The part family matrix file is used to represent part families as "matrix" structures. This file establishes the coding system of the manufacturer and must be established by the user. Thus, substantial effort and time is required to classify all of the parts in a given manufacturing facility and establish a coding system. This system allows for easy computer search and file storage techniques. See Figure 2.

The standard sequence file establishes a "standard" or general sequence for the manufacture of a part family. This "standard" plan is then modified or "varied" for each specific part, thus the term variant CAPP. This file is a database input that must be entered before the program is of use. The standard plan in this CAPP system is a sequential set of instructions that include general processing requirements, tools, machines and detailed operation instructions. These "standard" plans are grouped in a series of very similar parts or families.

CODE DIGIT										DESCRIPTION
1	2	3	4	5	6	7	8	9	10	
1										GENERAL CODE DESCRIPTION
	2									RAW MATERIAL FORM
		3								MATERIAL SPECIFICATION
			4							THICKNESS
				5						MAJOR MEASUREMENT
					6					MINOR MEASUREMENT
						7				EDGE PREPARATION
							8			PRE-FABRICATION
								9		FURTHER PROCESSES
									10	MACHINING PROCESSES

Figure 2.

Basic Structure of GTCC (Group Technology Coding and Classification) Scheme Developed for the C.G.E. (Canadian General Electric Company's CAM-I CAPP System. (Chang, Robert & Ham, Inyong, 1983)

An Operations-Code table must also be designed and input into the data base before the program may be used. The construction of standard plans for the CAPP system requires this data in order to identify, normalize, and standardizes the spectrum of manufacturing operations performed in the fabrication of machined parts at the users' facility. For example, "VMILL" would stand for "Machine on a vertical mill".

The Operation-Plan file is a sub-file addressed by the operation codes (ie: VMILL). This file permits expansion of the part-specific operation plan. Specific cutting information including speeds, feeds, special fixtures etc. are established in this file.

The part-family setup file is a storage file for plans that are either not complete or have not been approved for use. In general, this is a working file.

In operation, a part-classification code is entered into a part-family search routine. The system then systematically interrogates the part family matrix file for a matching matrix. If a family match is found, that data is temporarily stored in the part-family setup file to allow for the creation of a new file. This new file identification is established by the user and a description of the product to be planned entered. The entries are made through keyboard entry using element codes. This requires the user to know the element codes and for each manufacturer to develop specific codes for their use. For example, a "Z9" code stands for the order number. "Z9" must be entered followed by "/(order number). This series of information is denoted as header data. See Figure 3.

```
HD/HEADER,AO,PARTNO=**********,A1,ENG*CHG=***,
A3,PART*NAME=**************,A5,CLASS*CODE=*******,
A6,PLNR*INIT=******,A8,DATE=*******,BO,TYPE*PLAN=***,
D3,MK/BUY*CODE=***,D4,SPECILA*INST.=***********,
Etc.
.
.
.
```

Figure 3.
Sample Header Data input entry.

After the header data is entered, the system allows retrieval of a standard sequence of user dependent operation codes called OPCODES. "VMILL" is the representation for "machine on vertical mill" from the established OPCODE table. This system thus requires considerable knowledge on the users part to be able to look at the computer screen and make intelligent use of the information. This sequence of OPCODES forms the user-defined standard sequence for the part family. The planner may then edit or modify the data for the particular part being planned.

After the OPCODE sequence has has been edited, each individual operation code can be retrieved from the Operation-Plan Data File and edited to be specific for that particular part. The completed information is then stored in the process planning store file to be retrieved and printed when needed.

Several companies have started out with this basic shell program and tailored it for their individual needs. CAM-I's last release was revision 3.1 (1982) which was modified for the Canadian General Electric Company. This version included some hardware modifications and changes in the part-family search algorithms.

IMPLEMENTATION OF CAM-I'S CAPP AT KSU

During the spring and summer of 1987, an attempt was made to install and use CAM-I's CAPP version 2.1A at KSU. A number of the problems discussed in the previous section on problems with the transportability of CAPP type programs were incurred. Admittedly, this author was not tremendously familiar with main frame systems and THE FORTRAN programming language. However, this can occur in industry as well!

First, a short background on the acquisition of CAM-I's CAPP program at KSU. CAPP version 2.1A in the form of magnetic tape was acquired by the Industrial Engineering Department sometime during 1980. Shortly thereafter, work proceeded to get the program up and running on the KSU mainframe system. The program was reported to be very near operational when the student working on it graduated. The program was not touched until the spring of 1987, when another attempt was made. All of the work previously performed on the program was lost by a combination of a routine deletion of unused files by the university computing center and by changes in university staff.

It was understood and documented that the program was written for and IBM 360/370 series operating system. None the less, an attempted was made to load the program onto the engineering departments Harris 8685 Computer using a VOS (Virtual Memory Operating) system.

