AN ABSTRACT OF THE THESIS OF

Yu-An Li for the degree of Master of Science in Industrial Engineering presented on August 24, 1994 .

Title: Integrating Product Design and Manufacturing Process: A Framework and Implementation

Abstract approved: Redacted for Privacy

Sheikh Burhanuddin

The importance of integrating design and manufacturing becomes apparent when the increase in the degree of difficulty of change is observed as the product development proceeds from concept to production in a serial engineering process. The greatest opportunity in design for manufacture occurs at the initial design stage before any commitments to tooling and equipment have been made.

This research develops a framework and an implementation system dealing with integration of design, manufacturing and economic aspects in the development of a product. The objective is to evaluate process technology for a specified product design and to identify the best work/tool material combination and production conditions to optimize the production process. A commercial CAD/CAM package (SmartCAM) playing roles as a part design tool, a processing time simulator, and a NC code generator is integrated with a manufacturing database, and a machining cost model. This integrated system runs in Microsoft Windows environment under an external program which not only coordinates the activities of various modules but also enhances the capabilities of SmartCAM. This system allows product design evaluation for economic and technical criteria and recommends best manufacturing environment. An NC program containing recommended machining parameters is generated. Furthermore, the system reports on tool wear on each tool per part manufactured. This information is useful for cost analysis as well as for producing a tool replacement schedule.

Integrating Product Design and Manufacturing Process: A Framework and Implementation

by

Yu-An Li

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed August 24, 1994
Commencement June 1995

Approved:

Redacted for Privacy

Professor of Industrial and Manufacturing Engineering in charge of major

Redacted for Privacy

Head of Department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented August 24, 1994

Typed by Yu-An Li for Yu-An Li

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF SYMBOLS

C = Taylor's constant

C_t = Initial tool cost, $

C_D = Direct labor cost, $/min

C_s = Set-up cost per lot

d = Tool or work diameter

D = Depth of cut, inch

D_j = The jth candidate depth of cut

D_{max} = The maximum depth of cut allowable

D_{min} = The minimum depth of cut allowable

Dr = Recommended depth of cut

E = Motor drive efficiency

f = Feed rate, rpm

F = Feed rate, ipm

F_j = The jth candidate feed rate

F_{max} = The maximum feed rate allowable

F_{min} = The maximum feed rate allowable

Fr = Recommended feed rate

F_t = Feed per tooth

HP = The available machine horse power

HP_r = Required horse power for machining

HP_s = Required spindle horse power

HPu = Unit horse power, in^3/min/hp

i = Tool number

j = Data array number from 0 to 3

k = no. of different tools required for part processing

l = Length of cut, inch

Δl = Cutter approach and over travel, inch

L = Parts per lot

n = Tool life exponent

n_t = no. of teeth

N = Spindle speed, rpm

Q = Metal removal rate, in^3/min

V = Cutting speed, ft/min

V_e = Cutting speed for minimum cost

V_j = The jth candidate cutting speed

V_m = Cutting speed for maximum production rate

V_{max} = The maximum speed allowable

V_{min} = The minimum speed allowable

Vr = Recommended cutting speed

t_i = Cutting time per piece per tool type, min

t_{ci} = Tool changing or reindexing time, min

t_l = Part loading and unloading time, min

T_i = Tool life, min

T_e = Tool life for minimum cost

T_j = The jth candidate tool life

T_m = Tool life for maximum production rate

Tr = Recommended tool life

w = Width of cut, inch

Integrating Product Design and Manufacturing Process: A Framework and Implementation

CHAPTER 1
INTRODUCTION

1.1 Motivation and Problem Statement

The traditional product development process, still widely used in industry, involves disjointed design, manufacturing and evaluation activities. Due to the segregated nature of these activities, engineering change orders (ECOs) are frequently issued. The cost of changes increases as the product progresses from conception to design to production. The cost of an engineering change during final production could be 10,000 times the cost of change during design [1]. In the later stages of a product's life cycle, change costs are even higher because of time delays and larger number of personnel and manufacturing components involved. The design phase determines 80 percent of the cost of the product [1] and deserves much more attention than it is currently receiving. Potential manufacturing and economic problems may be avoided if attention is paid to manufacturing and economic issues during the design process.

The product development process is undergoing substantial changes due to increasing international competition. The key to success is to manufacture products with the least cost, rapidly, while satisfying customer concerns for quality and delivery. Accomplishing these objectives requires complete integration of design and production. Even though a large number of manufacturing organizations could benefit from integration of design and manufacturing, the rate of implementation of this concept has been rather slow.

1.2 Research Objectives

The objective of this research focuses on the design-manufacturing evaluation phase to integrate product design and manufacturing process with a machining database and economic models to identify the best work/tool material combination for minimizing the production costs.

The scope of the system encompasses the following aspects:

- **Computer-Aided Design**: includes work and tool material selection, geometry design/modification.
- **Computer-Aided Manufacturing**: includes optimizing production conditions, estimating production time, and generating NC codes.
- **Machinability Database**: provides the information bases for optimizing production environments for different processes and materials.
- **Economic Analysis**: estimates the time and cost data for producing a part, and reports tool wear for tool replacement scheduling.

The proposed criteria to optimize machining conditions are based on the calculations for minimizing production cost and maximizing production rate, and the database that have been obtained from Machining Data Handbook [6] which relates tool life to cutting conditions. In order to approach optimal machining conditions, the user-friendly integrated system allows user input of a machine's capability information, work/tool material characteristics and process types, and then the system will automatically search in database and calculate for recommended machining parameters. Those parameters will be used for estimating production time and cost in CAM module. The final reports of economic analysis contains detailed tooling time and cost data as well as tool wear information that are very useful for evaluate part design before production.

This system integrates a CAD/CAM software (SmartCAM) and implements under Microsoft Windows with an external program written in Visual Basic 3.0.

1.3 Summary of Remaining Chapters

Chapter 2 provides the general background and a review of current literature. The general background covers an overview of CAD/CAM, database design, traditional manufacturing processes, product design, and discussion of machining economics. The literature review provides the approaches that have been developed to estimate costs or optimize the machining conditions to minimize the costs.

Chapter 3 describes the development of a model to implement the CAD/CAM software and use the database for appropriate machining conditions. The chapter also describes the system integration and economic analysis model.

In Chapter 4, the model is carried out in a prototype system developed in the Visual Basic programming language. A demonstration example is presented along with a discussion of the integrated system.

Chapter 5 provides a summary of the thesis and presents conclusions drawn from this research. This is followed by recommendations for further research based on these conclusions.

CHAPTER 2

GENERAL BACKGROUND AND LITERATURE REVIEW

This chapter discusses a number of issues and reviews publications that are relevant to this thesis. These include:

- The concept of "Integration of Computer-Aided Design and Manufacturing"
- The interrelationships among Product Design-Manufacturing
- Discussion of manufacturing processes
- Multi-Tool machining center and machining economics
- Literature review

2.1 CAD/CAM

The term CAD/CAM is the combination of Computer-Aided Design and Computer-Aided Manufacturing. CAD refers to the use of the digital computer to support the design engineering function, and CAM is concerned with the use of the computer of support manufacturing engineering activities. CAD/CAM is a symbolic term of efforts to integrate the CAD and CAM functions rather than to treat them as two separate and different activities.

2.1.1 Computer-Aided Design

CAD is most commonly associated with the use of an interactive computer graphics system. There are several fundamental reasons for using a computer-aided design system to support the engineering design function [16]:

- **To increase the productivity of the designer.** CAD helps to reduce the time required by the designer to synthesize, analyze, and document the design.
- **To improve the quality of the design**. The use of a CAD system allows the designer to consider comprehensive and complex designs.

- **To improve design documentation.** A better graphical output or documentation can be achieved by the CAD system. It also gives excellent reproduction and modification ability, and reduces drafting errors.
- **To create a manufacturing data base.** Much of the required database (product geometric specification, dimensions of the components, materials specifications, bill of materials, etc.) to manufacture the product is also created.

2.1.2 Computer-Aided Manufacturing

Computer-Aided Manufacturing (CAM) can be defined as the effective use of computer technology in the planning, management, and control of the manufacturing function. CAM applications for manufacturing planning are those in which the computer is used indirectly to support the production function, but there is no direct connection between the computer and the process. The CAM applications included in this research are:

- **Cost estimating:** The computer is programmed to apply the appropriate models (machining process planning, machining simulation, economic model) and input necessary information (labor and overhead rates, and machine and tools data) to calculate the total product cost.
- **Computerized machinability data systems:** Create the computer programs to recommend or help determine the speeds and feeds that should be used to machine a given workpart. The calculations are based on data that have been obtained either in the factory or laboratory which relates tool life to cutting conditions.
- **Computer-Assisted NC part programming:** For complex part geometries, computer-assisted part programming for numerical control machines represents a much more efficient method than manual part programming.

2.1.3 Integration of CAD/CAM and Database

The emergence of Computer-Aided Design and Computer-Aided Manufacturing denotes integration of design and manufacturing activities by means of a computer system, which has had a major influence on manufacturing by standardizing product development and reducing design effort and prototype work. In addition, CAD/CAM results in significantly reduced costs and improved productivity.

In an integrated CAD/CAM system, a direct link established between product design and manufacturing computer-based systems has been developed [16], which creates much of the data and documentation required to plan and manage the manufacturing operations for the product. The manufacturing database includes all of the product design information (geometry dimensions, tool path, etc.), as well as parameters required for manufacturing (process conditions, material properties) and for economic evaluation (machine specifications, direct and indirect costs, etc.). Figure 2.1 shows the relationship of CAD/CAM and the database system.

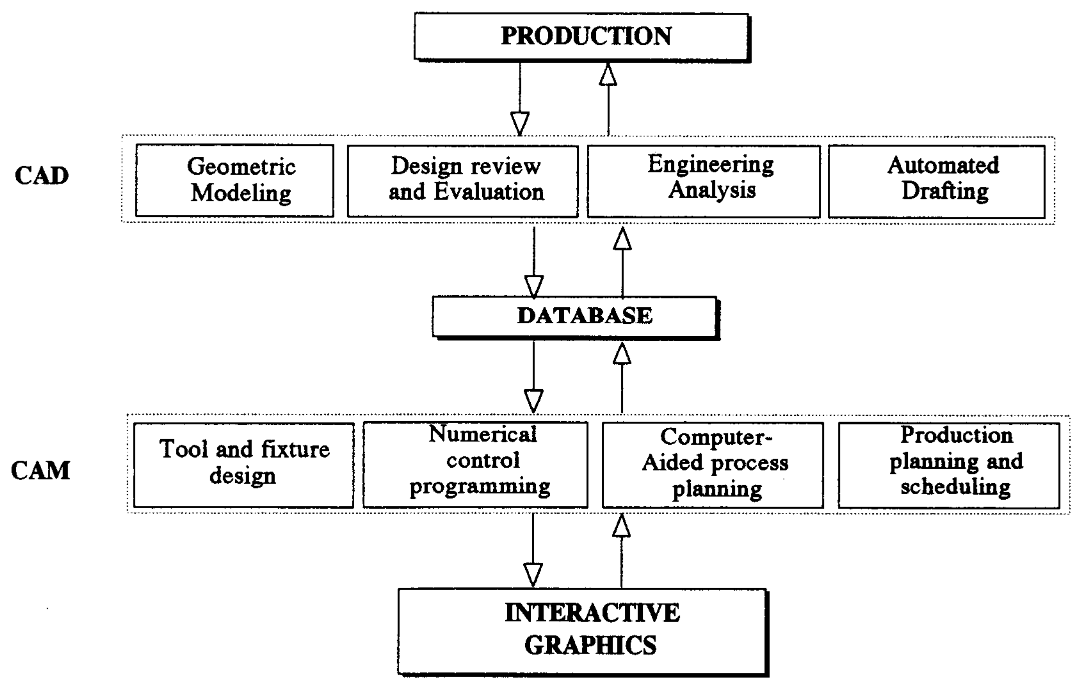

Figure 2.1. Desirable Relationship of CAD/CAM Database to CAD and CAM

2.2 Product Design - Manufacturing

To develop a new product, cost and quality are the most concerned factors. Figure 2.2 shows how a new product is designed and the factors influencing product cost [15].

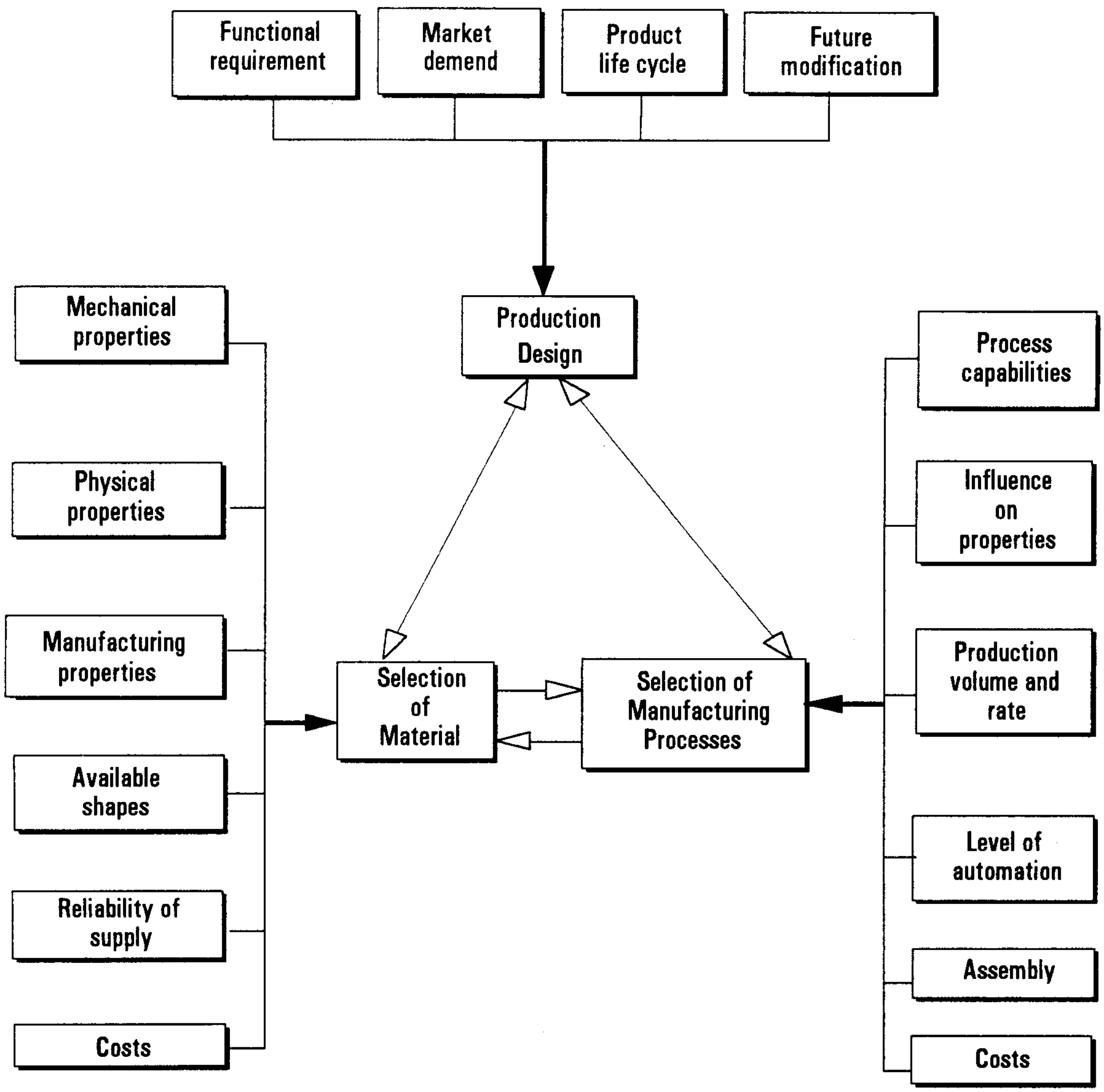

Figure 2.2. Interrelationships among Various Factors Influencing Product Costs

The cost of a product often determines its market ability and customer satisfaction. Each production step must be analyzed with a view toward ensuring an optimal economic outcome. To minimize costs, the following approaches are considered:

- Product design
- Selection of material
- Selection of manufacturing process

2.2.1 Product Design

As Figure 2.2 shows, product design refers to the functional requirement, market demand, product life cycle and future modifications. The functional requirement and market demand are totally driven by customers who demand the product. Product life cycle and modifications are determined by both designer and customers. Depending on the particular customer group and the progress of technologies, there will be differences in the way the product cycle is activated. To minimize the time span between the original concept and marketing a product, simultaneous or concurrent engineering can help reduce the time and make future modifications easy.

2.2.2 Selection of Material

When selecting materials for products, designer first concerns the mechanical properties: strength, toughness, ductility, hardness, elasticity and fatigue, etc.. The mechanical properties specified for a product and its components should be for the conditions under which the product is expected to function. Then the physical properties are also considered: density, specific heat, thermal expansion, conductivity, and electrical and magnetic properties as well as chemical properties such as oxidation, corrosion, toxicity and flammability. Manufacturing properties determine whether the material can be cast, formed, machined, welded, and heat treated with relative ease. Availability and cost of raw and processed materials are major concerns in manufacturing. Reliability of supply also plays a significant role, as well as demand, in affecting cost. Other costs such as requirements of expensive machinery, personnel with special skills, specialized training or education are involved in processing materials. The economic aspects of material selection are as important as the technological considerations of properties and characteristics of materials.

However, as new technology is developed or market pressures arise, it is common for a new material to be substituted in an existing design or manufacturing system. Often, the result is improved quality or reduced cost, but it is possible to cause more harm than good. It is obvious that the responsible engineer should first consider all of the design requirements before authorizing a material substitution. Approaching a design or material modification as thoroughly as a new problem may well prevent costly errors.

Based on the previous considerations, a tentative selection of several candidate materials should be made. Not all materials are suitable for all processes. The ultimate objective of the selection of material is to arrive at an optimal combination of material and process for the manufacturing of the desired product.

2.2.3 Selection of Manufacturing Processes

Many processes are used to produce parts and shapes. Selection of a particular manufacturing process depends not only on the product shape but also on the material and its properties. The categories of processing methods for material are:

- **Casting:** (expendable mold and permanent mold)
- **Forming and Shaping:** (rolling, forging, extrusion, drawing, sheet forming, powder metallurgy, and molding)
- **Machining:** (turning, boring, drilling, milling, planing, shaping, broaching, grinding, chemical, electrical, and electrochemical machining, and high intensity beam machining)

- **Joining:** (welding, brazing, soldering, diffusion bonding, adhesive bonding, and mechanical joining)
- **Finishing operation:** (honing, lapping, polishing, burnishing, deburring, surface treating, coating and plating)

In this research, the scope is limited to machining processes, especially to the cutting metal removal processes.

2.3 Manufacturing Process Variables

As discussed in previous sections, the selection of workpiece material is a major concern in product design phase. After the product material has been decided, then selecting an appropriate process and tools becomes the top issue. This research centers on traditional metal-removal processes, especially on cutting process, such as turning on a lathe, drilling, milling, or thread cutting. In the cutting processes, there are several variables that need to be concerned about.

2.3.1 Tool Material and Its Conditions

Selecting the optimum tool material is a major factor in realizing the full potential of a particular machine tool. There are several guidelines offered by *Machining Data Handbook* [6] as an approach for the logical selection of the best tool materials for machining a specific work material:

a. High speed steel tools are generally used in:

- High volume, low cutting speed operations
- Complex tool forms such as form tools, drills, cutoff tools, etc.
- All sizes of end mills, drills, reamers, taps and gear cutters
- Certain machining operations on problem materials, such as nickel base high temperature alloys
- High positive rake requirements.

b. **Cast alloy cutting tool** materials are selected as an intermediate between high speed steel and carbide tool materials. The high cobalt high speed steels also serve as intermediates, and there is a trend for them to substitute the cast alloy tools.

c. **Carbide tools** are generally applicable when one or more of the following conditions exist:

- Rigidity of the machine tool, tooling and workpiece is acceptable.
- Machine tool power is adequate for higher metal removal rates.
- Workpiece configuration and machining operation permit higher cutting speeds.
- High production rates are required.

d. **Ceramic tools, high strength carbides, diamond tools and the cast alloy tools** referred to previously have rather specific application in contrast with the wide usage of high speed steel and carbide tools.

2.3.2 Tool Shape, Surface Finish, and Sharpness

Various tool geometries are used for various work materials and result in different chip formations. The tool surface finish influences not only the dimensional accuracy of machined parts but also their properties. A dull tool has a large tip radius along its edges that will rub over the work surface and generate heat and surface residual stresses which may cause surface damage.

2.3.3 Workpiece Material, Condition, and Temperature

These variables can be adjusted in order to get an easy way for processing. For example, changing the workpiece heat treatment temperature can also change its hardness, flexibility and other material properties as well.

2.3.4 Cutting Conditions and Tool Life

These are the most concerned variables in this research. Cutting conditions, speed, feed and depth of cut, affect the efficiency of a metal-cutting operation, and the cost. Tool life is a variable dependent on cutting conditions. Although all cutting conditions have influences on tool life, cutting speed has been found to have the most significant effect. As cutting speed increases, tool life is rapidly reduced. On the other hand, if cutting speed is low, the tool life is long but the material removal rate is low, and in turn, the process time is long.

To minimize the manufacturing cost, we have to select the optimum cutting conditions to approach the shortest process time and longest tool life under reasonable circumstances.

2.3.5 Force and Energy in the Cutting Process

Force and energy are dependent variables that can only be analyzed but not changed. Analysis on forces is necessary for the proper design of machine tools for cutting operations, and energy or power required has to be determined so that a motor of suitable capacity can be used or not to be exceeded.

2.3.6 Other Variables

Some other important variables are:

- Use of a cutting fluid
- The characteristics of the machine tool, such as its stiffness and damping
- Type of chip produced
- Temperature rise in the workpiece, the chip, and the tool
- The surface finish produced on the workpiece after machining

2.4 Multi-Tool Machining Center

A machining center is a computer-controlled machine tool with automatic tool-changing capability. It is designed to perform a variety of cutting operations on different surfaces of the workpiece. A machining center can be characterized by the following features:

- A machining center is a single numerically-controlled machine with capability of handling a variety of part sizes and shapes efficiently and economically.
- A machining center contains multiple tools. Each tool is used for different machining operations or cuts on a product or on different products.
- A scheduled replacement policy is used for tool replacement or resharpening. After a fixed interval of time, tools are replaced. Premature failure replacements are made as required within the interval.
- The time required for loading and unloading workpiece, changing tools, and machine set-up is considered in calculating the minimum production costs.

Machining centers require significant capital expenditures. The selection of a machining center depends on the type of product, type of machining operations to be performed, and type and number of tools required. The production rate and the dimensional accuracy required are also considered.

2.5 Machining Economics

In machining a part, the total machining cost per piece consists of nonproductive cost, machining cost, tool-change cost, and cost of the cutting tool. The cutting speed is the most critical factor that influences these costs. Figure 2.3A shows the relationships between these costs and the cutting speed. Note that as cutting speed increases, the machine time (hence cost per piece) decreases. However, it is known that tool life decreases with cutting speed. Hence the tool cost increases, as does the tool-changing cost, since tools have to be changed more frequently. Adding these costs to obtain the total cost per piece, as shown in Figure 2.3B., can get one optimum cutting speed for minimum **cost per piece** and another optimum speed for minimum **time per piece**. The range between the two optimum speeds is called **High-efficiency machining range** [4].

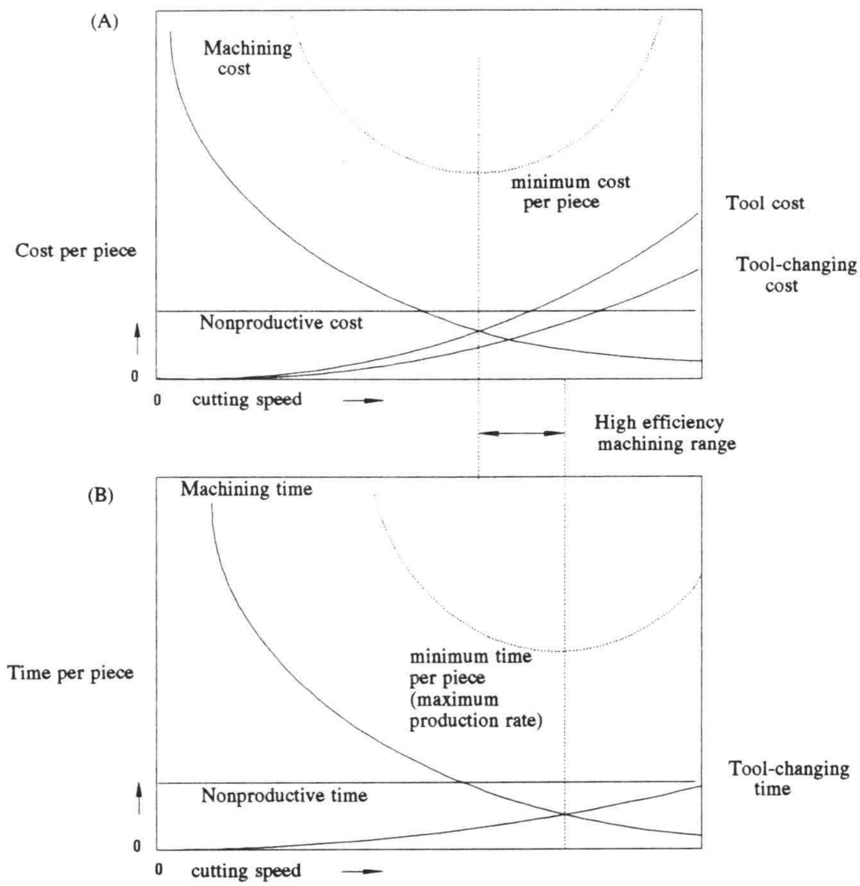

Figure 2.3 A) Cost per piece and B) Time per piece in machining. The range between the two is called the High-Efficiency Machining Range [4].

This analysis indicates the importance of identifying all relevant parameters in a machining operation, determining various cost factors, obtaining relevant tool life for the particular operation, and properly measuring the various time intervals involved in the overall operation.

2.6 Literature Review

There are many approaches that have been developed to estimate costs or optimize the machining conditions to minimize the costs.

Dewhurst and Boothroyd [21] developed a method for product design for efficient manufacture (DFM) which consists of two steps. The first step is the identification of the appropriate materials and manufacturing processes for the component parts of a new product design. The second step is the detailed design of the individual components consistent with the capabilities and limitations of the material-process combinations.

Artificial intelligence and knowledge-based approaches are also greatly applied to optimize the manufacturing process. Billatos and Tseng [26] developed a knowledge-based machining system that uses experimental machining database on the adaptive control with optimization (ACO) concept. ACO has attributes of both feedback system and optimal control system which evaluates the performance within the environment and makes the necessary on-line changes in its control characteristics to improve or optimize its performance. In their knowledge-based system, a linear recursive adaptive tool failure identifier model and a cutting condition constraint model using experimentally measured cutting forces are developed. They also built an optimization strategy and developed an intelligent controller to provide on-line direct measurement of tool failure.

Subramanyam and Lu [25] developed a computer-aided simultaneous system for components manufactured in small and medium lot-sizes. Simultaneous engineering is a widely integrated system to reduce product development time and cost, and to achieve high quality by considering the whole spectrum of product life cycle factors. With this concept, a computer-based design environment was created to cooperatively assist product designers.

Levin and Dutta [24] extended the domain of computer-aided process planning (CAPP) to include a class of parallel machine tools. Multiple tools can simultaneously machine a single workpiece and multiple workpieces can be machined simultaneously. In a conventional sequential machining environment, workpieces travel from machine to machine until all required operations have been completed.

Several research workers have attempted to calculate the optimum cutting conditions by using graphical technique, geometrical programming, or linear programming. Hinduja et al [27] created a procedure to calculate the optimal cutting conditions for multiple pass turning operations with minimum cost or maximum production rate as the objective functions. For a given tool/workpiece combination, the search for the optimum is confined to a feed versus depth of cut plane which is defined by the chip-breaking constraint. By geometric programming, Jha [23] developed a multiple objective function based on cost of production and rate of production in milling operations.

After a review of the above mentioned publications, it appears that most of the estimations of machining time are based on geometric calculations. Moreover, very few of them separated the machining time into contact and non-contact time. The non-contact time should be excluded from the calculation of tool life. In this research, a machining process simulation model will be applied to precisely estimate machining time, and it will include and calculate separately tool reindexing time, tool rapid traverse (noncontact) time, cutting (contact) time and tool dwell time. In order to select the appropriate or, if possible, optimal cutting conditions, a model of selecting machining parameters with a database is also developed.

CHAPTER 3
METHODOLOGY

This chapter presents the objectives of this research, the methodology used, a description of the foundation for the work and the integrated system. Partial of this research is based on the work by Hsueh-Jen Chen [29].

3.1 Design-Manufacturing Evaluation Framework

The complete design-manufacturing process is complex (Figure 3.1), involving numerous considerations such as market analysis, concept design, material and process selection, optimization, information and process control, costing, and manufacturing. The process starts with a problem definition phase that defines the need requirements for a product. These requirements are then translated into product specifications. The specifications define the characteristics of the product that satisfy its functional requirements, and the conditions and constraints to be met during the transformation of suitable resources to the physical product.

In the evaluation phase, the design-manufacturing combination is simulated and evaluated against economic and technical criteria. Sensitivity analysis on design and production parameters helps to identify the best production environment; in some instances it may indicate reconfiguration of product requirements and specifications. The design and product parameters identified at this stage are used in the physical manufacturing process.

3.2 Application Domain

The system described in this thesis focuses on the design-manufacturing evaluation phase, integrating the analysis procedure with appropriate information bases, and generation of an NC program to produce the part under optimal manufacturing conditions.

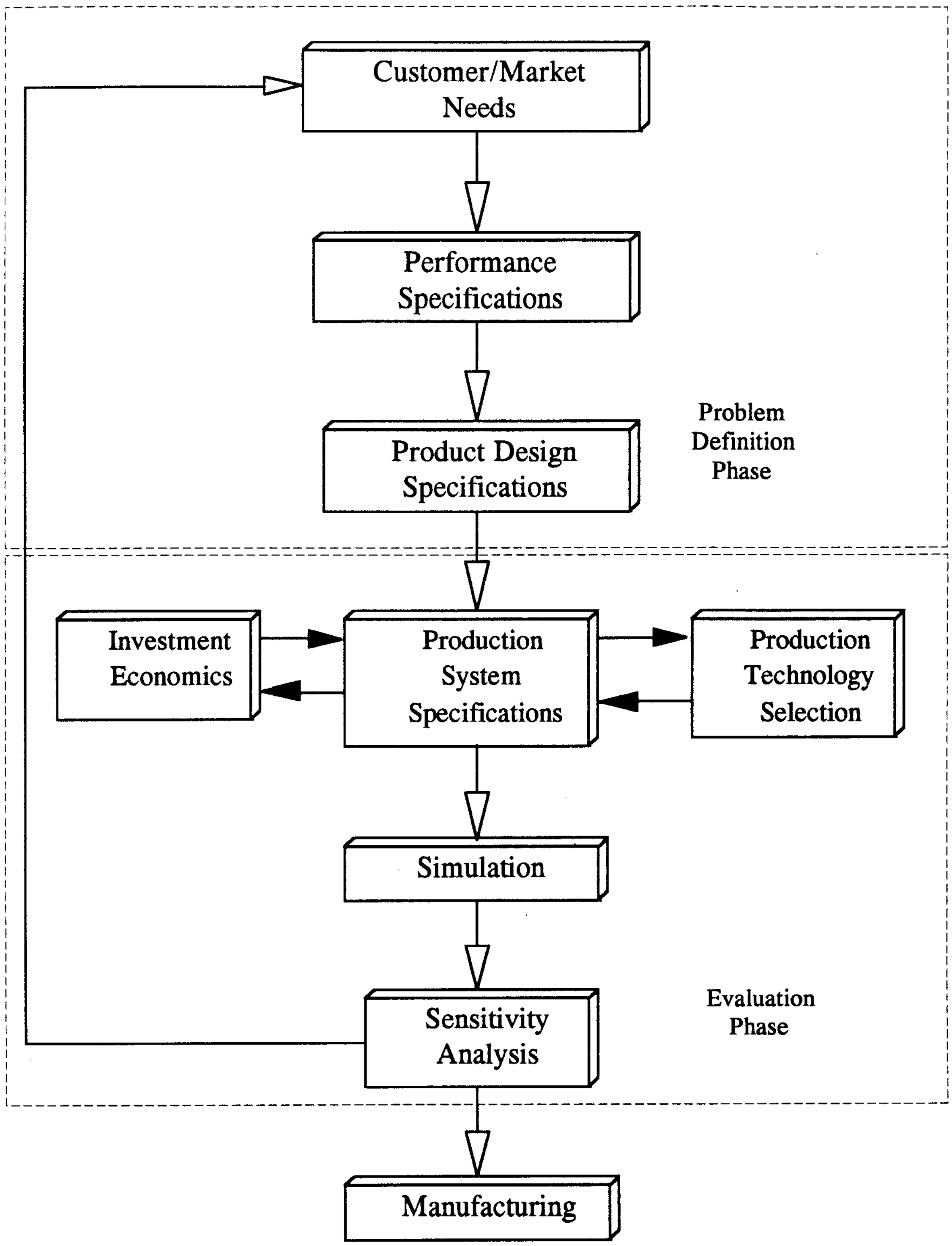

Figure 3.1. Product Design-Manufacturing Process

Figure 3.2 shows various functional modules and the structure of the system. The evaluation process starts with the selection of work and tool materials for the part shape to be produced. Next, the characteristics of the manufacturing processes for creating the part shape are selected. The CAD model of the part is simulated to generate machining data for cost analysis. Finally, a sensitivity analysis is performed to select optimal manufacturing environment for part processing. The system is designed for machining processes, the primary focus of flexible manufacturing. The processes currently included in the system are turning, cut-off, milling, drilling, reaming and boring. Development work on this system has included:

- Design of a user interface to provide front-end (data input) and back-end (output analysis) system interaction with the user. The focus has been to develop user-friendly screens for user data input and editing.

- CAD modeling and simulation of the manufacturing process for analyzing the system for production economics and exploring tradeoffs through sensitivity analysis.

- Creation of material and process databases that characterize information on tool and work materials, and different processes used in machining. The materials include low, medium and high carbon steels, aluminum and cast iron. The tool materials include high speed steel, and coated and uncoated sintered carbide. Depth of cut, feed rate and cutting speed are the three machining parameters included in the database. For each work-tool material combination, several sets of machining parameter values are stored in the database. The database also contains data on tool life parameters.

- Development of a production economics module to determine unit processing cost using data provided by the user and the information calculated by the manufacturing process simulator, such as machining times and tool life.

- Control structure for integrating various modules in the system (the databases, CAD modeling, production economics, and the user interface modules).

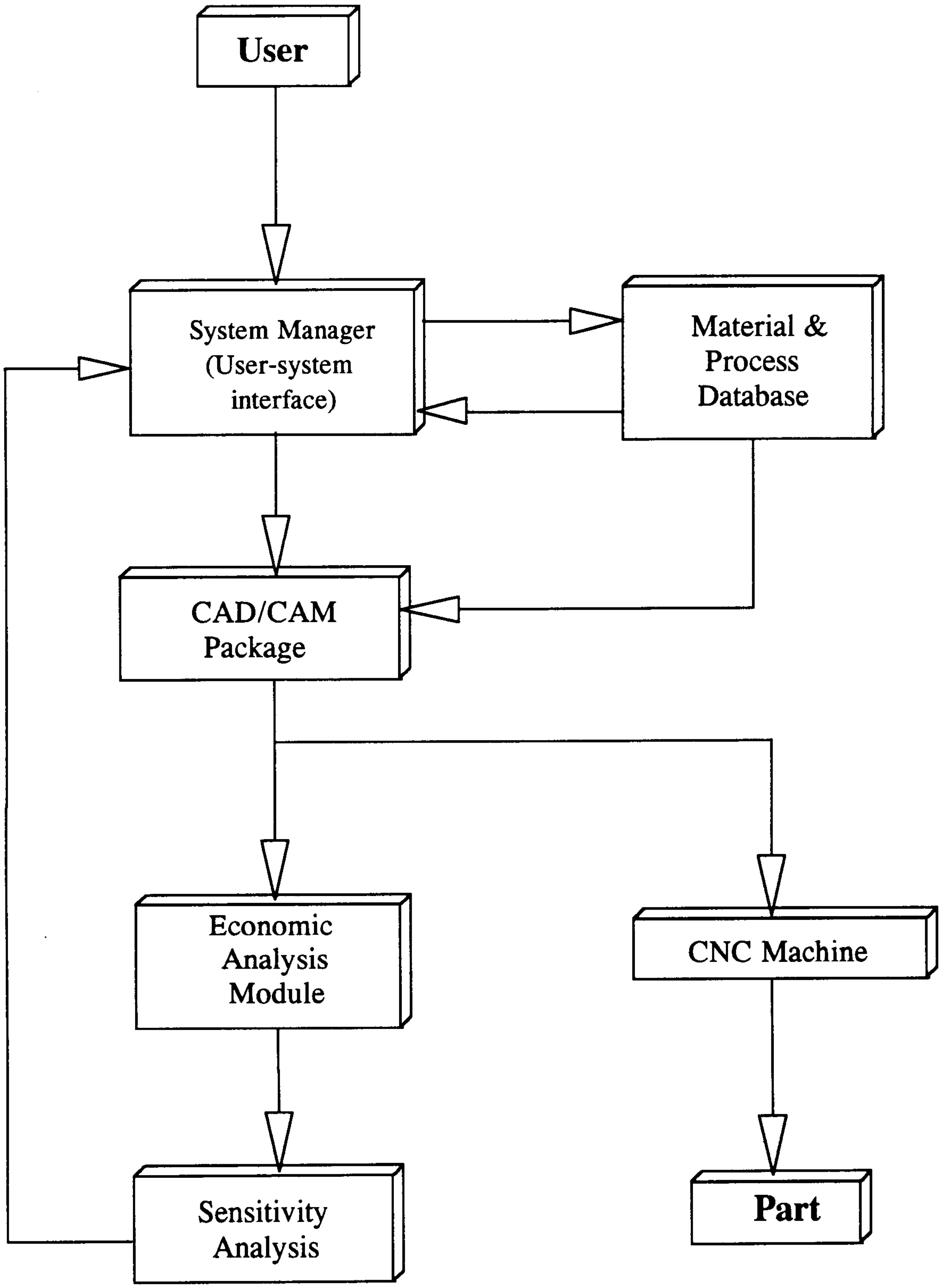

Figure 3.2. A Framework of Design-Manufacturing Integration

3.3 System Description

This section provides detailed descriptions of modules for system development which include:

- Database Design for Machining Conditions
- Criterion for Selection of Machining Conditions
- Simulation Model
- Economics Model
- System Integration
- Sensitivity Analysis

3.3.1 Database Design for Machining Conditions

Database plays a key role in this system. The criterion of machining parameters selection is based on the database from the *Machining Data Handbook* [6], in which all of the machining parameters are stored and ready to use.