It became readily apparent that this was not going to work. First, two of the program files are written in IBM source code which is not compatible with the Harris system. This was not an insurmountable problem, however, a great deal of reprogramming would be required. Secondly, the Harris system operates on a 24-bit processor system. The IBM is a 32-bit system. The logic scheme of CAM-I's CAPP system uses a great deal of "half words" (8-bit fields, or 4 "half words" per computer word) for the data matricies, thus requiring a 32-bit processor. A tremendous amount of the program logic would have to be changed. The idea of using the Harris system was thus abandoned.

The tape of CAM-I's CAPP was then loaded onto the University main frame system. KSU currently uses an NAS 6630 computer system in CMS (Conversational Monitor System) mode. Program editing and debugging was performed using a standard RS-232 monitor (Selanar Hirez 100XL).

A substantial number of modifications and error corrections had been published since the university acquired the magnetic tape of the program. A number of these were from CAM-I, other corrections came from the

Lockheed-Georgia Company (Marita, Georgia) and the McDonald Douglas Aircraft Company (Saint Louis, Missouri); users and developers of the program. In all, over 400 corrections. Additionally, another 131 changes were necessary to use the Hazeltine 1510 terminal as the program on tape was developed for a Hazeltine 2000 terminal. This is an important issue because the program is written to be terminal specific.

After these corrections were made, the FORTRAN source code (Tape File 1) was compiled using the VS-FORTRAN, version 4.1, level G compiler. It compiled without any error messages.

The ASSEMBLER Source code (Tape File 2) was the next file to work on. In the implementation section of the CAPP manual, it indicates that "FORTRAN source must be compiled using the level H or H-extended compilers." The extent of this "must" was not known, so work continued. The first major error encountered was CPU (Central Processor Unit) control of the program.

This program was originally written for a TSO (Time-Sharing Option) mode of operation. The program used a continuous call to the CPU while waiting for the program user to enter data. This is found in the subroutine BREAKR. This method ties up a tremendous amount of CPU time, a highly undesirable feature for a university system with several thousand users. This procedure was written out of the code with the help of the KSU computer consultants.

The next errors encountered were code differences from the use of a level G versus a level H or H-extended compiler. The read (TGET) and write (TPUT) statements had to be replaced with RDTERM and WRTERM statements respectively. The university system no longer carried the level H compiler because of cost, and additionally, had no documentation as to the differences of these instruction codes.

The fourth error was found in the installation routine INSTLN. FORTRAN 66 allowed for characters in numeric fields. However, this compiler responded with an error code of severity 12 (must fix error) to the initialization of several fields. These fields were merely being initialized to zero, thus the hexadecimal code for zero was written into the source code and the problem resolved.

The Assembler (Tape File 2) and Utility (Tape File 3) Source code files were then compiled. A large number of a particular WARNING message still remained in the compilation. "ERROR 1195 (W) Either one or both operands of a relational expression are of logical type.

This is a non-standard usage which is allowed for LANGLVL 66 only (language level 66)." The extent of this warning code was of major concern. However, all three of the program's files and file 4, a sample data file, were executable.

The program files were compiled and linked for usage with the Hazeltine 1510 terminal. The Hazeltine was set up as recommended in the CAPP and Hazeltine manuals:

```
Baudrate: 9600
Parity:   Even
Full:     Dup  (Full Screen mode)
Case:     Upper Case Letters
EIA:      Standard RS-232 Communication
TERMINAL: Line Size OFF  (prevents word "wraping"
                          for lines with less than
                          80 characters)
```

The program was executed with high hopes which were soon dashed. Portions of the main menu would come up on the screen, but were not completely intelligible. This was not of immediate concern, however the lack of screen control was. It was not possible to log off or get out of the program. This had to be done by the dispatch operator.

Computer consultant services were again sought. Using a technique which allows all terminal control codes along with the expected terminal response information to come up on the screen (a transparent mode), it was found that the screen control was not coming back to the terminal. The Hazeltine type of terminal uses screen control functions that are created by the program in the CPU and sent to the terminal.

This method is not prevalent in use today. The computing department at KSU uses an IBM 7171 Communications Controller to interface a number of ASCII type terminals to the main frame computer. This communication controller contains information that translates terminal specific ASCII code into IBM code (EBCDIC) and from the CPU in EBCDIC back to the terminal specific ASCII code. The Controller is not programmed for translation of the Hazeltine 1510 or 2000 series terminals.

Again, this problem was not insurmountable. The system has available a "Cluster Controller" that does not stop or filter the terminal control information. Work continued! It was then discovered that the CAPP program provides a means for this character conversion in the Utility Source code (Tape File 3). A "core dump" was provided by the computing center that showed the systems ASCII to EBCDIC and visa versa conversation tables.

The conversion tables were in need of a significant number of corrections. Any time a program and terminal combination of this nature is used, the conversion portion of the program must be checked for correctness. These changes were made.

Once again the program was compiled, linked and executed. Somewhat better results were achieved. The line and character spacings were not correct, but there was some terminal control; at least a log off could be achieved! In addition, three of the menus could be brought up. However, they also had the same scrambled appearance.

The problem of menus being scrambled was investigated. All of the titles and menu designs in the program are written in Hexidecimal code. This made it very difficult to determine where the errors were in the code. The program was written in this manner for a specific purpose. The menu information is location specific so that as input information is entered, it can be "overwritten" on the unused bottom portion of the screen.

Further investigation into this problem found that inconsistent letter and number transpositions were occurring in the menus that were accessible. This proved to be very baffling, especially since the same errors could not always be reproduced.

Several days of investigation turned up nothing. It was decided to return to earlier problems and investigate their effects on the present stage. The extent of the "must" in using a level H or H-extended compiler on the Source was suspect, especially due to the warning messages that occurred when compiling the source code. It was finally found in the VS FORTRAN Program Guide, pg XV, Ref. Release Notes. "If the program either references or defines a user program that has a character-type argument or is itself of character type, it must be compiled using VS FORTRAN v 3.0 .

Once again this term must occurred. The KSU system no longer supports version 3.0. It changed over to v 4.1 in 1984, and abolished the use of version 3.0 in 1986. Two system changes, the elimination of the level H compiler and the change in VS FORTRAN versions caused major roadblocks in the implementation of this program.

The attempt to implement CAM-I's CAPP v. 2.1 at KSU was terminated as the necessary requirements to complete full implementation were beyond the scope of this project. It was concluded that the program, in its present state, is not readily implementable at KSU. In order for this CAPP program to operate on the existing system, the following recommendations are made as possible solutions to these substantial problems:

1) Using the existing system software and writing a subroutine in the CAPP program to convert the character to numeric fields. Additionally, further research as to the differences and changes necessary to use the level G compiler and rewriting portions of the CAPP program to use the CMS mode instead of the TSO mode.

2) Rewrite the entire program in FORTRAN 77 and make the changes to a CMS mode.

3) Spend the money to get the VS FORTRAN version 3.0 and the level H compiler back on the system.

Even with these changes, The problems caused by the terminal specificity could still occur. Thus;

4) Write out the terminal specific portions of the program.

A tremendous amount of changes have obviously been made in both system software, programming languages and computer terminals since CAM-I developed CAPP in 1973. These changes are not minor and can make the software virtually useless. The lesson learned in this case study is that match between hardware and software is of tremendous importance when implementing a program of this nature.

BIBLIOGRAPHY

Blore, D., (1984), "Computer Aided Process Planning", Chartered Mechanical Engineer (GB), vol. 31, no. 5, May 1984, p. 31.

CAM-I Inc., CAPP 2.1 Users Manual #PS-76-PPP-03, CAM-I Inc., Arlington, TX., 1976.

Chang, R. and Ham, I., (1983), "A CAM-I/CAPP Implementation with Inhouse Developed Group Technology Classification and Coding System", Proceedings of the 20th Annual Meeting and Technical Conference, Numerical Control Society (NCS), Cincinnati, OH., 10-13 April, 1983.

Cochran, W.D., (1984), "An Approach to CAPP Cost Justification", AUTOFACT-6 Conference Proceedings, Anaheim, CA., 1-4 Oct., 1984.

Ellis, T.M.R. and Semenkov, O.I., (1982), "Advances in CAD/CAM", Proceedings of the 5th International IFIP/IFAC Conference on Programming Research and Operations Logistics in Advanced Manufacturing Technology, PROLOMAT 82, Leningrad, USSR, 16-18 May, 1982.

Evans, S. and Sackett, P.J., (1982), "Financial Justification of State-of-the-Art Investment: A Study Using CAPP", Engineer Management International (Netherlands), vol. 1, no. 3, Dec. 1982, pp. 221-25.

Evans, S. and Sackett, P.J., (1984), "Flexibility in CAPP Through System Design", CAD '84, Proceedings of the 6th International Conference and Exhibition on Computers in Design Engineering, Brighton, Sussex, England, 3-5 April, 1984.

Eversheim, W. and Fuchs, H., (1980), "Integrated Generation of Drawings, Process-Plans, and NC-Tapes", Advanced Manufacturing Technology, edited by P. Blake, North Holland Pub., 1983, p. 303.

Granville, C.S., (1986), "The Impact of Applying Artificial Intelligence with Group Technology and Computer-Aided Process Planning Systems", CASA/SME Conference, "Building Efficient Manufacturing Operations through Computer-Aided Process Planning", Denver, CO., 4-5 March, 1987.

Gongaware, T., McGuire, P., and Vogel, S., (1984), "CAPP for Electronics Manufacturing", 2nd Biennial International Tool Technical Conference Proceedings, Chicago, IL., 5-13 Sept., 1984.

Grover, M.P., (1980), Automation, Production Systems and Computer-Aided Manufacturing, Printice-Hall, Englewood, NJ., 1980, pp. 559-562.

Halevi, G., (1980), The Role of Computers in Manufacturing Processes, John Wiley and Sons, New York, NY., 1980, p. 76.