3.3.1.1 Structure of Database

The machining parameters database is based on the material properties appearing in tablet form in handbooks published by technical societies such as ASM[a], ASME[b], SME[c], and SAE[d]. The properties included in the database are taken from [6] and [7]. Some rearrangements are necessary and have been done in order to manipulate the database. Two groups of data; tool data and process data, are included in the machining conditions database. The required data for each process are different and are listed in Table 3.1. In addition, the machining parameters depend on work and tool material combinations. In order to efficiently access data, the database is built in Quattro Pro under MicroSoft Windows environment. The structure of the database, described in Table 3.2, encompasses the work and tool materials mentioned earlier.

---

[a] American Society for Metals
[b] American Society of Mechanical Engineers
[c] Society of Manufacturing Engineers
[d] Society of Automotive Engineers

Table 3.1 The Required data for Different Processes

Process	**Required parameters**
Turning	Tool parameters: not applicable Process parameters: depth of cut, feed rate, cutting speed
Milling	Tool parameters: diameter of cutter Process parameters: depth of cut, feed rate, cutting speed
Cutoff	Tool parameters: width of cutter Process parameters: feed rate, cutting speed
Drilling	Tool parameters: diameter of cutter Process parameters: feed rate, cutting speed
Reaming	Tool parameters: diameter of cutter Process parameters: feed rate, cutting speed
Boring	Tool parameters: not applicable Process parameters: depth of cut, feed rate, cutting speed

Table 3.2 Structure and Contents of the System Database

Field Name	Description
COMBINATION #	Key attribute of this database for identifying different combinations of work and tool material parameter used in an operation
WORKMATERIAL NAME	The work materials used in this database includes: . Aluminum alloys . Lowcarbon steel . Medium carbon steel . High carbon steel . Gray cast irons
WORKMATERIAL CODE	Standard AISI grading for work material . Aluminum alloys: "alum" represents all aluminum alloys . Lowcarbon steel: *1005 (1006, 1008, 1009, 1010, 1012, 1015, 1017, 1020, 1023, 1025) 1011 (1013, 1016, 1018, 1019, 1021, 1022, 1026, 1029, 1513, 1518, 1522) . Medium carbon: 1030 (1033, 1035, 1037, 1038, 1039, 1040, 1042, 1043, 1044, 1045, 1046, 1049, 1050, 1053, 1055, 1525, 1526, 1527) 1524 (1536, 1541, 1547, 1548, 1551, 1552) . High carbon: (1064, 1065, 1069, 1070, 1074, 1075, 1078, 1080, 1084, 1085, 1086, 1090, 1095, 1561, 1566, 1572) . Gray cast irons: **ASTM A48 Class 20 - 60
LOWER HARDNESS	Lower bound of work material hardness (BHN)
UPPER HARDNESS	Upper bound of work material hardness (BHN)
TOOLMATERIAL NAME	The tool materials used in this database includes: . HSS (High speed steel) . Sintered carbide

Table 3.2 Structure and Contents of the System Database (cont.)

Field Name	**Description**
TOOLMATERIAL CODE	Tool material grade . HSS (High speed steel): M1, M2 (Molybdenum high speed steels for general purpose) M42, T15 (High speed steels containing Cobalt for heavy cuts and extreme abrasion resistant) . Sintered carbide: C2, C3, C5, C6 (Uncoated carbide) CC2, CC6 (Coated carbide)
TOOL TYPE	Tool type used in the operation: . MILL . REAM . CUTOFF . TURN . DRILL . BORE
TOOL DATA	Tool specifications: MILL: diameter of cutter (inch) REAM: diameter of cutter (inch) CUTOFF: width of cutter (inch) TURN: not applicable DRILL: diameter of cutter (inch) BORE: not applicable
D1, D2, D3, D4	Depth of cut (four sets) MILL: radial depth of cut (inch) REAM: assume thin depth of cut applied (1 is used for system checking) CUTOFF: not applicable (1 is used for system checking) TURN: depth of cut (inch) DRILL: not applicable (1 is used for system checking) BORE: depth of cut (inch)

Table 3.2 Structure and Contents of the System Database (cont.)

Field Name	Description
S1, S2, S3, S4	Speed (feet per minute)
F1, F2, F3, F4	Feed rate MILL: (inch per tooth) REAM: (inch per revolution) CUTOFF: (inch per revolution) TURN: (inch per revolution) DRILL: (inch per revolution) BORE: (inch per revolution)
n	Constant *n* for Taylor's tool life equation
C	Constant *C* for Taylor's tool life equation
HPu	Unit horse power which depends on the work material

* Material index number based on [6], Section 26
** American Society for Testing and Materials

3.3.1.2 Establishment of Database

Creating such a large database is time consuming. In order to reduce the different types of softwares involved in the system, Quattro Pro for Windows was chosen to develop the database because of its flexibility and ease of use. The considerations and advantages of using Quattro Pro are:

- Easy to use and set-up data structure.
- Excellent calculation and data manipulation ability.
- Database can be saved in different formats such as Paradox, dBase III, etc. (Figure 3.3)
- Database can be conveniently extended in the future.
- Database file can be handled independently without the need of a database management package.

For example, COMBINATION #, the key attribute of this database, is defined as the composition of the first letter of TOOL TYPE + WORKMATERIAL CODE + UPPER HARDNESS + TOOLMATERIAL CODE + (1000 TOOL DATA). This can be done in Quattro Pro by implementing a cell equation and pasting it to other cells in an efficient and error-free way.

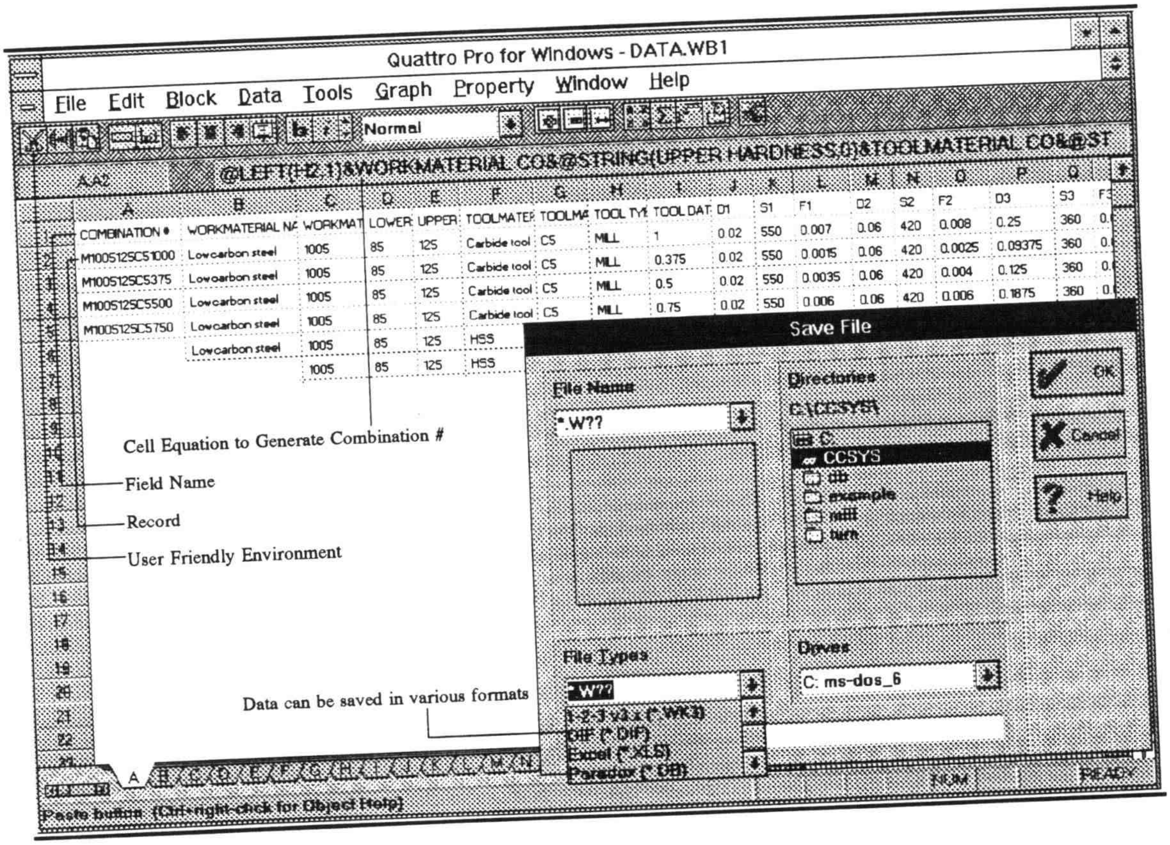

Figure 3.3. Building the Machining Parameters Database in Quattro Pro/w

For the purpose of optimizing cutting conditions, it is necessary to determine the mathematical relationship between tool life and the pertinent cutting parameters such as speed, feed and depth or width of cut. The most common approach is to assume that Taylor's equation relating tool life to cutting speed applies:

$$VT^n = C$$

where V = Cutting speed, T = Tool life and n and C are constants.

The values of n and C are obtained from [4,5,6,7,8] for various machining processes, and specific work and tool material combinations.

3.3.1.3 Data Access

Two types of data are involved in conjunction with CAD/CAM simulation and evaluation modules. The operation information specified by the user and the simulation outputs are both saved and written to a sequential text file, and accessed when needed. To demonstrate how to access those sequential data files in the system, the flow of data transfer is displayed in Figure 3.4. The process set-up file, with an extension of ".PRC", may be accessed in both process set-up and evaluation screens. In the Set-up stage, the SmartCAM template file(.TMP) and recommendation file(.REM) are also constructed. After completing the CAD/CAM simulation, the resulting output, typically without any extension, is sent along with process set-up data to cost evaluation screen which performs economic analysis and generates a summarized report(.EVL).

Accessing machining parameters from database (DATA.DB), a totally different data accessing pattern from a sequential file, needs locating the data address in order to retrieve records. Keyed access method is an efficient way and has been used in this research. The COMBINATION #, the key field pointing to the unique data address, is generated by the composition of other attributes. Once all of the necessary information has been collected and verified, the recommended machining data provided by the database may be retracted to the system. Figure 3.5 shows the user interface developed in Visual Basic to search for machining parameters. The system automatically checks for sufficient inputs before the searching begins.

3.3.2 Criterion for Selection of Machining Parameters

In this phase, the cutting tools and parameters for the machining operation are determined either automatically or manually based on the restrictions of the machine tool's capabilities and the combination of work and tool materials.

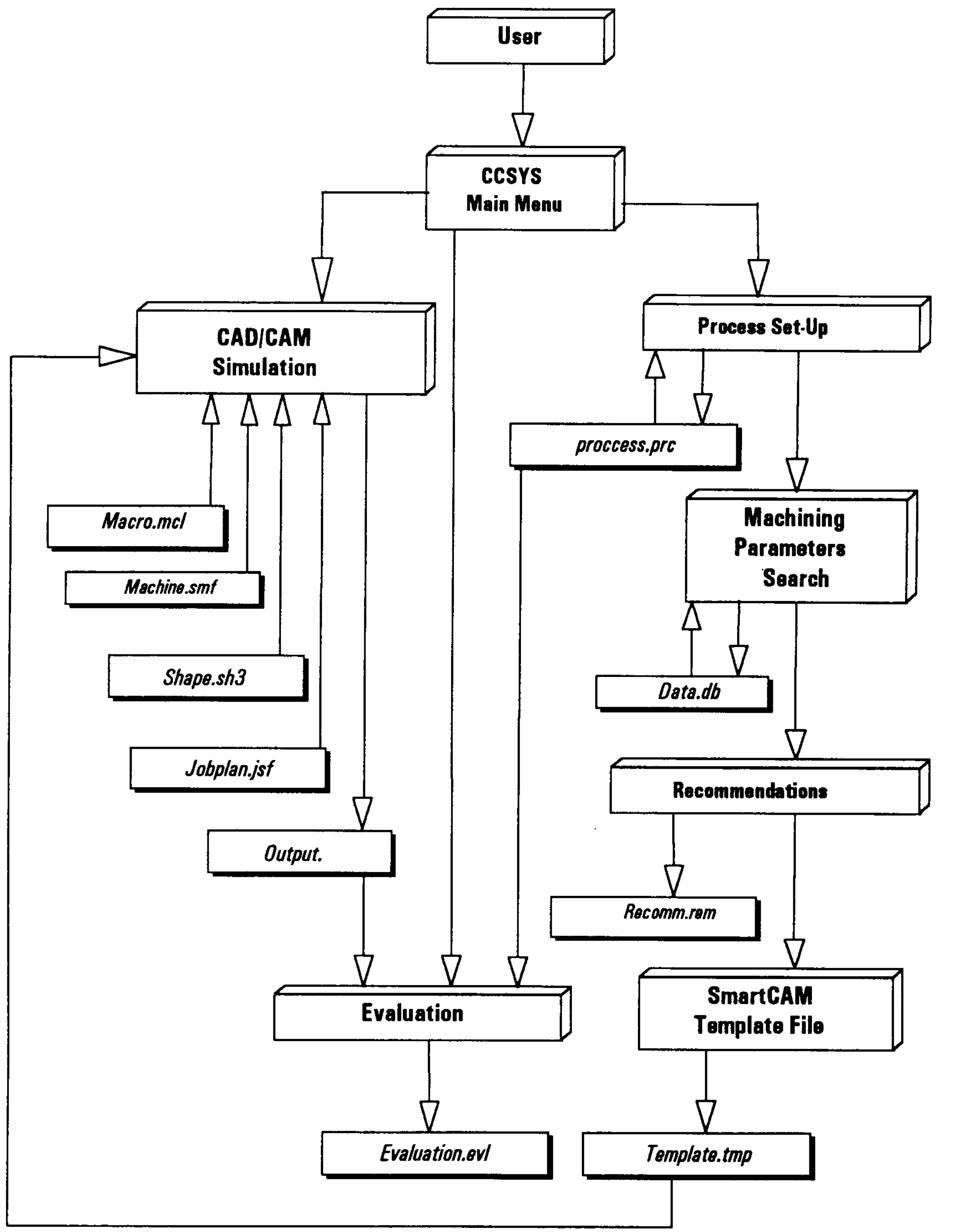

Figure 3.4 Flow of Data Transfer

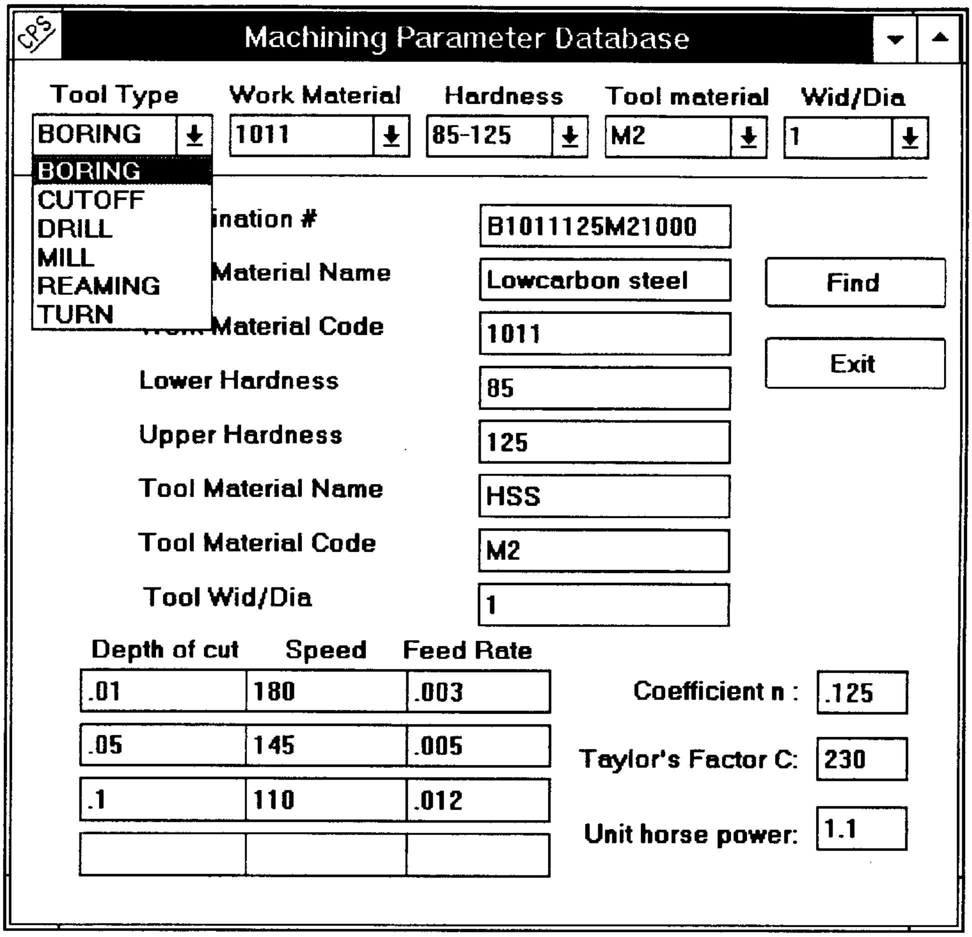

Figure 3.5. Data Extraction from the System Database

3.3.2.1 Selection Constraints

To optimize the process conditions, the factors that impose constraints on the cutting conditions need to be considered. The following constraints are considered in the system:

- Machine tool specification constraints:

The maximum speed available	$V <= V_{max}$
The minimum speed available	$V >= V_{min}$
The maximum feed available	$F <= F_{max}$

The maximum feed available $F <= F_{max}$

The minimum feed available $F >= F_{min}$

The maximum depth of cut allowable $D <= D_{max}$

The minimum depth of cut allowable $D >= D_{min}$

The critical tool life allowed

- Machineability constraints:

  The maximum machine horse power available $HPr <= HP$

3.3.2.2 Selection Criterion

Figure 3.6 illustrates the parameters selection procedure applied in the system. The system will first look for correct COMBINATION # and load the record into memory, and then calculate required horse power for machining (HPr) for each set of data and compare to the available horse power (HP). If there are more than one set of data qualified, the system continues looking for the speeds between *Ve* (cutting speed for minimum cost) and *Vm* (cutting speed for maximum production) or the nearest. The calculation for *Ve* and *Vm* will be discussed in detail in section 3.3.4. A subprogram developed in Visual Basic acts as a filter selecting the appropriate conditions from the database. However, it is possible that no appropriate condition is available from the database, thus the system will prompt messages and ask for user's input.

Once the cutting conditions are chosen, the system will compare them to the machine capability and make reasonable modifications. For example, if the speeds or feeds extracted from the database exceed the limits of the machine, the system is designed to use the machine's maximum speed and feed as default instead of the data from the database. For turning, milling and boring processes, the system will repeat the calculations for the horse power required by the new speed and feed and increase the depth of cut within the limits of machine.