Herrell, R., (1987), "Implementing CAPP as a Key to CIM", Proceedings of a CASA Conference, "Building Efficient Operations Through Computer-Aided Process Planning", Denver, CO., 4-5 March, 1987.

Hsu-Pin-Wang and Wysk, R.A., (1986), "Applications of Microcomputers in Automated Process Planning", Journal of Manufacturing Systems, vol. 5, no. 2, 1985, pp. 103-111.

Hughes, D.R., and Maull, R.S., (1985), "Software-The Key to Computer Integrated Manufacture", Effective CAD/CAM '85, IMech Conference Publications 1985-7, Cambridge, U.K., 1985, pp. 15 - 22.

Kochan, D. editor, (1984), "Integration of CAD/CAM", Proceedings of the IFIP 5.2/WG 5.3 Conference, Gaussig, GDR, 8 - 10 Nov., 1983.,

Lesko, J.F., (1983), "MIPLAN - Implementation at Union Switch and Signal", Proceedings of the 20th Annual Meeting and Technical Conference, Numerical Control Society, Cincinnati, OH., 10-13 April, 1983.

Logan, F.A., (1986), "Evolution to Automatic Process Planning by Generative Expert System", CASA/SME Conference, "Building Efficient Manufacturing Operations through Computer-Aided Process Planning", Denver, CO., 4-5 March, 1987.

Logan, F.A., (1983), "Process Planning - The Vital Link Between Design and Production", Proceedings of the 6th Annual Conference on Design Engineering, Birmingham, England, 4-6 Oct., 1983. (xref Tulkoff, 1985).

Love, D.M. and George, A.R., (1985), "Designing a Computerized Component Classification System to Minimize Implementation Costs", Effective CAD/CAM '85, IMechE Conference Publications 1985-7, Cambridge, U.K., 1985.

Rembold, U., Blume, C., and Dillmann, R., (1985), Computer-Integrated Manufacturing Technology and Systems, Marcel Decker Inc., New York, NY., 1985.

Sackett, P.J., Evans, S., and Shannon, J.A., (1984), "CAPP - The Key to Effective Manufacturing Information", Flexible Manufacturing Systems Magazine (U.K.), June, 1984.

Schaffer, G.H., (1981), "Implementing CIM", American Machinist, August, 1981, pp. 164-68.

Ssemakula, M.E. and Davies, B.J., (1984), "Integrated Process Planning and NC Programming for Prisimatic Parts", Flexible Manufacturing Systems Magazine (U.K.), June, 1984.

Steudel, H.J., (1984), "Computer Aided Process Planning: Past, Present and Future", International Journal of Production Research (G.B.), vol. 22, no. 2, March-April, 1984.

Stix, G., (1984), "Computers Accelerate Manufacturing", Computer Decisions, vol. 16, no. 2, 15 Sept., 1984, pp. 45 - 55.

Tipnis, V.A., (1980), "Computer Aided Process Planning", American Society of Quality Control 34th Annual Conference Transactions, Atlanta, GA., 20-22 May, 1980.

Tulkoff, J. (1987), "New Trends in Computer Aided Process Planning", CASA/SME Conference, "Building Efficient Manufacturing Operations Through Computer Aided Process Planning", Denver, CO., 4 - 5 March, 1987.

Tulkoff, J. (1985), CAPP, Computer Aided Process Planning, Society of Manufacturing Engineers, Dearborn, MI., 1985.

Tulkoff, J., (1981), "Process Planning in the Computer Age", Machine and Tool Blue Book, Nov., 1981.

Waldmen, H., (1983), "Process Planning at Sikorsky", Proceedings of CAD/CAM Technology, Summer 1987. (xref. Tulkoff, 1985)

Zdeblick, W.J., (1987), "CUTPLAN/CUTTECH: A Hybrid Computer Aided Process and Operations Planning System", Metcut Research Associates, Cincinnati, OH. (xref Tulkoff, 1985).

APPENDIX A: MIPLAN

MIPLAN is another variant CAPP program. It was originally developed by the Organization for Industrial Research (ORI) (Waltham, Mass.). It was first implemented at the Lamp Equipment Operation (LEO) of the General Electric Corporation (Cleveland, Ohio). It's initial purpose was to obtain standard times for creating process plans (Steudel 1984). The interactive conversational software was designed to run on various computers including the Digital Equipment PDP-11 family (including VAX), all IBM mainframes using OS or DOS, and GE time-share hookups.

Since it's beginning in the early 1970's, major additions have been made to the program. In 1975, a classification and coding module called MICLASS was added. It now consists of over 15 other modules that each serve a practical process-planning need. Most recently (1981), the system has been integrated with a computer-graphics package by Computervision Corporation (Bedford, Mass). This combined package is called CV-MIPLAN.

The nature of the variant CAPP programs requires an extensive classification and coding system. This is often a significant problem in implementation of the CAPP program. The MICLASS module of this program provides the consistent and systematic group technology classification to make a variant CAPP operate effectively.

MICLASS is a hybrid semipolycode system based on the features, equipment or processes required to manufacture the part. It uses up to 30 digits to deal with the group technology (GT) features. The first four digits describe the main shape, the shape elements, and the position of the shape elements. The next four digits of the MICLASS code classify the main dimensions, the ratio of the dimensions, and an auxiliary dimension.

The ninth and tenth digits classify the part tolerances including dimensional tolerance and surface finish. The last two digits for the main code indicate the part's material and machinability qualities.

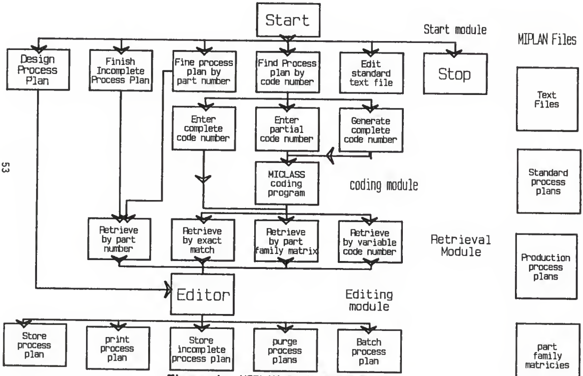
An additional 18 digits are available as a supplementary code to cover specific company related information. This may include lot size, piece time, major machining operations, special heat treating, vendor codes or existing in-house manufacturing data.

The first 12 digits are, in a way, a universal-type code applicable to most companies. According to ORI, it is identical for 99% of its customers. Digits 13 to 18 tend to apply universally to 50% of its customers while only 10% of its customers use the same arrangement for digits 18 through 30; that's where the code is really customized (Schaffer 1981).

According to Schaffer, the actual coding with the MICLASS system is accomplished by means of an interactive, "conversational" computer program, in which the computer interrogates the user with a series of questions in simple English. The number of questions asked varies according to the complexity of the part being coded. For a simple part, a minimum of seven questions are involved; 10-20 are required for the average complexity. The computer program automatically generates a code number based on the answers supplied by the user. Several interrelated coding programs are available to allow flexible and efficient organization of the coding task.

Once the part has been classified and coded, the MIPLAN portion of the program is used to develop the process plan. The process planner has a choice of four options for the creation of a process plan. See Figure 4.

1) A plan can be created from scratch, from standard process-description texts that have been prepared and stored in the computer files. The text files are generated by the user, who is free to define how the files are arranged. For example, some companies may want to access standard text via operation codes (somewhat like the OPCODES of CAM-I's CAPP) while other prefer to use



Fiqure 4. MIPLAN System Schematic

machine-tool-identification numbers.

Regardless of the organization, a menu of standard text material is associated with a machine, a work station, or a process-identification code, and these menus can be consistently updated and edited as planners work with the system. These texts can then be assembled and edited for each step in the process plan.

2) An incomplete process plan can be retrieved from the computer and finished. This option is not only handy for necessary interruptions but is useful when some information turns out to be missing or unavailable after planning has been started. The incomplete plan need not be discarded; it can be retrieved and the information added when it becomes available or is convenient.

3) A process plan can be retrieved by entering an existing part number. If the new part to be processed is different from the existing part, the retrieved process plan can be edited and a new plan created without the original plan being destroyed. In other words, existing plans can be modified to create new plans for similar parts, a step toward family-of-parts planning.

4) A process plan can be retrieved through the group-technology code number for the same or a similar part. To do this, the planner can enter a complete code number or a

partial code number. If there is only a partial code number or no available code number for the part, the planner can generate one by invoking the MICLASS interactive classification and coding program.

After the process plan has been satisfactorily edited, the plan can be stored, printed or purged. One of the major advantages of any CAPP system is that the computer is also available for the many related calculations that must be made. Schaffer (1981) listed the following options and sub-programs of the MIPLAN program:

MICHECK, which checks the data files for unusual values and identifies abnormal circumstances, such as huge lot size or long setup times.

MIDVL, which checks the data files for duplicate code number to identify different structures.

MIMIX, which shows the product mix by graphing the frequencies with which a specific part attribute or any specific machine-tool routing occurs in the data file. Additionally, it calculates the percentage of total population or loading.

MICLUS, which analyzes production flow and simplifies routing by using a similarity-coefficient calculation. It can assign machine tools to various groups (cells) according to their frequency and sequence of use in part production.

MIPROM retrieval routines, which develop matricies of the code numbers of those parts produced by specific machine-tool code.

MIFAMT, which identifies the additional machine-tool requirements and secondary operations needed to produce the parts assigned to a work cell.

MIMSP, which divides the analysis data file into one or more files containing all of the parts not selected by these matrixes.

MILOAD, which calculates for each machine-tool-tool code the manufacturing loads as determined by the production requirements of the parts. This information is then compared with available machine-tool-capacities. Machine overloads are "flagged" and possible alternatives are displayed.