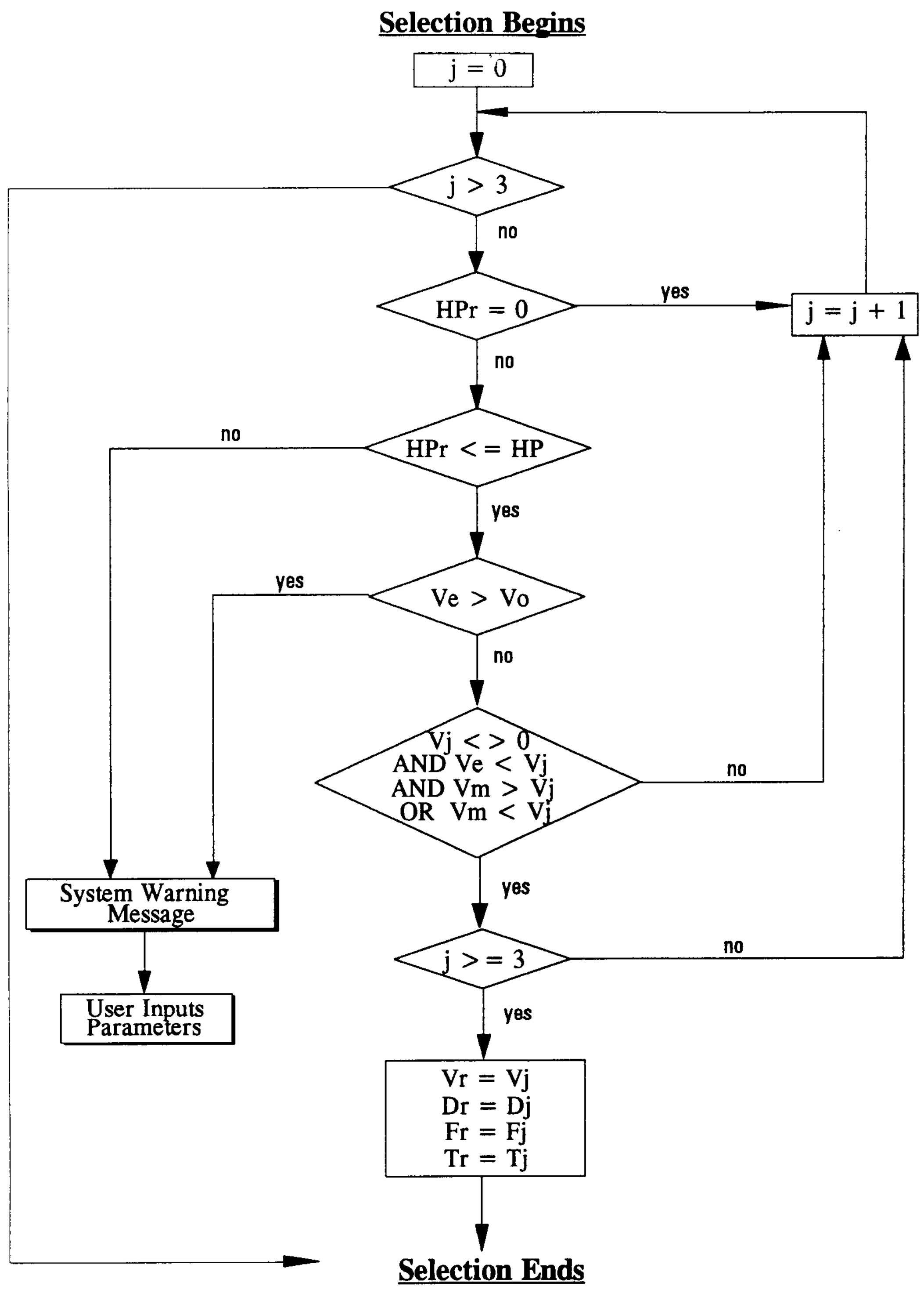

Figure 3.6. Machining Parameters Selection Process

3.3.3 Simulation Model

In this system, SmartCAM is implemented to simulate part processing and generate NC codes. This section describes the simulation model and the essential considerations. The flow chart of the manufacturing model in SmartCAM is shown in Figure 3.7. The information for the NC code generation process comes from four SmartCAM files:

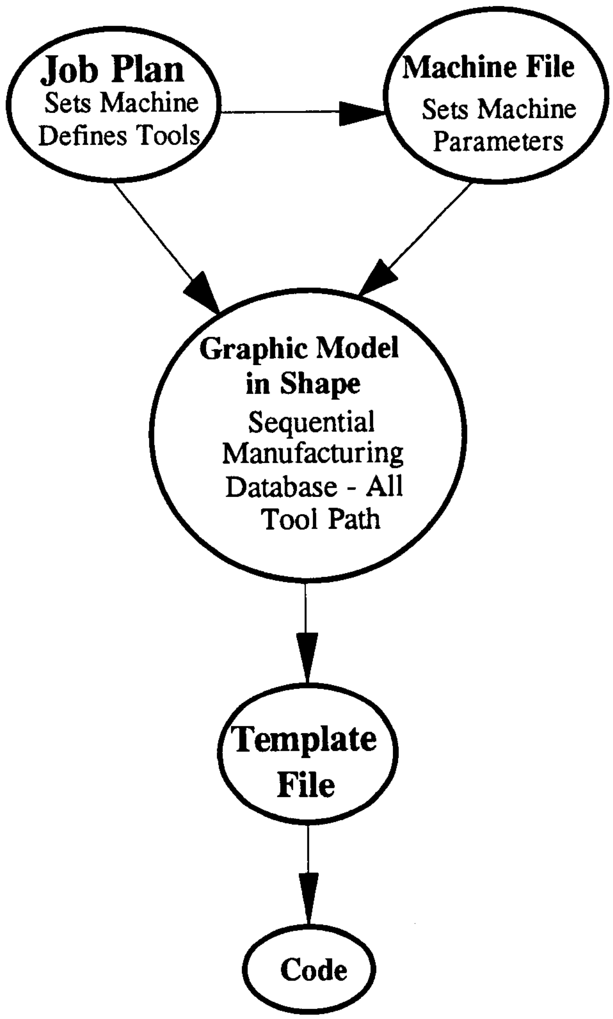

Figure 3.7. Overview of Manufacturing Model in SmartCAM

Shape File The Shape file contains the tool path information for the part's geometric model which has been built in SmartCAM. SmartCAM graphically displays this file on the screen as a CNC process model. When generating code, Shape outputs information about each element to the code generator for processing. SmartCAM will calculate the actual tool path with information from the database and send the tool path information to the files used to generate code. To simulate the tool path correctly, Shape needs information from the Job Plan. For example, Shape must know the type, size and shape of the tool to display it.

Job Plan File The Job Plan contains the list of tools to produce the part. The tool list contains the type, size and shape of tool as well as the tool angle and feeds that controller uses for the process. SmartCAM sends all the Job Plan information to the code generator when it produces code.

Machine File Machine file contains information specific to the machine tool for which the user is generating the code. The machine file is a comprehensive list of questions about machine tool configuration and numeric formats that are necessary to produce NC code. For example, the Shape database knows the element is a line, but it does not know the linear cutting code for the machine tool. This information is retrieved from the Machine File. When SmartCAM generates code for a Shape element, it transfers a numeric value for the appropriate template word through the machine file to the Template file. The switches in the machine file direct the appropriate value to the Template file.

Template File Template file contains information specific to the code format for the machine tool. It also contains information that determines the code output sequence and format. The user can also program the math functions and logic statements. The arrangement and format of this file is the key to producing code exactly as the user wants it. A Template file is divided into sections. In each section, SmartCAM uses template words such as #MOV, and literals such as the letter X, which may be within conditional brackets < >. Each section contains sequences of template words that can generate code for a specific operation. For example, the template section @LINE contains template words for generating linear cutting codes. The contents and arrangement of this @LINE section determine the format of the code output.

Code Output File The Code Output contains the NC machine code or the user's specific purpose file. Figure 3.7 also shows how SmartCAM transfers tool path information from a Shape file to a Template file, and how NC code is generated from a Template file.

3.3.3.1 Template File Program

In SmartCAM, Template file is used to generate output codes, such as NC programs or specific customized reports. The Template file that acts as the link between CAD and simulation is composed of sections and template words which are described in Table 3.3. Although SmartCAM has a template word #TLTIME to estimate the cycle time for each tool type, it does not satisfy the need of separating them into contact and noncontact time. This is why the system needs to generate a specific template file to produce the detailed tooling information. SmartCAM executes requested sections according to the tool path created in the Shape file and extracts the noncontact time (mostly the rapid traverse and dwell time) from the total tooling time. Furthermore, the tool change and reindex time can be calculated whenever a tool change occurs. The detailed program is attached in Appendix A, page 106 - 112. The machining parameters, tool life, tool features and cost for each tool type are stored at the end of the programs. This system writes appropriate template file for two main machining applications, turning and milling, which are to be simulated in SmartCAM.

Table 3.3. Characteristics and Functions of Sections in Template File (including user defined sections)

Section Name	Turning Process	Milling Process
@START	SmartCAM executes this section first before processing any element. The output from this section includes NC block number, a call for the first tool to be used, and the first rapid move of the program.	
@TOOLCHG	This section processes tool changing.	
@TLCHG	A user defined subsection that contains math models to output tool number, cutting time, non-contact time, tool time, cycle time for each tool, tool cost, tool teeth or edge, resharpen times before discarding, tool life and tool description.	
@END	This is the last section that SmartCAM calls after all the elements in the database have been processed. It contains home-return and machine stop codes.	
@ENDROF	SmartCAM executes this section when it encounters the end of a profile. It usually contains retract tool, cancel cutter compensation, and end cutting cycle commands.	
@RAP	For most machines this section processes rapid moves only. This includes moves to the beginning of a profile or to a point.	
@LINE	When SmartCAM encounters a line it outputs code for a linear cutting move.	
@ARC	SmartCAM calls this section when it encounters an arc and outputs code for arcs, both clockwise and counterclockwise.	
@FXD1	Thread cycle	Drill cycle
@FXD2	Turning cycle	Spot drill with dwell
@FXD3	Facing cycle	Tapping cycle
@FXD4	Tapping cycle	Bore cycle
@FXD5	Peck drill cycle	Peck drill cycle
@FXD6	OD or ID groove cycle	not applicable
@FXD7	Face groove cycle	not applicable
@FXCYCLE	not applicable	Subsection that contains basic math models for @FXD processes

Table 3.3. Characteristics and Functions of Sections in Template File (cont.)

Section Name	Turning Process	Milling Process
@FXCYCLE1 @FXCYCLE2 @FXCYCLE3	Subsection that contains basic math models for @FXD processes	not applicable
@ZCHKMV	not applicable	SmartCAM executes this section when it encounters a move to Z-Check
@ZCLRMV	not applicable	SmartCAM executes this section when it encounters a move to Z-Clear
@ZDPTHMV	not applicable	SmartCAM executes this section when it encounters a move to Z-Level
@DWELL	not applicable	SmartCAM executes this section when it encounters a tool dwell situation.
@TOOL<#>	Information of Feeds, Speeds, Depth of Cuts, and Tool Life for each tool is stored in those sections. SmartCAM will retrieve those tool conditions data directly from the template file instead of Job Plan for which it will provide only tool number, offset position, descriptions, tool length, and lead angle, etc..	

3.3.4 Economic Model

Design and production parameters affect part processing cost. Machining parameters used in part processing need to satisfy conflicting requirements. For example, as cutting speed is increased, it reduces processing time. However, tool wear increases resulting in more frequent tool changing and higher tool cost. The economic model used in this research consists of five basic cost components [4]: machining cost, tool cost, tool changing cost, part handling cost, and set-up cost. Individual cost components are calculated on the basis of unit processing cost as follows:

$$\text{Tool cost} = \sum_{i=1}^{k} (C_{t_i} \frac{t_i}{T_i})$$

$$\text{Tool changing cost} = C_D \sum_{i=1}^{k} (t_i \frac{t_{ci}}{T_i})$$

$$\text{Loading/unloading cost} = C_D\, t_l$$

$$\text{Machining cost} = C_D \sum_{i=1}^{k} t_i$$

$$\text{Set-up cost} = \frac{C_s}{L}$$

C_D	direct labor cost, $/min	C_{ti}	initial tool cost for tool type i, $
T_i	tool life for tool type i, min	C_s	set-up cost per lot

t_i cutting time per piece for tool type i, min

t_{ci} tool changing or reindexing time for tool type i, min

t_l part loading and unloading time, min

k no. of different tools required for part processing

L parts per lot

The rate of metal removal and power requirements for machining affect machining cost and are dependent upon cutting speed, depth of cut and feed rate. Expressions for computing metal removal rate and machine horsepower requirement are as follows [8]:

Machine, rpm $N = \dfrac{V}{\pi \times d}$

Feed rate, F (ipm) $= N \times f$ for Turning ($= f_t\ n_t\ N$ for Milling)

Cutting time, t (min) $= \dfrac{(l + \Delta l)}{F}$

Metal removal rate, Q ($^{in^3}\!/_{min}$) $= 12\ V\ F\ D$ for Turning ($w\ F\ D$ for Milling)

Spindle horsepower required, $HPs = Q\ HPu$

Machine horse power required, $HPr = \dfrac{HPs}{E}$

d	tool or work diameter	f	feed, in/rev
f_t	feed per tooth	n_t	no. of teeth
l	length of cut	Δl	cutter approach and over travel, in
D	depth of cut, in	w	width of cut, in
V	cutting speed, ft/min	HP_u	unit horsepower in^3/min/hp
E	motor drive efficiency		

The metal removal rate can be maximized by using the largest possible depth of cut without overpowering the machine. This system protects the user from selecting a depth of cut and corresponding cutting speed from the database which can result in motor horsepower requirements in excess of available horsepower. These variables are used in conjunction with the database to compute the metal removal rate.

Tool life affects tool cost and tool changing cost. Several models have been proposed for estimating tool life and discussed in the database section. The model used in this research is given by [8]:

$$VT^n = C$$

where *n* and *C* are constants whose values depend on work and tool material combination. Values for *n* and *C* for work and tool materials specified earlier are provided in the database.

Minimum cost and maximum production rate are two major criterion used for determining the range of cutting speed [8]. The cutting speed for minimum cost is derived by differentiating the total machining cost with respect to cutting speed, and setting the results equal to zero. The cutting speed for minimum cost is given by,

$$V_e = \frac{C}{\left[\left(\frac{1}{n} - 1\right) \left(\frac{C_D t_c + C_t}{C_D} \right) \right]^n}$$

The tool life for minimum cost is given by,

$$T_e = \left[\frac{C}{V_e} \right]^{\frac{1}{n}}$$

Maximum production rate cutting speed is obtained by substituting the tool cost equal to zero in the equation for the cutting speed for minimum cost. The equation for the cutting speed that corresponds to maximum production rate is,

$$V_m = \frac{C}{\left[\left(\frac{1}{n} - 1\right) t_c \right]^n}$$

The tool life for maximum production rate is given by,

$$T_m = \left[\frac{C}{V_m} \right]^{\frac{1}{n}}$$

3.3.5 System Integration

From an implementation point of view, the system shown in Figure 3.8 is completely integrated. Visual Basic [9] has been extensively used for designing various modules in this system. The user-system interface/shell and the cost evaluation module are designed in Visual Basic and shown in Figure 3.9. The database containing data on several work and tool materials and machining processes is designed in Quattro Pro [10] and is compatible with Paradox [11]. The data is retrieved using the built-in database engine in Visual Basic.

SmartCAM [12] has been used as a CAD/CAM package in this system. A post-processor program makes NC programs generated by SmartCAM compatible with Emco Maier CNC machines. Macros in SmartCAM have been designed to enhance the capabilities of SmartCAM. Due to the enhancements in SmartCAM the system can determine the total machining time as is done under normal SmartCAM operation as well as the amount of contact time between each tool and the work. The contact time between each tool and the work is required for determining part processing cost and producing tool changing schedules.

The user selects work and tool materials and provides information about process and tool set-up interactively using several screens. Full editing capability is provided to the user. User provided information along with process parameters extracted from the database are used by the system to create Template, Job Plan and Machine Define files in SmartCAM. CAD modeling data is stored by the system in the Shape file. At this stage the system takes over and generates machining times data including total machining time and contact time between each tool and the work. The system substitutes this information along with other data into the cost model to determine unit processing cost and presents results to the user.

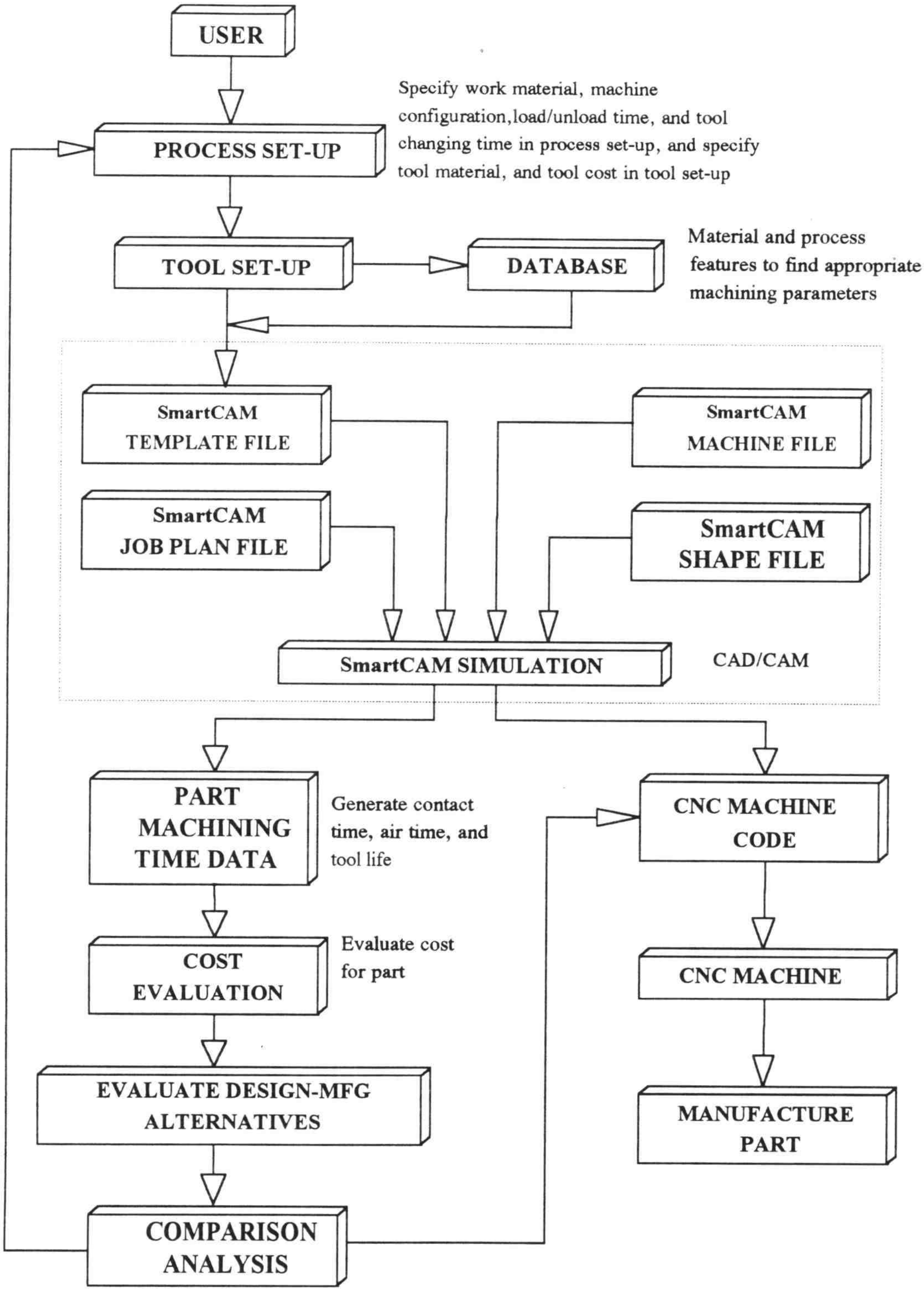

Figure 3.8. Information Flow Among Various Modules

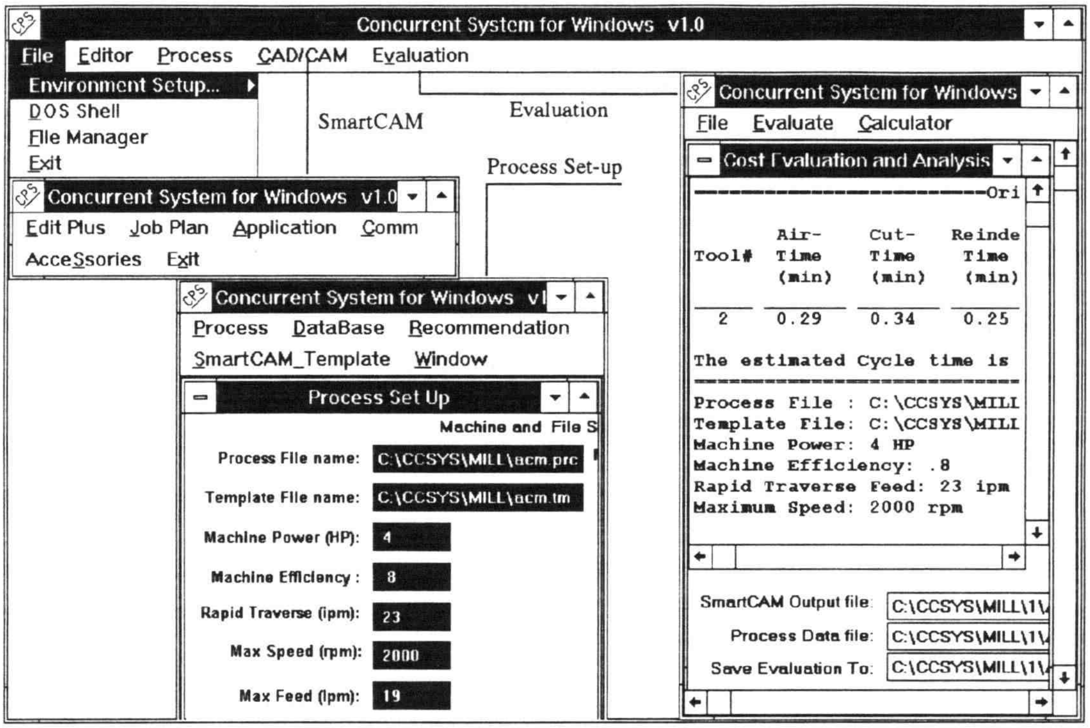

Figure 3.9. The Integrated System Designed in Visual Basic

3.3.6 Sensitivity Analysis

Sensitivity analysis concentrates on the simplification of the product specifications to promote ease of manufacture, improve quality, and reduce manufacturing costs. For example, a change in work or tool materials or a change in one or more design and production specifications may result in substantial cost reductions, and improvements in tooling, fixturing and material handling required to support the process. The availability of a CAD/CAM package in this system allows quick changes in the shape of the part in addition to changes in tool and work materials, and production parameters.

CHAPTER 4

SYSTEM UTILIZATION

This chapter presents an application example to demonstrate how this integrated system works on multi-tool machining centers.

4.1 Example Description

Consider the manufacturing of the part shown in Figure 4.1. This part requires turning, cut-off and milling operations. Turning and cut-off are to be performed on the same CNC lathe machine, and the slot will be processed on the milling machine.

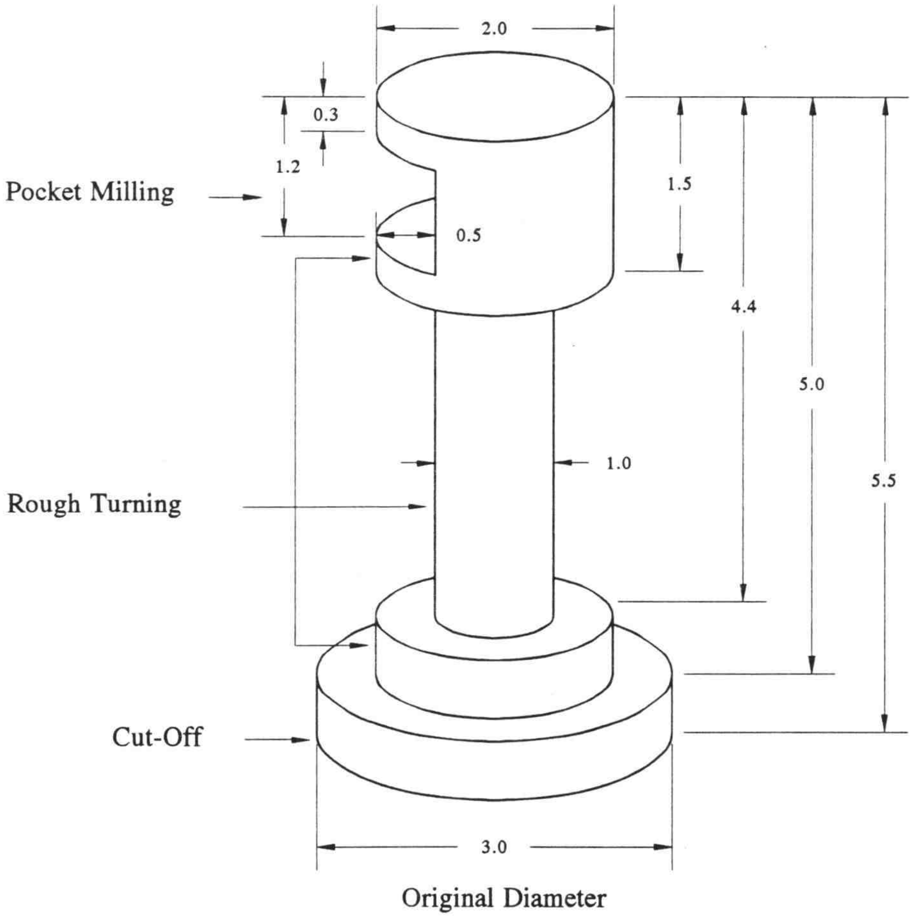

Figure 4.1 Example Part Drawing

4.1.1 Work and Tool Materials

Assume the part material to be the alternative of aluminum alloy, and low carbon steel. Either high speed steel (HSS) or carbide tool may be used as the tool material.

4.1.2 Machine Tool Specifications and Operation Data

The manufacturing component in this design-manufacturing example is implemented using an Emco-Maier manufacturing cell consisting of a CNC lathe and a CNC mill. The specifications of the machines and the operation data are given in Table 4.1.

Table 4.1. Machine Tool Specifications and Operation Data

	Lathe	**Mill**
Machine power (HP)	4	4
Machine Efficiency	0.8	0.8
Rapid Traverse Feed (ipm)	26.5	26.5
Maximum Speed (rpm)	2000	2000
Maximum Feed (ipm)	19.9	23
Maximum Depth of Cut (inch)	0.5	1.5
Tool Change Time (min)	3.3	1.5
Tool Reindex Time (min)	.25	.25
Operation Cost ($ / hr)	32	32
Load / Unload Time (min)	3	1.5
Machine Setup Time (min)	15	15
Number of Tools Used	3	1
Lot Size per Batch (piece)	50	50

4.2 Implementation Procedures

The five steps to implement this application are:

1) Process Set-Up: The user selects work and tool materials and provides standard time and cost data through the system interactive screen. The system extracts data on feeds, depth of cut and speeds from the database for user specified work and tool material combination and displays it along with recommended cutting speeds and tool lives for minimum cost and maximum production rate for each tool.

2) Simulation: The user may select cutting speed for minimum cost, maximum production rate, or a compromise between these two values. The manufacturing process is then simulated; the simulation includes showing the part on the computer screen with the machining taking place. The system calculates contact and non-contact machining times, tool life for each tool, and total machining or cycle time for processing the part. The system also calculates the frequency of tool change for each tool.

3) Evaluation: The system calculates unit machining cost for each tool, and the total processing cost per part.

4) Analysis: To investigate the optimal process parameters and tool/work material combination for manufacturing a part, the sensitivity analysis is employed by comparing the cost evaluation results.

5) NC Codes Generation: After the machining parameters are determined, the part can be manufactured. SmartCAM is used for generating the NC codes and communicating with NC machines.

4.2.1 Process Set-Up

The part shown in Figure 4.1 may be produced with four combinations of work and tool materials which are shown in Table 4.2. Figure 4.2 shows the user-input screen for Process information. The user selects work and tool materials and provides information about process and tool setup interactively in Process screen which is divided into seven portions:

Table 4.2. Combinations of Work and Tool Material

no.	Work material name	Work material code	Hardness Range	Tool material name	Tool material code
1	Aluminum	alum	30-150	Carbide	C2
2	Aluminum	alum	30-150	HSS	M2
3	Low Carbon	1011	225-275	Carbide	C5
4	Low Carbon	1011	225-275	HSS	M2

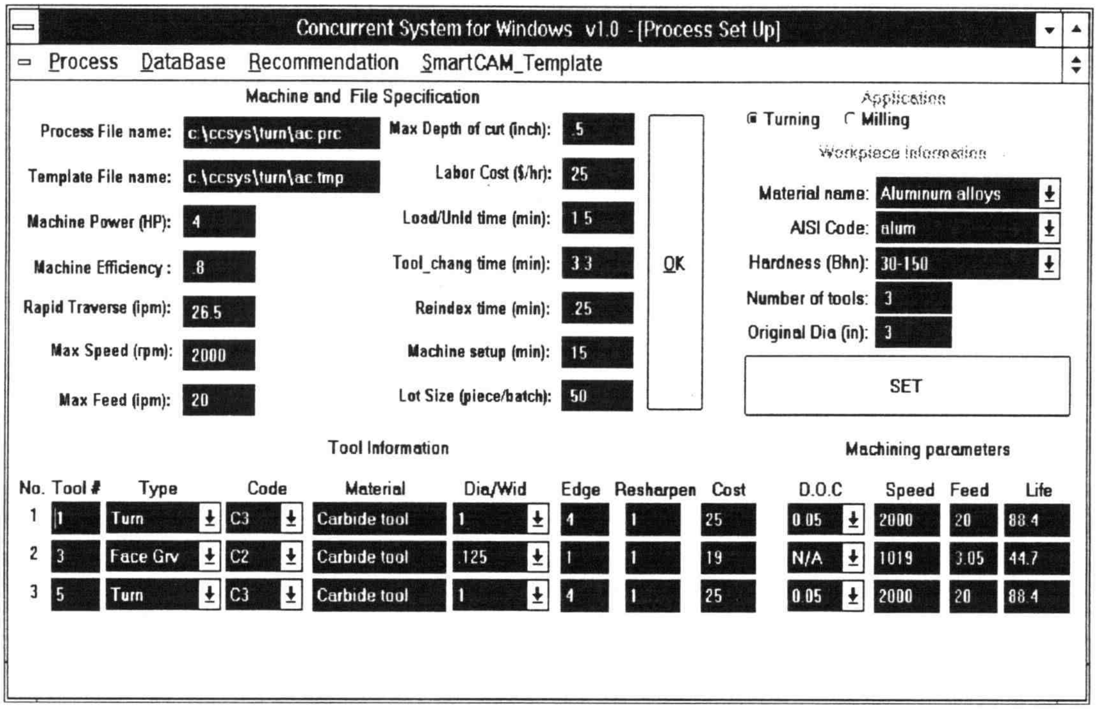

Figure 4.2. Process Information and Parameters Input Screen

1) Application: Turning and Milling applications can be applied in this system. In this example, cut-off and turning are processed in Turning and pocket milling in Milling application.

2) Machine and File Specification: When building a new process, the user provides necessary machine specifications and files management as follows:

- **Process File name:** Specify the Process file path and name to save or retrieve.
- **Template File name:** Specify the SmartCAM template file to be written to.
- **Machine power (HP):** Indicate the available machine's horse power.
- **Machine efficiency:** Indicate the efficiency of machine valued from 0 to 1.
- **Rapid Traverse (rpm):** The machine's rapid traverse speed.
- **Max Speed (rpm), Max Feed (ipm), Max Depth of Cut (in):** Indicate the maximum machining conditions the machine can handle.
- **Labor cost ($/Hr)**
- **Load/Unload Time (min):** The workpiece load and unload time.
- **Tool_Changing Time (min):** The average time to replace worn tool.
- **Reindex Time (min):** The average time to switch tool for a different operation.
- **Machine Set-up (min):** The average time to set-up machine for a new batch.
- **Lot Size (piece/batch):** The number of parts to be produced in each batch.

3) Workpiece Information: The user can select a variety of work materials from a database-driven list and provide detailed data for process.

- **Material name:** User can specify the work material from the work material list.
- **Material AISI Code:** User indicates the material AISI code from the work material list.
- **Hardness (BHN):** User indicates the hardness range of material from the work material list.
- **Number of Tools:** Specify how many tools are used in the process.
- **Original Dia (in):** The original workpiece diameter to be processed in turning.

4) Tool Information:

- **Tool #:** This tool number should be consistent with the tool number defined in Job Plan and Shape file and must not be duplicated.
- **Tool Type:** User chooses from a list of tool types pre-defined in the database.
- **Code:** The AISI tool material code defined in the database.
- **Material:** The tool material name (will be displayed automatically when tool material code is selected).
- **Dia/Wid:** The diameter or width of cutter.

- **Edge/Teeth**: The number of edges of a turning insert or the number of teeth of a milling cutter.
- **Resharpen**: The number of times a cutter can be resharpened before being discarded.
- **Cost**: The tool price.

5) Machining Parameters: If above information has been correctly given, user can start machining parameters selection with three options: 1) handbook database (default); 2) minimum cost; 3) maximum production rate. The system will select the qualified parameters and display them. The parameters are:

- **Depth of cut (in)**
- **Speed:** The spindle speed (rpm)
- **Feed:** The cutting feed rate (ipm)
- **The estimated tool life (min)**

To this point, user still has the power to alter machining parameters. However, all modifications will be reevaluated based on the extracted constant data n, C and HPu.

6) Recommendation: If the machining parameters have been correct, the system will pop up a report screen describing the recommended contents when the user clicks Recommendation on the menu bar. Appendix C lists the system-generated summaries of user specifications and machining parameters extracted from the database for all required tools for the aluminum alloy and carbide tool combination.

7) SmartCAM_Template: The last but the most important task in Process Set-up is to generate a SmartCAM template file to be used for computing the time data in the simulation module. User may have this done by clicking Smart_Template on menu bar.

4.2.2 CAD/CAM Model and Simulation in SmartCAM

The second stage is to build up CAD/CAM models in SmartCAM to simulate the cycle time of manufacturing the part, generate the NC codes, and manufacture the part eventually. As stated in the last chapter, the four files, Job Plan, Machine, Shape, and Template are established at this stage. SmartCAM provides ready-to-use Machine file for each different kind of NC machine. The cutting tool data are defined in the Job Plan and listed in Table 4.3.

Table 4.3. Cutting Tools Used in the Processes

Tool Number	Turning Process	Milling Process
1	Right-hand turn	N/A
3	0.120" width cut-off tool	N/A
5	Left-hand turn	N/A
2	N/A	0.375" dia end mill

Figures 4.3 graphically displays the CAD models developed in SmartCAM for turning and milling processes respectively, in which the tool paths are shown in 3D. The Template files for generating time data were created in the previous stage. However, because different depths of cut are recommended for processing different work-tool material combinations, both the roughing in turning and pocket milling should be able to use various depths of cut as the combination changes. Macro programs were developed during the process of building the CAD models and are revised for customized use which makes the change of design an easy task. Detailed descriptions of the macro programs are listed in Appendix B. By combining these four files, the cycle time data can be generated and the output sent to a sequential text file.

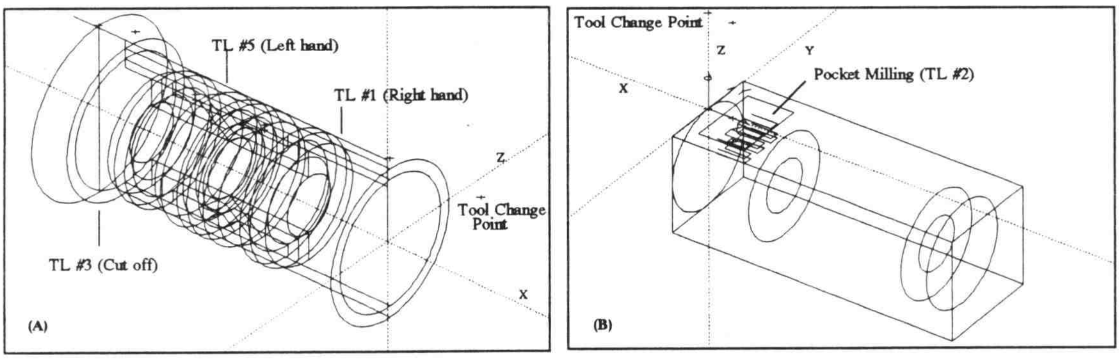

Figure 4.3. (A) 3D Turning Process (B) 3D Milling Process Model

4.2.3 Cost Evaluation

The third stage is to evaluate the tooling cost for each combination. Figure 4.4 shows the evaluation screen of the system where the user needs to specify the names of three files. They are:

1. The cycle time output file produced by SmartCAM,
2. The process information file, and
3. The evaluation output file.

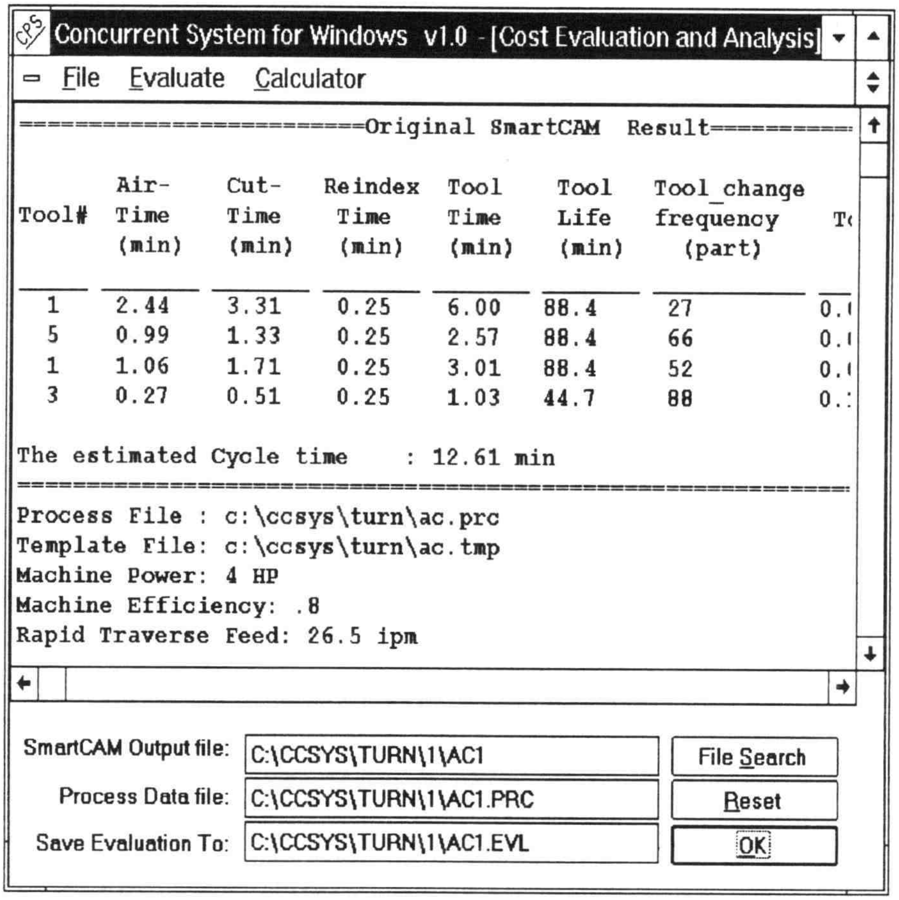

Figure 4.4. Evaluation Screen

The evaluation report will be displayed on the screen and also saved to a text file. This report includes the following information:

1) Original SmartCAM Result:

The summary of the simulation output for each tool includes:

- **Tool #**: The sequence order of the tools has been used.
- **Air-Time (min)**: The non-contact move time.
- **Cut-Time (min)**: The tool-work contact cutting time.
- **Reindex Time (min)**: The time to switch a tool for next operation.
- **Tool Time (min)**: The total cycle time for each tool.
- **Tool Life (min)**: The estimated tool life.
- **Tool_change Frequency**: The frequency to replace a tool, it is the quotient of Tool Life divided by Cut-Time.
- **Tool Description**: Brief description of tool specification.
- **The estimated Cycle time**: The sum of all Tool Times.