MICELD, which produces a matrix of several MILOAD outputs showing the possible loading of the same machine tool in several work cells.

MIFLOW, which is used to effect high-volume changes or deletions to machine-tool codes in the data file.

MICOST, which calculates manufacturing costs of work pieces.

MISEP, a conversational retrieval program that searches for drawings based on an entered code number, drawing number, or name.

MIAPP, a conversational program that searches for process plans based on entered code number or drawing number.

MIGRAPHICS, which permits design and manufacturing information retrieval on a computer graphics terminal.

APPENDIX B: ICAPP

Interactive computer-aided process planning system for prismatic parts (ICAPP) is considered a constructive CAPP program in this paper. It was developed at the University of Manchester Institute of Science and Technology (UMIST) in the United Kingdom by M.E. Ssemakula and B. J. Davies. They describe the system as follows. (Ssmakula, M.E. & Davies, B.J., 1984)

This interactive or constructive planner is feature-oriented. The information describing the component to be produced is entered into the system by describing individual features on the component. This is done on an interactive basis with the computer asking for information regarding each feature which is generally readily available on the part drawing. The system then uses this given information in determining the details of how each feature is to be produced. The system can handle eight different geometric features which are commonly associated with prismatic parts. They are based on eight machining operations:

1) Face Milling
2) Peripheral milling
3) Drilling
4) Boring
5) Reaming
6) Tapping
7) Counterboring
8) Countersinking

The necessary machining operations to produce each feature are determined by the system from the feature type and its dimensions, taking accuracy and tolerances into account. Suitable tools are selected from an established tool file.

For a selected range of materials, equations have been established from which cutting conditions for each machining process can be calculated. These calculated conditions are then displayed on the screen and the user has the option to alter any of the values It has been found that this ability to override calculated conditions makes the system more readily acceptable to potential users as they feel that their expertise can still be incorporated in the resulting process plans. The planning logic used in ICAPP is a combination of variant planning via the part family concept, and the generative planning concept.

A major extension to the capabilities of the ICAPP system has been the incorporation of an option whereby COMPACT II programs (COMPACT II is a registered trademark of MDSI) can be generated. These programs can be directly run on NC machines or machining centers. This is one step towards the integration or linking of the ICAPP process planning with the wider Computer Aided Manufacturing field.

APPENDIX C: GECAPP

General Electric Computer Aided Process Planning (GECAPP) was developed at the General Electric Industrial Electronics Development Laboratory (IEDL) in Charlottesville, Virginia. It was planned and written to minimize software customization. GECAPP was developed on a VAX 11/780 as a stand-alone system and a prototype has been integrated with CALMA's VAX DDM CAD/CAM system. GECAPP provides capabilities for external system interfaces and uses graphics to provide complete process plan detail. It will perform in the manual, variant or generative process planning modes (Gongaware, T., et al, 1984).

GECAPP was designed for the process planning of printed circuit boards. This system uses a group technology coding system to describe the part to the system. It uses an interactive system based on a classification tree scheme to code each board. The coding software in GECAPP traverses the classification tree by presenting a menu of part features to select from at each branching in the tree. See Figure 5. This provides a user friendly data entry.

Once the part has been coded, GECAPP is ready to generate a process plan for it. GECAPP's generative plan

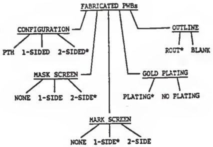


Figure 5

PRINTED WIRE BOARD FABRICATION CLASSIFICATION TREE

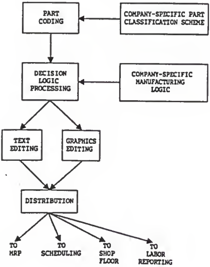

PART CODING
COMPANY-SPECIFIC PART CLASSIFICATION SCHEME
DECISION LOGIC PROCESSING
COMPANY-SPECIFIC MANUFACTURING LOGIC
TEXT EDITING
GRAPHICS EDITING
DISTRIBUTION
TO MRP
TO SCHEDULING
TO SHOP FLOOR
TO LABOR REPORTING

Figure 6

GECAPP PROCESS PLANNING USE

uses "if-then" logic rules to create process plans by comparing the part code against the "if' part of each rule in the manufacturing logic data base. If the part characteristics as described by the part code fall within the range of part features described in the "if" part of a rule, the "then" part of the rule is put into the part's process plan. See Figure 6.

When all of the rules in the data base have been checked, the invoked rules are sequenced by operation number to complete the process plan. At this point, the planner has the option to review and/or modify the generated plan.

GECAPP uses graphics as part of the process plan to display setup instructions, assembly details, text notes, tooling details, fixturing instructions, etc. for operator use on the shop floor. These graphics were developed on different CAD systems, and each can be processed for use by GECAPP.

The manufacturing logic data base for generative process planning is maintained by software utilities provided with the system. GECAPP also provides standard software for the user to create a custom data dictionary for each GECAPP system application. This data definition language allows coordination between management and the user. Additionally, it allows for fast and easy system installation. Editing capabilities are also provided to analyze the rule data base to eliminate contradictions, redundancies, and exclusions.