2) Process Information Summary:

It reports the file paths, machine's specifications, operation data, and work material properties.

3) Tool Information:

This tablet displays information on tool number, type, material, AISI code, diameter/width, edge/teeth, resharpen times, and costs.

4) Revised Simulation Result:

Sometimes the tools are used in the process repeatedly therefore we need to sum them up. For example, in Appendix D, the original output from SmartCAM shows that tool #1 is used twice in the turning process, so all tooling time data of tool #1 should be combined except tool life. The Revised Simulation Result reports:

- **Air-Time (min):** The time for rapid traverse move, tool reindex, and other non-contact moves.
- **Cut-Time (min):** The time for feeding and other contact cutting.

- **Tool Time (min):** The tool cycle time for each tool.
- **Tool Life (min):** The estimated tool life.
- **Tool_change frequency:** The frequency of replacing a tool.
- **Tool Description**
- **The estimated cycle time:** The Sum of above time data.

5) Evaluation Report:

The evaluation alters the time data into cost data, which includes:

- **Rapid Trav Cost ($)**: The costs of rapid traverse or non-contact move.
- **Feed Cost ($)**: The costs of feed or contact cutting.
- **Tool Change Cost ($)**: The costs of replacing failed tools.
- **Tool Depr Cost ($):** Tool depreciation costs.
- **The Load /Unload cost:** The cost of loading and unloading a part.
- **The Set-up cost:** The average machine setup cost for each part in a batch.
- **The Total Processing cost per piece:** Sum of above costs.

4.2.4 Sensitivity Analysis

Several feasible work/tool material combinations can be used in this integrated system in order to get estimated cost data for sensitivity analysis. The analyses are made under three machining parameters selection criterion(Machining Data Handbook, minimum cost, and maximum production) to approach the optimal work/tool material combination.

1. Selecting machining parameters from Machining Data Handbook database

Figures 4.5 shows the recommended machining conditions and estimated time and tooling cost for each tool. Figure 4.6 indicates the evaluated costs for different material combinations based on the machining parameters from the Machining Data Handbook recommendations. It shows that aluminum and carbide tool combination has the lowest cost of $8.58 per part. Aluminum can be machined at a much faster rate than low carbon steel using carbide tools.

2. Selecting machining parameters by the minimum cost criteria

Figures 4.7 shows the recommended machining conditions by the minimum cost criteria, and the estimated time and tooling cost for each tool. Figure 4.8 indicates the evaluated total costs for different material combinations. It shows that high speed steel tool costs lower than carbide tool in aluminum, and low carbon steel and carbide tool combination has a lower machining cost of $22.59 per part than HSS combination. Aluminum and high speed steel tool combination has the lowest cost ($7.94 per part).

3. Selecting machining parameters by the maximum production rate criteria

Figures 4.9 shows the results by the maximum production rate criteria and Figure 4.10 indicates the evaluated costs for different material combinations. Note that high speed steel tool costs lower than carbide tool for processing aluminum, and low carbon steel and HSS combination has lower cost of $29.17 per part, however, the low carbon steel and carbide tool combination has higher production rate (shorter cycle time) compared with low carbon steel and HSS tool combination.

4. Optimal combination for minimum cost

Since the system calculates production cost for each type of tool, one can use different tool materials for different tools and may even get lower total cost than the cost obtained for a specific work-tool material combination. For example the lowest possible total cost for aluminum can be obtained by choosing HSS for tool #1, carbide for tool #3, HSS for tool #5 and HSS for tool #2. The lowest possible cost is $8.11 (Figure 4.11 and 4.12) compared with $8.58 for aluminum-carbide tool combination as reported in Figure 4.6.

This system calculates the tooling cost and production rate for each different work/tool material combination for optimal production environments. In some instances, the estimated tooling costs of different combinations are so close that it may not be easily determined; for example, the tooling costs for aluminum and carbide or HSS combination in Figure 4.10 are almost the same ($9.09 vs. $9.07). However, the factors that influence the final decision are not limited to lower cost or higher production rate. The material's mechanical, physical, and manufacturing properties, the reliability and availability of supply, and costs of accessories can be included in consideration for sensitivity analysis.

4.2.5 NC Code Generation

From the above analysis, the lowest cost option was selected and NC programs for milling and turning on Emco Maier CNC machines were generated as shown in Appendix E. Since this system contains a CAD/CAM package the user may modify the shape of the part for the same function to investigate the effect of a change in part shape on part processing cost.

TOOL #	WORK +TOOL	SUGGESTED MACHINE DATA				MACHINING TIME			COST DATA				
		D (in)	V (rpm)	F (ipm)	T (min)	RAPD (min)	CUT (min)	CYCL (min)	RAPD ($)	CUT ($)	TL CHNG	TL DEPR	COST ($)
#1	Al+Carb	0.05	2000	20	88	4.09	5.04	9.13	1.71	2.10	0.08	0.36	4.24
	Al+HSS	0.12	1273	8.9	20	2.34	4.84	7.18	0.98	2.02	0.34	0.50	3.83
	Low C+Carb	0.04	637	4.46	13.38	4.64	21.05	25.69	1.93	8.77	2.16	9.82	22.68
	Low C+HSS	0.17	159	1.11	4.3	2.07	22.47	24.54	0.86	9.36	7.18	10.45	27.86
#3	Al+Carb	--	1019	3.06	44.7	0.52	0.51	1.03	0.22	0.21	0.02	0.22	0.66
	Al+HSS	--	255	0.77	45.0	0.52	2.03	2.55	0.22	0.84	0.06	0.32	1.44
	Low C+Carb	--	369	0.48	45.4	0.52	3.23	3.75	0.22	1.35	0.10	1.35	3.01
	Low C+HSS	--	115	0.15	50.1	0.52	10.37	10.89	0.22	4.32	0.28	1.45	6.27
#5	Al+Carb	0.05	2000	20	88	1.24	1.33	2.57	0.52	0.55	0.02	0.09	1.18
	Al+HSS	0.13	1273	8.9	20	0.73	1.4	2.13	0.30	0.58	0.10	0.14	1.13
	Low C+Carb	0.05	637	4.46	13.38	1.49	6.21	7.7	0.62	2.59	0.64	2.90	6.75
	Low C+HSS	0.18	159	1.11	4.3	0.65	7.26	7.91	0.27	3.03	2.32	3.38	9.00
#2	Al+Carb	0.5	2000	12	31852	0.54	0.34	0.88	0.22	0.14	0.001	0.001	0.36
	Al+HSS	0.5	2000	12	636042	0.54	0.34	0.88	0.22	0.14	0.0001	0.0001	0.36
	Low C+Carb	0.19	2000	8	828.7	0.61	1.53	2.14	0.25	0.63	0.001	0.05	0.95
	Low C+HSS	0.5	1222	2.44	45	0.54	1.67	2.21	0.22	0.69	0.02	0.48	1.42

Figure 4.5. Recommended Machining Parameters, Simulated Time and Tooling Cost (Machining Data Handbook)

RAPD= Rapid traverse, CUT= Feed cutting, CYCL= Cycle time, TL CHNG= Tool change, TL DEPR= Tool depreciation

WORK + TOOL	MACHINING TIME (min)			COST DATA ($)						
	RAPD	CUT	CYCL	RAPD	CUT	TL CHNG	TL DEPR	LD/UNLD	SETUP	CPP
Al+Carb	6.39	7.22	13.61	2.66	3.01	0.11	0.67	1.88	0.25	8.58
Al+HSS	4.13	8.61	12.74	1.72	3.59	0.50	0.96	1.88	0.25	8.89
Low C+Carb	7.26	32.02	39.28	3.03	13.34	2.90	14.12	1.88	0.25	35.52
Low C+HSS	3.78	41.77	45.55	1.57	17.41	9.81	15.76	1.88	0.25	46.68

Figure 4.6. Estimated Total Tooling Cost Per Part (Machining Data Handbook)

LD/UNLD= Load/unload, CPP= Cost per part

TOOL #	WORK +TOOL	SUGGESTED MACHINE DATA				MACHINING TIME			COST DATA				
		D (in)	V (rpm)	F (ipm)	T (min)	RAPD (min)	CUT (min)	CYCL (min)	RAPD ($)	CUT ($)	TL CHNG	TL DEPR	COST ($)
#1	Al+Carb	0.05	2000	20	88	4.09	5.04	9.13	1.71	2.10	0.08	0.36	4.24
	Al+HSS	0.13	1115	7.80	56.4	2.15	4.59	6.74	0.90	1.91	0.11	0.16	3.08
	Low C+Carb	0.06	453	3.17	73.6	3.73	21.99	25.72	1.55	9.16	0.41	1.87	12.99
	Low C+HSS	0.24	115	0.80	57.4	1.78	22	23.78	0.74	9.17	0.53	0.77	11.21
#3	Al+Carb	--	765	2.29	121	0.52	0.68	1.20	0.22	0.28	0.01	0.11	0.61
	Al+HSS	--	221	0.66	119.2	0.52	2.34	2.86	0.22	0.97	0.03	0.14	1.35
	Low C+Carb	--	278	0.36	121.3	0.52	4.29	4.81	0.22	1.79	0.05	0.68	2.73
	Low C+HSS	--	101	0.13	121.2	0.52	11.81	12.33	0.22	4.92	0.13	0.68	5.95
#5	Al+Carb	0.05	2000	20	88	1.33	1.37	2.7	0.55	0.57	0.02	0.10	1.24
	Al+HSS	0.13	1115	7.80	56.4	0.73	1.47	2.2	0.30	0.61	0.04	0.05	1.01
	Low C+Carb	0.06	453	3.17	73.6	1.15	6.36	7.51	0.48	2.65	0.12	0.54	3.79
	Low C+HSS	0.24	115	0.80	57.4	0.56	7.74	8.3	0.23	3.22	0.19	0.27	3.91
#2	Al+Carb	0.5	2000	12	31852	0.54	0.34	0.88	0.22	0.14	0.001	0.001	0.36
	Al+HSS	0.5	2000	12	636042	0.54	0.34	0.88	0.22	0.14	0.0001	0.0001	0.36
	Low C+Carb	0.19	2000	8	828.7	0.61	1.53	2.14	0.25	0.63	0.001	0.05	0.95
	Low C+HSS	0.5	989	1.97	189.8	0.54	2.07	2.61	0.22	0.86	0.007	0.14	1.23

Figure 4.7. Recommended Machining Parameters, Simulated Time and Tooling Cost (Minimum Cost)

RAPD= Rapid traverse, CUT= Feed cutting, CYCL= Cycle time, TL CHNG= Tool change, TL DEPR= Tool depreciation

WORK + TOOL	MACHINING TIME (min)			COST DATA ($)						
	RAPD	CUT	CYCL	RAPD	CUT	TL CHNG	TL DEPR	LD/UNLD	SETUP	CPP
Al+Carb	6.48	7.43	13.91	2.70	3.10	0.11	0.56	1.88	0.25	8.59
Al+HSS	3.94	8.74	12.68	1.64	3.64	0.17	0.35	1.88	0.25	**7.94**
Low C+Carb	6.01	34.17	40.18	2.50	14.24	0.58	3.15	1.88	0.25	**22.59**
Low C+HSS	3.40	43.62	47.02	1.42	18.17	0.85	1.86	1.88	0.25	24.43

Figure 4.8. Estimated Total Tooling Cost Per Part (Minimum Cost)

LD/UNLD= Load/unload, CPP= Cost per part

TOOL #	WORK +TOOL	SUGGESTED MACHINE DATA				MACHINING TIME			COST DATA				
		D (in)	V (rpm)	F (ipm)	T (min)	RAPD (min)	CUT (min)	CYCL (min)	RAPD ($)	CUT ($)	TL CHNG	TL DEPR	COST ($)
#1	Al+Carb	0.05	2000	20	88	4.09	5.04	9.13	1.71	2.10	0.08	0.36	4.24
	Al+HSS	0.12	1247	8.72	23	2.34	4.91	7.25	0.98	2.04	0.29	0.43	3.74
	Low C+Carb	0.04	638	4.46	13.3	4.64	21.09	25.73	1.93	8.79	2.18	9.91	22.81
	Low C+HSS	0.21	129	0.90	22.9	1.78	20.05	21.83	0.74	8.35	1.20	1.75	12.05
#3	Al+Carb	--	1663	4.98	8.2	0.52	0.31	0.83	0.22	0.13	0.05	0.72	1.12
	Al+HSS	--	289	0.86	19.2	0.52	1.79	2.31	0.22	0.74	0.13	0.65	1.74
	Low C+Carb	--	605	0.78	8.1	0.52	1.97	2.49	0.22	0.82	0.33	4.62	5.99
	Low C+HSS	--	132	0.17	19.6	0.52	9.03	9.55	0.22	3.76	0.63	3.23	7.84
#5	Al+Carb	0.05	2000	20	88	1.33	1.37	2.7	0.55	0.57	0.02	0.10	1.24
	Al+HSS	0.12	1247	8.72	23	0.73	1.41	2.14	0.30	0.59	0.08	0.12	1.10
	Low C+Carb	0.04	638	4.46	13.3	1.49	6.28	7.77	0.62	2.62	0.65	2.95	6.84
	Low C+HSS	0.21	129	0.90	22.9	0.56	7.26	7.82	0.23	3.03	0.44	0.63	4.33
#2	Al+Carb	0.5	2000	12	31852	0.54	0.34	0.88	0.22	0.14	0.001	0.001	0.36
	Al+HSS	0.5	2000	12	636042	0.54	0.34	0.88	0.22	0.14	0.0001	0.0001	0.36
	Low C+Carb	0.19	2000	8	828.7	0.61	1.53	2.14	0.25	0.63	0.001	0.05	0.95
	Low C+HSS	0.5	1556	3.11	8.7	0.54	1.31	1.85	0.22	0.54	0.09	1.96	2.82

Figure 4.9. Recommended Machining Parameters, Simulated Time and Tooling Cost (Maximum Production rate)

RAPD= Rapid traverse, CUT= Feed cutting, CYCL= Cycle time, TL CHNG= Tool change, TL DEPR= Tool depreciation

WORK + TOOL	MACHINING TIME (min)			COST DATA ($)							TIME
	RAPD	CUT	CYCL	RAPD	CUT	TL CHNG	TL DEPR	LD/ UNLD	SETUP	CPP	Total Cycle
Al+Carb	6.48	7.06	13.54	2.70	2.94	0.15	1.17	1.88	0.25	9.09	18.4
Al+HSS	4.13	8.45	**12.58**	1.72	3.52	0.51	1.20	1.88	0.25	**9.07**	**18.3**
Low C+Carb	7.26	30.87	**38.13**	3.03	12.87	3.17	17.54	1.88	0.25	38.73	**50.24**
Low C+HSS	3.40	37.65	41.05	1.42	15.69	2.37	7.57	1.88	0.25	**29.17**	51.24

Figure 4.10. Estimated Total Tooling Cost Per Part (Maximum Production rate)

Total Cycle= Total cycle time (including tool changing, load/unload and set-up time)

Work Material	Criterion	TL #1	TL #3	TL #5	TL #2
Aluminum	Database	HSS	Carbide	HSS	HSS
	Minium cost	HSS	Carbide	HSS	HSS
	Maxium rate	HSS	Carbide	HSS	HSS
Low Carbon	Database	Carbide	Carbide	Carbide	Carbide
	Minium cost	HSS	Carbide	Carbide	Carbide
	Maxium rate	HSS	Carbide	HSS	Carbide

Figure 4.11. Summarized Work/Tool Material Combinations

Criterion	Work Material	TL #1 ($)	TL #3 ($)	TL #5 ($)	TL #2 ($)	Subtotal ($)	LD/UNLD +SETUP	Total Cost ($/part)	Cycle Time *(min/part)
Machining Data Handbook	Aluminum	3.83	0.66	1.13	0.37	5.98	**2.13**	**8.11**	16.82
	Low carbon	22.68	3.01	6.75	0.95	33.39	**2.13**	35.52	50.74
Minimum Cost	Aluminum	3.08	0.61	1.01	0.37	5.07	**2.13**	**7.20**	16.44
	Low carbon	11.20	2.73	3.79	0.95	18.68	**2.13**	20.81	44.42
Maximum Production rate	Aluminum	3.74	1.12	1.10	0.37	6.32	**2.13**	**8.45**	16.53
	Low carbon	12.05	5.99	4.33	0.95	23.33	**2.13**	25.46	43.51

Figure 4.12. Estimated Costs and Cycle Time for Summarized Combinations

*Cycle Time includes tool changing, load/unload and set-up time

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

This thesis has discussed the practical and systematic steps in achieving the product design, production and economic analysis by developing an optimization system centered around CAD/CAM package. Through the discussion and the example demonstration in the previous chapters, a summary of this system and a number of suggestions are presented in this chapter.

5.1 Conclusions

The system has been developed to assist in integrated conceptual design. This phase of product design involves selection of work and tool material, and the manufacturing processes to produce the desired product characteristics, and analysis of the processing system in terms of operational parameters and production economics. The system with an interactive user interface was developed to obtain product design, process, and cost information allowing user friendly input of a machine's capability information such as machine power, efficiency, speed limit, depth of cut, feed rate, and other related information, and also input of work and tool material and process characteristics which are extracted from the system database in order to calculate the best machining conditions. In addition, using Taylor's equation, tool life estimates are made and a tool replacement policy is scheduled. This information along with product design specifications from the CAD model is used to obtain operating parameters for optimal manufacturing environment and production economics.