General Electric primarily used the process plan as an assembly aid on the factory floor. The plan for a control panel assembly references graphical instructions outlining which options have been selected for a part in a customer order. In addition, assembly programs for such items as the automatic insertion of components can be generated. This automatic assembly information is then stored in the process plan for future reference.

APPENDIX D: GENPLAN

Genplan, as the name would seem to indicate, is a generative type CAPP program for turning operations on cylindrical parts. It was developed by Lockheed-Georgia to capture the large amount of knowledge and experience from an aging (and retiring) process planning group. (Tulkoff 1981) Lockheed-Georgia was a member of CAM-I and had first implemented CAM-I's variant CAPP program in 1976 on an IBM 370-168 main frame computer . This initial start into CAPP helped toward the development of the database necessary for a generative type of program.

In the late 1970's, Lockheed developed it's own coding and classification program (one does not exist in CAM-I's CAPP). It characterizes engineering drawings based on geometry, size, and manufacturing processes. Additionally, capacities and capabilities of shop equipment are inventoried and added to the database. A technological manufacturing database which includes process decision logic, machine data, factory rules, tooling data, and labor formulas was also developed. With this information, there is enough data to support a generative process planning system.

GENPLAN can produce a complete process plan without relying on a standard process plan for a similar part

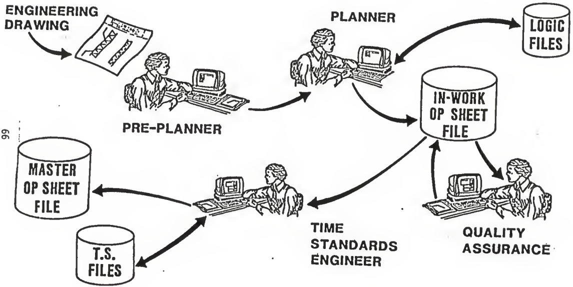


Figure 7. GENPLAN Flow (Tulkoff, 1987)

based upon the decision logic intrinsic to the system. The process planner assigns the special code based on the part description. The GENPLAN software then quickly analyzes the data, evaluates alternatives, and makes the basic planning decision (Schaffer 1981). The process plan so generated requires only minor fill-ins by the planner.

The result: process plans that are consistent not only in methodology but also in sequence, format, and terminology and incorporate the latest technology. This is all done without reliance on retrieval of standard plans. "The system capture both the art and science of manufacturing," says Tulkoff.

APPENDIX E: LOCAM

LOCAM is a Computer Aided Process Planner available through Prime Computer Inc. (Natick, MA). It is considered a generative/expert system that is supplied with multiple utilities and routines which enable the user to define and update the logic rules, manufacturing data base, and Process Plans. It is a generative CAPP program that uses an expert system to interface between various functions and capabilities.

The package consists of several sub-systems that comprise a major portion of a complete Computer Integrated Manufacturing system. (Logan, F. A., 1984) They include: Process Planning, Planning Management & Administration, Work Analysis, Time Standards, and Coding/Classification.

The Process Planning sub-systems enable process plans and associated documentation to be generated automatically from a planner's responses to basic questions or his use of key words that define the logic of manufacture. The system consists of modules for:

1) Comprehensive creation and maintenance of an engineering database. This includes times and descriptive information to any level of detail required by a company. These can be taken from time studies, estimates, MTM data, standard allowances, standard instructions and other

related data.

2) Developing manufacturing logic based on a company's current standards and practices, resources and production /industrial engineering expertise.

3) Production user-defined documents and information files such as process layouts, routing files and shop instructions.

The LOCAM routines were developed from practical experience. They simulate the information and decisions which would traditionally be carried out manually. These decision structures can be easily modified by the user through an interactive decision logic module.

The LOCAM system was developed for transportability between a variety of industries including electrical/ mechanical assembly, press work, fabrication and machining through to the preparation of product specifications and sales estimates. Industry standard databases can be provided for some fabrication and machining processes.

The generative nature of LOCAM allows for reevaluation and regeneration of existing process plans when new technology, processes, or equipment is added to the manufacturing plant and subsequently to the LOCAM database. Additionally, the system provides for documentation of these and other changes. Operation layouts, route sheets, tool lists, NC tapes, customer quotations, etc. can stay with the part. For example, this feature is especially helpful in the electronics manufacture for tracking engineering changes, updates, and failure rates of complex electronic circuit boards.

The Planning Management and Administration sub-system provides a link between the manufacturing engineers/shop floor and the front office personnel. Work-in-Process routines can be used to determine the exact state of a manufacturing order. This also allows sales personnel to better estimate delivery times to customers. Optimally, this system is linked to a Manufacturing Requirements Planning (MRP II) program. This link is easily provided with the "expert system" provided in the LOCAM package.

The LOCAM system has a self-contained Group Technology/Coding Classification (GT/CC) system. This system allows the user to automatically generate user defined classification codes for any component, assembly or sub-assembly. In addition, component information can be retrieved at selectable levels of classification. The classification and coding system was designed for extensive use of a high level generative process planning system.

APPENDIX F: CMPP

A highly sophisticated generative process planning system for cylindrical parts is the Computer Managed Process Planning (CMPP) system (Waldman, 1983). This system was developed by United Technologies Corporation in conjunction with the U.S. Army Missile Command at a cost of 3.5 million. It is considered a break through in the marriage between a CAD/CAM and a CAPP system. It is capable of accepting geometric part data from a CAD system and can perform planning functions to generate manufacturing documentation, drawings, or N/C programming data.

CMPP is a data base driven program made up of over 1000 routines. It is written in Fortran 77 and is compatible with compilers such as IBM's Fortran H (extended and enhanced), Univac's ASCII, and those of Digital Equipment and Control Data (Waldman, 1983). The program and documentation is available to the private sector through the U.S. Army.

Figure 8 shows a general overview of the CMPP system. The CMPP system has three primary subsystems data base files:

Part Design File - this subsystem includes software

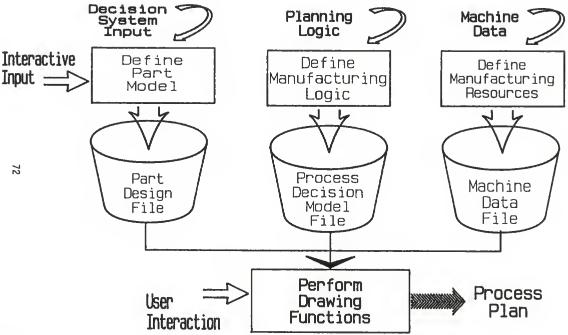


Figure 8. CMPP System Overview

to build and maintain the part design data required for execution of the planning functions. Input is required for both the raw material and the finished part, and includes both geometric and non-geometric data.

Process Decision Model File - this subsystem defines manufacturing logic and includes software to define and maintain "process decision models:" that contain the manufacturer's rules for processing parts. This logic is executed during the planning session to produce a process plan.

Machine Data Files - This subsystem defines manufacturing resources and is used to build data base files containing information on machine classes, machine tools, and the cut parameters (stock removals, tolerances, etc.) appropriate to specific materials for each machine class.

An English-like Computer Process Planning Language (COPPL) is used by the process planner to write process decision models. The language is oriented to process planning terminology providing a "readable" English-like description of the manufacturer's logic for processing a family of parts. The nature of the language enables process planning departments to develop, revise, and evaluate manufacturing rules without relying on programming support from system personnel.

Once a model is written, is is compiled into a sequence of computer-like instructions that are stored in the Process Decision Model File. These instructions are interpreted and executed by the model executor during a CMPP planning session.

A key feature of the COPPL language is its expandable vocabulary. The language structure allows the use of many user-supplied vocabulary terms. When a model is compiled, these vocabulary terms are converted to subroutine calls that will be performed during model execution. Most terms have a corresponding routine in the planning system that performs the required activities. These may involve simple queries of the part model or more complicated calculations involving searches through the data on several part surfaces or features. Other terms, referring to blueprint notes, do not require special routines. A set of frequently used terms/routines is included with the base CMPP system.

To operate the CMPP program, the part description including: general part data, cylindrical features, noncylindrical features, surface finish, surface dimensions, raw material etc. are fed to the Part Design File either directly from a CAD/CAM system or interactively on menu driven screens. Using the user-defined manufacturing logic and part design data, it makes its own decisions (generates) in producing a process plan.

Any portion of the plan may be modified and the program reexecuted from that point. The system provides for the detailed modeling of a finished part and its raw material. The CMPP system also includes an automated tolerance charting procedure to determine and analyze the dimensions, tolerances, and stock removals on all cuts in each operation. Blue print dimensions and tolerances can thus be achieved by the generated process plan. The base implementation of CMPP produces all printed output on a standard line printer, and all sketches and associated lettering on a Calcomp plotter.

This program is quite powerful. However, it is limited to machined cylindrical parts.

COMPUTER AIDED PROCESS PLANNING: THE IMPLEMENTATION AND EVOLUTION OF CAPP SYSTEMS

by

Ronald W. Wilms

B.S., Mechanical Engineering,
Kansas State University, 1981

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTERS OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

ABSTRACT

The implementation of a Computer Aided Process Planning (CAPP) system is a complex task. A case study is presented in this paper envolving the actual installation problems of CAM-I's CAPP program at Kansas State University. The specific hardware and software issues are addressed in detail. Additionally, the topics of: group technology, classification coding, management considerations, short/long term planning, and time/money considerations of implementation are discussed.

The evolution of CAPP software continues to amplify the hardware/software issue. Many systems are available; each with a varing level of sophistication. An overview of the following systems is also included: MIPLAN, ICAPP, GECAPP, GENPLAN, CMPP and LOCAM systems.