The program developed for this research was coded in Visual Basic 3.0 and compiled as a Microsoft Windows executable. Instead of writing a single program, the program uses a CAD/CAM software (SmartCAM) to simulate processing time and generate NC codes. The purpose of doing so is to employ the full advantages of this CAD/CAM software. Data output including recommended machining parameters and cost analysis reports can be viewed on the screen and output as a standard ASCII text file.

A product's status cannot be assessed independent of available technology. This system accomplishes that task by integrating product design characteristics and manufacturing operations. A wide range of alternatives including work/tool material combinations and process types is created as an information base; three optimization criteria (1. Machining Data Handbook, 2. Minimum cost, 3. Maximum production rate) help selecting machining parameters; changing the geometry of part while satisfying the functional requirements can be analyzed without prior commitment to any particular alternative.

5.2 Recommendations

5.2.1 Development of Cost Analysis Associated with Product Design Requirements

The cost analysis involved currently in this system does not include the costs caused by the change of product design requirement. The cost of producing a part greatly varies if different tolerances are requested. For example, a part with a tolerance of $\pm$ 0.0001 unit required is much costly than a similar geometry part with $\pm$ 0.1 unit tolerance required. However, the product function requirements are commonly the key factors that determine manufacturing methods and costs. To be more accurate on estimating manufacturing costs, establishing an advanced model for cost analysis associated with product design requirements will be helpful.

5.2.2 Development of a Concurrent System

Although this system includes the capabilities of CAD, machining parameter database, process simulation, economic analysis, and CAM, these jobs are generally managed in different divisions which may communicate to each other via a network. The system developed under the Microsoft Windows environment has the characteristics of dynamic data exchange (DDE), that is, the information can be linked and updated simultaneously in several applications. This is very useful to gather data from all divisions and keep the final result updated.

The concept of concurrent or simultaneous engineering is to integrate a wide spectrum of product life-cycle concerns. The degree of integration of product design and the design of the facility and the process plan for the facility (process design) can be in terms of the degree of simultaneity [25]. The approach of this system which concerns about product design and process design in existing manufacturing facilities has achieved "minimum simultaneity". However, the poor understanding between product designer and process designer is often the main drawback of the above approach. Facility design, which concerns the machining-related factors such as machines, fixtures, cutting tools, gauging equipment and material-handling equipment in the cell, is helpful and needed to enhance the degree of simultaneity and achieve maximum benefits.

5.2.3 Expanding the Processes and Machining Parameter Databases

The machining parameters and process databases established in this system are the prototype that can solve most metal removal cases. However, expanding the work and tool material database can provide more process options. Furthermore, there are more machining processes available in SmartCAM such as punch, EDM, burner, router, and contour/laser cutting. Revision and complement of the system databases are required in dealing with more complicated processes.

5.2.4 Extension of System Scope

Quality control is another important component in product manufacturing. Part or tool inspection is performed before, during, and after manufacturing to ensure that the quality level of the product is achieved. To reduce functional variation, that is, increase the consistency of product/process performance, it is essential that we identify the basic sources of functional variation so that appropriate countermeasures can be implemented.

Measuring of the values of the product or tool characteristics and collecting those data to construct statistical control charts can significantly help quality control. It is possible to adopt automatic measuring devises such as coordinate measuring machines (CMM) to measure parts and collect data (part dimensions, deviations, and so on) automatically. Whether on-line/off-line or manual/automated, SPC techniques are useful for analyzing the results of the inspection procedure and improve the product quality.

REFERENCES

[1] Anderson, D.M., Design for Manufacturability, CIM Press, Lafayette, CA, 1990

[2] Ettlie, J.E. and Stoll, H.W., Managing the Design-Manufacturing Process, McGraw-Hill, Inc., New York, NY, 1990

[3] Pugh, S., Total Design, Addison-Wesley Publishing Co., Wokingham, England, 1990

[4] Lindberg, R.A., Processes and Materials of Manufacture, Allyn and Bacon, Needham Heights, MA, 1990

[5] DeGarmo, E.P., Black, J.T., and Kosher, R.A., Materials and Processes in Manufacturing, Macmillian Publishing Co., New York, NY, 1988

[6] Machining Data Handbook, Vol. 1 & 2, Machinability Data Center, 3rd Edition, 1980

[7] Metals Handbook, Vol. 16 -Machining, American Society For Metals, 9th Edition, 1989

[8] Schey, J.A., Introduction to Manufacturing Processes, 2nd Edition, McGraw-Hill, Inc., New York, NY, 1987

[9] Microsoft Visual Basic : Programming system for Windows, Version 3.0, Microsoft Corporation, 1993

[10] Quattro Pro for Windows Version 1.0, Borland, 1992

[11] Paradox 3.0: Introduction to Paradox, Borland International, Scotts Valley, CA, 1988

[12] SmartCAM V7.0, Point Control Co., OR, 1992.

[13] Mitchell, F.H., CIM Systems: An Introduction to Computer-Integrated Manufacturing, Prentice-Hall, 1911.

[14] Kusiak, Andrew, Intelligent Manufacturing Systems, Prentice-Hall, 1990.

[15] Kalpakjian, Serope, Manufacturing Engineering and Technology, 2nd Edition, Addison-Wesley Publishing Company, Inc., 1992.

[16] Groover, Mikell P., and Zimmers, Emory W., CAD/CAM: Computer-Aided Design and Manufacturing, Prentice-Hall, 1984.

[17] Nicks, J. E., Basic Programming Solutions for Manufacturing, 1st Edition, Society of Manufacturing Engineers, 1982.

[18] Billatos, S. B., Kendall, L. A., "A Replace Model for Multi-Tool Transfer Liner", Journal of Engineering for Industry, Vol. 112, pp253-259, August 1990.

[19] Billatos, S. B., Kendall, L. A., "A General Optimization Model for Multi-Tool Manufacturing System", Transactions of the ASME, Vol. 113, pp10-16, February 1991.

[20] Hough, C. L., Jr., "Sufficient Conditions for Cutting Rate - Tool Life Characteristic Functions for Metal Removal Processes", Journal of Engineering for Industry, Vol. 108, pp235-237, August 1986.

[21] Dewhurst, P., Boothroyd, G., "Early Cost Estimating in Product Design", Journal of Manufacturing Systems, Vol. 7/No.3, pp183-191, 1988.

[22] Kramer, B.M., "On Tool Material for High Speed Machining", Journal of Engineering for Industry, Vol. 109, pp87-91, May 1987.

[23] Jha, N. K., " A Discrete Data Base Multiple Objective Optimization of Milling Operation Through Geometric Programming", Transactions of the ASME, Vol. 112, pp368-374, November 1990.

[24] Levin, Jordan B., Dutta, Debasish, "Computer-Aided Process Planning for Parallel Machines", Journal of Manufacturing Systems, Vol. 11/No.2, pp79-92, 1992.

[25] Subramanyam, S., Lu, S. C-Y., "Computer-Aided Simultaneous Engineering for Components Manufactured in Small and Medium Lot-Sizes", Transaction of the ASME, Vol. 113, pp450-464, November 1991.

[26] Billatos, Samir B., Tseng, Pai-Chung, "Knowledge-Based Optimization for Intelligent Machining", Journal of Manufacturing Systems, Vol. 10/No. 6, 1991.

[27] Hinduja, S., Petty, D. J., Tester, M., Barrow, G., "Calculation of Optimum Cutting Conditions for Turning Operations", Proc Instn. Mech. Engr., Vol. 199, No. B2, 1985.

[28] Haddock, Jorge, Nitin Seshadri, V. R. Srivatsan, " A Decision Support System for Simulation Modeling", Journal of Manufacturing Systems, Vol. 10/ No. 6, 1991.

[29] Hsueh-Jen Chen, "Design and Implementation of a System for Integrating Material and Process Selection in Automated Manufacturing", Master's Thesis, Oregon State University, 1992.

APPENDICES

APPENDIX A. System Program Source Codes Listing

This appendix contains the listing of the system program source code. This code was written to be compiled by "Visual Basic 3.0" by MicroSoft.

CONCURSY.MAK

```
FRMMDI.FRM
C:\WINDOWS\SYSTEM\OLECLIEN.VBX
C:\WINDOWS\SYSTEM\CMDIALOG.VBX
C:\WINDOWS\SYSTEM\GRID.VBX
C:\WINDOWS\SYSTEM\MSOLE2.VBX
C:\WINDOWS\SYSTEM\THREED.VBX
CCSYS.BAS
EDITOR.BAS
DBASE.FRM
FRMPOST.FRM
FRMSETU.FRM
EDITOR.FRM
FILEOPEN.FRM
FRMCAM.FRM
FRMCOST.FRM
ProjWinSize=32,800,220,300
ProjWinShow=2
IconForm="frmPost"
Title="CONCURSY"
ExeName="CONCURSY.EXE"
```

CCSYS.BAS

```
Option Explicit
'Define global variables

Type setup
   profile As String            'Process File Name
   tmpfile As String            'SmartCAM Template File Name
   mpower  As Single            'Machine Power
   eff     As Single            'Machine Efficiency
   rapid   As Single            'Machine Rapid Traverse Speed
   maxsp   As Integer           'Machine Maximum Speed
   maxfd   As Single            'Machine Maximum Feed Rate
   maxdoc  As Single            'Machine Maximum Depth of Cut
   opcost  As Single            'Operation cost rate
   loadtime   As Single         'Load/Unload Time
   changtime   As Single        'Tool Changing Time
   reindex As Single            'Tool Reindexing Time
   setuptime As Single          'Machine Setup Time
   lotsize As Integer           'Lot size per batch
   wmatl As String              'Work Material Name
   wcode As String              'Work Material AISI Code
   whard As String              'Work Material Hardness
   tool As Integer              'Number of Tool Used in Process
   wdia As Single               'Work Material Diameter (Turning process)
End Type

Global main As Form             'Current Active Form
Global OldPath As String        'Old Path
Global FileNum As Integer       'File Number to be opened
Global turn As Integer, mill As Integer 'Process switch
Global sara As setup            'Process Setup Data (Type)
Global sdata As Variant
Global tdata As Variant         'Tool Setup Data
Global procfile As String       'Process Data File Name
Global tempfile As String       'Template File Name
Global hp As Integer            'Horse Power of Machine Tool
Global efficiency As Single     'Machine Efficiency
Global rapids As Single         'Machine Rapid Traverse Speed
Global maxsp As Integer         'Machine Maximum Speed
Global maxfd As Single          'Machine Maximum Feed Rate
Global maxdoc As Single         'Machine Maximum Depth of Cut
Global opcost As Single         'Operation cost rate
Global loadtime As Single       'Load/Unload Time
Global chngtime As Single       'Tool Changing Time
Global reindextm As Single      'Tool Reindexing Time
Global setuptm As Single        'Machine Setup Time
Global lotsize As Integer       'Lot size per batch
Global datasource As String
Global arraysize As Integer
Global combination As String
Global workmaterialname As String     'Work Material Name
Global workmaterialcode As String     'Work Material AISI Code
Global workhardness As String         'Work Material Hardness
```

```
Global workmaterialdia  As Single
Global no_of_tool As Integer
Global toolnum() As Integer
Global tooltype() As String
Global toolmaterialname() As String
Global toolmaterialcode() As String
Global toolteeth() As Integer
Global tooledge() As Integer
Global tooldia() As Single
Global toolresharpen() As Integer
Global toolcost() As Single
Global Tlife() As Single
Global tooltotalcost() As Single
Global combinationsetup() As String
Global TTX() As String
Global TTG() As String
Global oldlife() As Single
Global oldspeed() As Integer
Global oldfeed() As Single
Global olddoc() As Single
Global newdoc() As Single
Global newlife() As Single
Global newspeed() As Integer
Global newfeed() As Single
Global D() As Single
Global S() As Single
Global F() As Single
Global SFM() As Single
Global FED() As Single
Global n() As Single
Global c() As Integer
Global pu() As Single
Global Elife() As Single
Global Espeed() As Single
Global Mlife() As Single
Global Mspeed() As Single
Global hpu() As Single
Global hptest() As Single
Global testdoc As Single
Global recomdoc() As Single
Global recomspeed() As Integer
Global recomfeed() As Single
Global recomlife() As Single
Global ok As Integer
Global feedcost(), rapidcost(), tlchngcost(), tldeprcost() As Single
Global loadcost, setupcost As Single
```

```
Sub loadform (typeofshow As Integer)

   Select Case typeofshow              ' Set patterns of loading form
   Case 3 'quick show in real size
      main.Show
   Case 4 'show in screen center
      main.Top = screen.Height / 2: main.Left = 15
      main.Width = screen.Width - main.Left - 30
      main.Height = 20
      main.Show
   Case 5 'quick maximized
      main.Show
   End Select

End Sub

Sub millON ()

   turn = 0     ' Set current process to be MILL
   mill = 0
   mill = True
   turn = False

End Sub

Sub turnON ()

   turn = 0     ' Set current process to be TURN
   mill = 0
   turn = True
   mill = False

End Sub
```

<u>**EDITOR.BAS**</u> (This subprogram is used to retrieve or save process file)

```
Dim ArrayNum As Integer      ' Index value for the menu control array mnuFileArray.
Global filename As String    ' This variable keeps track of the filename information for
                             opening and closing files.
Const MB_YESNO = 4, MB_ICONQUESTION = 32, IDNO = 7, MB_DEFBUTTON2 = 256
```

```
Sub chooseprocess (filename As String)  'Retrieve Process file

   Dim F As Integer
   Dim response As Integer
   Dim getstring As String
   Dim app As String

   If filename = frmsetup.txtSetup(0).Text Then       ' Avoid opening the file if already
loaded.
     response = MsgBox("Reopen current process file ?", 1, "Warning!")
     Select Case response
          Case 2
             Exit Sub
          Case 1
          GoTo ignor
     End Select

   Else
     On Error GoTo OpenProcessErr

ignor:          F = FreeFile

   Open filename For Input As F          ' Open file selected on File Open About.
        Line Input #F, getstring
        app = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
        getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
           If CInt(app) = 1 Then
             frmsetup.optApp(0).Value = 1
           ElseIf CInt(app) = 0 Then
             frmsetup.optApp(1).Value = 1
           End If

        For i = 0 To 13
           frmsetup.txtSetup(i).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
           getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
        Next i

     frmsetup.Command1.Value = 1

        frmsetup.Combo3.Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
           getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
        frmsetup.Combo1.Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
           getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
        frmsetup.Combo2.Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
           getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
        frmsetup.tool.Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
           getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
```

```
    frmsetup.workdia.Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))

frmsetup.cmdSet.Value = 1

    ReDim TTX(arraysize + 1) As String
    ReDim TTG(arraysize + 1) As String

For j = 0 To arraysize
    frmsetup.txttoolNum(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       toolnum(j) = CInt(frmsetup.txttoolNum(j).Text)
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
    frmsetup.cmbtooltype(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       tooltype(j) = frmsetup.cmbtooltype(j).Text

    TTX(j) = frmsetup.cmbtooltype(j).Text
   If TTX(j) = "Turn" Or TTX(j) = "Face" Then
           TTG(j) = "TURN"
   ElseIf TTX(j) = "Face Grv" Or TTX(j) = "OD Grv" Or TTX(j) = "ID Grv" Then
           TTG(j) = "CUTOFF"
   ElseIf TTX(j) = "Drill" Or TTX(j) = "Tap" Or TTX(j) = "Spot Drill" Then
           TTG(j) = "DRILL"
   ElseIf TTX(j) = "Bore" Or TTX(j) = "OD Thrd" Or TTX(j) = "ID Grv" Then
           TTG(j) = "BORING"
   ElseIf TTX(j) = "Reamer" Or TTX(j) = "C'Bore" Or TTX(j) = "C'Sink" Then
           TTG(j) = "REAMING"
   ElseIf TTX(j) = "End Mill" Or TTX(j) = "Ball Mill" Or TTX(j) = "Face Mill" Or TTX(j)
   = "Draft Mill" Then
           TTG(j) = "MILL"
   End If
    frmsetup.cmbtooltype(j).Tag = TTG(j)
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
    frmsetup.cmbtoolcode(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       toolmaterialcode(j) = frmsetup.cmbtoolcode(j).Text
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
    frmsetup.txttoolmaterial(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       toolmaterialname(j) = frmsetup.txttoolmaterial(j).Text
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
    frmsetup.cmbtooldia(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       If frmsetup.cmbtooldia(j).Text = "" Then
         tooldia(j) = 1
       Else
         tooldia(j) = CSng(frmsetup.cmbtooldia(j).Text)
       End If
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
    frmsetup.txttoolteeth(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       If turn Then
          If frmsetup.txttoolteeth(j).Text = "" Then
             tooledge(j) = 1
          Else
             tooledge(j) = CInt(frmsetup.txttoolteeth(j).Text)
          End If
       ElseIf mill Then
          toolteeth(j) = CInt(frmsetup.txttoolteeth(j).Text)
```

```
            End If
            getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
         frmsetup.txtresharpen(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
            toolresharpen(j) = CInt(frmsetup.txtresharpen(j).Text)
            getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
         frmsetup.txttoolcost(j).Text = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
            toolcost(j) = CSng(frmsetup.txttoolcost(j).Text)
            getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
     Next j

         frmsetup.txttoolNum(0).SetFocus
     Close F
     Exit Sub
   End If
OpenProcessErr:
     MsgBox "Error encountered while trying to open file, please retry.", 48, "Text Editor"
     Close F
     Exit Sub

End Sub

Sub CloseFile (filename As String)

Dim F As Integer
On Error GoTo CloseError         ' If there is an error, display the error message below.

 If Dir(filename) <> "" Then          ' File already exists, so ask if overwriting is desired.
      response = MsgBox("Overwrite existing file?", MB_YESNO + MB_QUESTION +
MB_DEFBUTTON2)
      If response = IDNO Then Exit Sub
 End If
 F = FreeFile
 Open filename For Output As F       ' Otherwise, open the file name for output.
   Print #F, frmEditor!txtEdit.Text    ' Print the current text to the opened file.
 Close F                         ' Close the file
 filename = frmOpenSave!dirOpenSave.Path + "\" + "Untitled" ' Reset the caption of the
main form
Exit Sub
CloseError:
   MsgBox "Error occurred trying to close file, please retry.", 48
   Exit Sub

End Sub

Sub DoUnLoadPreCheck (unloadmode As Integer)

   If unloadmode = 0 Or unloadmode = 3 Then
        Unload frmEditor
        Unload frmOpenSave
   End If

End Sub
```

Sub OpenFile (filename As String)

```
Dim F As Integer
   If "Text Editor: " + filename = frmEditor.Caption Then  ' Avoid opening the file if already
loaded.
      Exit Sub
   Else
      On Error GoTo ErrHandler
      F = FreeFile
      Open filename For Input As F                        ' Open file selected on File Open
About.
         frmEditor!txtEdit.Text = Input$(LOF(F), F)
      Close F                              ' Close file.
      frmEditor!mnuFileItem(3).Enabled = True        ' Enable the Close menu item
      UpdateMenu
      frmEditor.Caption = "Text Editor: " + filename
      Exit Sub
   End If
ErrHandler:
      MsgBox "Error encountered while trying to open file, please retry.", 48, "Text Editor"
      Close F
      Exit Sub
```

End Sub

Sub openfiles (filename As String, contents As Control)

```
Dim F As Integer
   On Error GoTo ErrHandlers
         F = FreeFile
         Open filename For Input As F  ' Open file selected on File Open About.
         contents.Text = Input$(LOF(F), F)
         Close F                              ' Close file.
         Exit Sub
ErrHandlers:
      MsgBox "Error encountered while trying to open file, please retry.", 48, "Text Editor"
      Close F
      Exit Sub
```

End Sub

Sub SAVEFILE (filename As String, contents As Control)

```
Dim F As Integer
On Error GoTo CloseErrors              ' If there is an error, display the error message below.
   If Dir(filename) <> "" Then         ' File already exists, so ask if overwriting is desired.
      response = MsgBox("Overwrite existing file?", MB_YESNO + MB_QUESTION +
MB_DEFBUTTON2)
      If response = IDNO Then Exit Sub
   End If
   F = FreeFile
   Open filename For Output As F       ' Otherwise, open the file name for output.
      Print #F, contents.Text          ' Print the current text to the opened file.
   Close F                             ' Close the file
   Exit Sub
CloseErrors:
   MsgBox "Error occurred trying to close file, please retry.", 48
   Exit Sub
```

End Sub

Sub SaveProcess (filename As String)

```
Dim F As Integer
On Error GoTo SaveProcessError         ' If there is an error, display the error message below.
   If Dir(filename) <> "" Then         ' File already exists, so ask if overwriting is desired.
      response = MsgBox("Overwrite existing file?", MB_YESNO + MB_QUESTION +
MB_DEFBUTTON2)
      If response = IDNO Then Exit Sub
   End If
   F = FreeFile
   Open filename For Output As F       ' Otherwise, open the file name for output.
   If turn Then                  ' Print the current setup data to the opened file.
      Print #F, "1," + sdata + tdata
   ElseIf mill Then
      Print #F, "0," + sdata + tdata
   End If
   Close F                       ' Close the file
   Exit Sub
SaveProcessError:
   MsgBox "Error occurred trying to save file, please retry.", 48
   Exit Sub
```

End Sub

Sub UpdateMenu ()

```
   frmEditor!mnuFileArray(0).Visible = True   ' Make the initial element visible / display
separator bar.
   ArrayNum = ArrayNum + 1                    ' Increment index property of control array.
   Load frmEditor!mnuFileArray(ArrayNum)      ' Create a new menu control.
   frmEditor!mnuFileArray(ArrayNum).Caption = filename ' Set the caption of the new menu
item.
   frmEditor!mnuFileArray(ArrayNum).Visible = True  ' Make the new menu item visible.
```

End Sub

DBASE.FRM (This subprogram is used to search machining parameters in database)

```
Dim datasource As String

Sub blankdata ()

Dim I As Integer
For I = 0 To 22
TEXT1(I).Text = ""
Next I

End Sub

Sub cmdFind_Click ()

Dim criteria As String
criteria = "[Combination #] =" + " '" + combination + "'"
MsgBox criteria
setrecordsource
data1.Recordset.FindFirst criteria

If data1.Recordset.NoMatch Then
        MsgBox "There is no such combination in database!"
        blankdata
End If

resetrecordsource

End Sub

Sub dataoff ()

workcode.Enabled = False
hardness.Enabled = False
toolcode.Enabled = False
tooldata.Enabled = False

End Sub

Sub dataon ()

workcode.Enabled = True
hardness.Enabled = True
toolcode.Enabled = True
tooldata.Enabled = True
End Sub

Sub dbexit_Click ()

   dataoff
   Unload Me

End Sub
```

```
Sub Form_Deactivate ()

turn = 0
mill = 0
dataoff

End Sub

Sub Form_Load ()

   datasource = "data"        'Specify the database file "DATA"

End Sub

Sub hardness_Click ()

If tooldata.Text <> "" Then
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text + CStr(CInt(CSng(tooldata.Text) * 1000))
Else
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text
End If
Refresh

End Sub

Sub hardness_GotFocus ()

Dim sql As String
sql = "select distinct [Lower hardness],[Upper hardness] from " + datasource
sql = sql + " where [Workmaterial code] = '" + workcode.Text + "'"
data1.RecordSource = sql
data1.Refresh
hardness.Clear
Do While Not data1.Recordset.EOF
        hardness.AddItem CStr(data1.Recordset("Lower hardness")) + "-" +
        CStr(data1.Recordset("Upper hardness"))
        data1.Recordset.MoveNext
Loop
Refresh

End Sub

Sub resetrecordsource ()

Dim I As Integer
For I = 0 To 22
        TEXT1(I).DataField = ""
Next I

End Sub
```

```
Sub setrecordsource ()

data1.RecordSource = "select * from " + datasource
data1.Refresh
TEXT1(0).DataField = "[Combination #]"
TEXT1(1).DataField = "[Workmaterial name]"
TEXT1(2).DataField = "[Workmaterial code]"
TEXT1(3).DataField = "[Lower hardness]"
TEXT1(4).DataField = "[Upper hardness]"
TEXT1(5).DataField = "[Toolmaterial name]"
TEXT1(6).DataField = "[Toolmaterial code]"
TEXT1(7).DataField = "d1"
TEXT1(8).DataField = "s1"
TEXT1(9).DataField = "f1"
TEXT1(10).DataField = "d2"
TEXT1(11).DataField = "s2"
TEXT1(12).DataField = "f2"
TEXT1(13).DataField = "d3"
TEXT1(14).DataField = "s3"
TEXT1(15).DataField = "f3"
TEXT1(16).DataField = "d4"
TEXT1(17).DataField = "s4"
TEXT1(18).DataField = "f4"
TEXT1(19).DataField = "tool data"
TEXT1(20).DataField = "N"
TEXT1(21).DataField = "C"
TEXT1(22).DataField = "PU"

End Sub

Sub toolcode_Click ()

If tooldata.Text <> "" Then
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text + CStr(CInt(CSng(tooldata.Text) * 1000))
Else
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text
End If
Refresh

End Sub
```

```
Sub toolcode_GotFocus ()

Dim sql As String
sql = "select distinct [toolmaterial CODE] from " + datasource
sql = sql + " where [TOOL TYPE]= '" + tooltype.Text + "'"
data1.RecordSource = sql
data1.Refresh
toolcode.Clear
Do While Not data1.Recordset.EOF
    toolcode.AddItem data1.Recordset("toolmaterial CODE")
    data1.Recordset.MoveNext
Loop
Refresh

End Sub

Sub tooldata_Click ()
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text + CStr(CInt(CSng(tooldata.Text) * 1000))
Refresh

End Sub

Sub tooldata_GotFocus ()

Dim sql As String
sql = "select distinct [tool data] from " + datasource
sql = sql + " where [TOOL TYPE]= '" + tooltype.Text + "'"
data1.RecordSource = sql
data1.Refresh
tooldata.Clear
Do While Not data1.Recordset.EOF
    tooldata.AddItem data1.Recordset("tool data")
    data1.Recordset.MoveNext
Loop
Refresh

End Sub

Sub tooltype_Click ()

If tooldata.Text <> "" Then
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text + CStr(CInt(CSng(tooldata.Text) * 1000))
Else
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text
End If
Refresh
dataon

End Sub
```

```
Sub tooltype_GotFocus ()

Dim sql As String
sql = "select distinct [tool type] from " + datasource
data1.RecordSource = sql
data1.Refresh
tooltype.Clear
Do While Not data1.Recordset.EOF
    tooltype.AddItem data1.Recordset("tool type")
    data1.Recordset.MoveNext
Loop
Refresh

End Sub

Sub workcode_Click ()

If tooldata.Text <> "" Then
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text + CStr(CInt(CSng(tooldata.Text) * 1000))
Else
        combination = Left$(tooltype.Text, 1) + workcode.Text + Right$(hardness.Text,
        3) + toolcode.Text
End If
Refresh

End Sub

Sub workcode_GotFocus ()

Dim sql As String
sql = "select distinct [workmaterial code] from " + datasource
sql = sql + " where [tool type] = '" + tooltype.Text + "'"
data1.RecordSource = sql
data1.Refresh
workcode.Clear
Do While Not data1.Recordset.EOF
    workcode.AddItem data1.Recordset("Workmaterial code")
    data1.Recordset.MoveNext
Loop
Refresh

End Sub
```

FILEOPEN.FRM (This subprogram defines the file format to be accessed)

```
Option Explicit

Sub cmdOpenSave_Click (Index As Integer)

   If Index = 0 Then            ' OK button pressed.
      On Error Resume Next
      filOpenSave.FileName = txtOpenSave.Text
      If Err Then               ' Something wrong with supplied path or filename.
         Err = 0
         If Dir(txtOpenSave.Text) = "" Then
            If Err Then ' Bad path or filename,
               MsgBox Error, 48        ' so report that error.
            ElseIf Caption = "Save Process as" Then
               SaveProcess (txtOpenSave.Text)
               Unload frmOpenSave
            ElseIf Caption = "Open Process " Then
               chooseprocess (txtOpenSave.Text)
               Unload frmOpenSave
            Else               ' File doesn't exist, and opening file, so report error.
               MsgBox "File doesn't exist.", 48
            End If
            Exit Sub
         End If
      End If
   Else
      Unload frmOpenSave       ' If the OK button wasn't pressed then
                               ' Cancel must have been so unload the dialog.
   End If

End Sub

Sub dirOpenSave_Change ()

filOpenSave.Path = dirOpenSave.Path  ' When the directory changes, update the file path
lblOpenSave(2).Caption = dirOpenSave.Path ' and the path displayed.
ChDir dirOpenSave.Path                   ' Set the current working directory to the new path.

End Sub

Sub drvOpenSave_Change ()
dirOpenSave.Path = drvOpenSave.Drive ' When drive changes, update path of directory box
ChDrive drvOpenSave.Drive            ' and set the current working drive to the new drive.
End Sub

Sub filOpenSave_Click ()

' Display the file selected in the text box up in the File Name text box.
   txtOpenSave.Text = filOpenSave.List(filOpenSave.ListIndex)

End Sub
```

```
Sub filOpenSave_DblClick ()

' When at root of drive, there is no "\" at the end of the path, so add it.
   If Right(filOpenSave.Path, 1) = "\" Then
      FileName = filOpenSave.Path & filOpenSave.FileName
   Else
      FileName = filOpenSave.Path & "\" & filOpenSave.FileName
   End If

   Select Case frmOpenSave.Caption      ' Check to see if "Open" or "Save" dialog.
      Case "Open Process"
         chooseprocess (FileName)
      Case "Save Process as"
         SaveProcess (FileName)
   End Select
   Unload frmOpenSave

End Sub

Sub filOpenSave_PathChange ()

' Notice that this doesn't cause a cascading event.
   txtOpenSave.Text = filOpenSave.Pattern
   drvOpenSave.Drive = filOpenSave.Path
   dirOpenSave.Path = filOpenSave.Path

End Sub

Sub filOpenSave_PatternChange ()

' Notice this doesn't cause a cascading event.
   drvOpenSave.Drive = filOpenSave.Path
   dirOpenSave.Path = filOpenSave.Path

End Sub

Sub Form_Load ()

' The form is horizontally and vertically centered when loaded.
  frmOpenSave.Left = Screen.Width / 2 - frmOpenSave.Width / 2
  frmOpenSave.Top = Screen.Height / 2 - frmOpenSave.Height / 2
' Set current drive and directory.
   drvOpenSave.Drive = CurDir
   dirOpenSave.Path = CurDir
' Display the current path.
lblOpenSave(2).Caption = dirOpenSave.Path

End Sub
```

FRMCAM.FRM (SmartCAM menu)

```
Dim oldpath

Sub camAcce_Click (index As Integer)

   oldpath = CurDir
   ChDir "c:\sm6"
      Select Case index
      Case 0
         X = Shell("machdef.exe", 1)
      Case 1
         X = Shell("camdxf2d.exe", 1)
      Case 2
         X = Shell("camigs2d.exe", 1)
      End Select
      ChDir oldpath

End Sub

Sub camApp_Click (index As Integer)

   oldpath = CurDir
   ChDir "c:\sm6"
      Select Case index
      Case 0
         turnON
         X = Shell("turn.bat", 1)
         frmCam.WindowState = 2
      Case 1
         millON
         X = Shell("mill.bat", 1)
         frmCam.WindowState = 2
      End Select
   ChDir oldpath

End Sub

Sub camComm_Click (index As Integer)

   Dim oldpath
   On Error Resume Next
   oldpath = CurDir
   ChDir "c:\sm6"
Select Case index
Case 0
      Open curpath + "\communic.bat" For Output As #1
      Print #1, "copy comm.exe comme.exe"
      Print #1, "comme.exe"
      Print #1, "del comme.exe"
      Print #1, "exit"
      Close #1
      X = Shell("communic.bat", 1)
      Kill curpath + "communic.bat"
```

```
Case 1
      X = Shell("camcont.bat", 1)
End Select
ChDir oldpath

End Sub

Sub camItem_Click (index As Integer)

   oldpath = CurDir
   ChDir "c:\sm6"
      Select Case index
      Case 0
         X = Shell("edplus.exe", 1)
      Case 1
         X = Shell("jplan.exe", 1)
      Case 5
         Unload Me
      End Select
   ChDir oldpath

End Sub

Sub Form_Resize ()

If main.WindowState = 2 Then
        main.Caption = "CAD/CAM <SmartCAM>"
Else
        main.Caption = "SmartCAM"
End If

End Sub
```

FRMCOST.FRM (Cost evaluation module)

```
Dim Evatag As String
Dim tlnum(), tlair(), tlcut(), tltm(), Cytm(), tllf(), tledge(), tlresharpen(), tlcost(), tldesc()

Sub calcost (lnth As Integer)

ReDim feedcost(lnth), rapidcost(lnth), tlchngcost(lnth), tldeprcost(lnth) As Single
For i = 0 To lnth - 1
   feedcost(i) = opcost / 60 * tlcut(i)
   rapidcost(i) = opcost / 60 * tlair(i)
   tlchngcost(i) = opcost / 60 * chngtime * tlcut(i) / tllf(i)
   If turn Then
        tldeprcost(i) = tlcut(i) / tllf(i) * tlcost(i) / tledge(i) * tlresharpen(i)
   ElseIf mill Then
        tldeprcost(i) = tlcut(i) / tllf(i) * tlcost(i) / tlresharpen(i)
   End If
Next i
loadcost = opcost / 60 * loadtime
setupcost = opcost / 60 * setuptm / lotsize

End Sub

Sub cmdEva_Click (indexEva As Integer)

Select Case indexEva

Case 0          'file search
      If Evatag = "sm6" Then
         diaeva.FilterIndex = 1
         diaeva.Filename = "*.*"
         diaeva.Action = 1
         txtEvafile(0).Text = diaeva.Filename
      ElseIf Evatag = "prc" Then
         diaeva.FilterIndex = 2
         diaeva.Filename = "*.prc"
         diaeva.Action = 1
         txtEvafile(1).Text = diaeva.Filename
      ElseIf Evatag = "evl" Then
         diaeva.FilterIndex = 3
         diaeva.Filename = "*.evl "
         diaeva.Action = 1
         txtEvafile(2).Text = diaeva.Filename
      ElseIf Evatag = "" Then
      End If

Case 1          'Reset
      For i = 0 To 2
         txtEvafile(i).Text = ""
      Next i

Case 2          'OK
   evaFilerec(0).Visible = True
   For i = 1 To 3
```

```
        evaFilerec(i).Caption = txtEvafile(i - 1).Text ' Set the caption of the new menu item.
        evaFilerec(i).Visible = True
    Next i
    evafile(1).Enabled = 1
    evafile(2).Enabled = 1
    Evaitem(1).Enabled = 1
    'fraCost.Visible = 0
End Select
```

End Sub

Sub Evafile_Click (indexevafile As Integer)

```
Select Case indexevafile

Case 0          'Read
    diaeva.DialogTitle = "Read Data file"
    txtEvafile(0).SetFocus
Case 1          'Open
    diaeva.DialogTitle = "Open Text file"
    diaeva.Action = 1
    Filename = diaeva.Filename
    openfiles Filename, frmcost!txtCost
    evafile(2).Enabled = 1
    evafile(3).Enabled = 1
Case 2          'Save
    diaeva.DialogTitle = "Save text file as"
    diaeva.Action = 1
    Filename = diaeva.Filename
    SAVEFILE Filename, frmcost!txtCost
Case 3          'Print
    printer.Print txtCost.Text
Case 4          'dash

Case 5          'Leave
    Dim msg, response, title
    msg = "Are you sure to exit from this Evaluation?"
    title = "Caution!"
    response = MsgBox(msg, 52, title)
    If response = 6 Then
        Unload Me
    ElseIf response = 7 Then
    End If
End Select
```

End Sub

Sub Evaitem_Click (Indexevaitem As Integer)

```
Dim f As Integer
Dim lnth, blnth, alnth As Integer
Dim getstring As String
Dim cytime As Single
Select Case Indexevaitem
```

```
    Case 0      'File
    Case 1      'Evaluation
       f = FreeFile
       Open evaFilerec(1).Caption For Input As f      ' Open file selected on File Open
About.
         m = 0
       Do While Not EOF(f)
         Line Input #f, getstring
         m = m + 1
       Loop
       Close f
       lnth = m
       ReDim tlnum(lnth), tlair(lnth), tlcut(lnth), tltm(lnth), Cytm(lnth)
       ReDim tllf(lnth), tledge(lnth), tlresharpen(lnth), tlcost(lnth), tldesc(lnth)

       f = FreeFile
       Open evaFilerec(1).Caption For Input As f      ' Open file selected on File Open
About.
       For i = 0 To lnth - 1
         Line Input #f, getstring
       tlnum(i) = CInt(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tlair(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tlcut(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tltm(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       Cytm(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tllf(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tledge(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))       'or teeth for
milling
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tlresharpen(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tlcost(i) = CDbl(Mid$(getstring, 1, InStr(1, getstring, ",") - 1))
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       tldesc(i) = Mid$(getstring, 1, InStr(1, getstring, ",") - 1)
       getstring = Mid$(getstring, InStr(1, getstring, ",") + 1, Len(getstring))
       Next i
       Close f

       cytime = Cytm(lnth - 1)
       blnth = lnth
       f = FreeFile
       Open evaFilerec(3).Caption For Output As f
       Print #f, "==========Original SmartCAM Result==========="
       Print #f, ""
       Print #f, "     Air-   Cut-  Reindex Tool   Tool  Tool_change "
       Print #f, "Tool# Time   Time   Time   Time   Life  frequency   Tool Description"
       Print #f, "     (min)  (min)  (min)  (min)  (min)   (part) "
       Print #f, "_____ _____ _____ _____ _____ ____ _____ _________________"
```

```
        For i = 0 To lnth - 1
        Print #f," " + CStr(tlnum(i)); Tab(8); CStr(Format(tlair(i), "0.00")); Tab(16);
CStr(Format(tlcut(i), "0.00")); Tab(24); CStr(Format(tltm(i) - tlair(i) - tlcut(i), "0.00"));
Tab(32); CStr(Format(tltm(i), "0.00")); Tab(39); CStr(Format(tllf(i), "0.0")); Tab(48);
CStr(Format(tllf(i) / tlcut(i), "0")); Tab(59); tldesc(i)
        tlair(i) = tlair(i) + tltm(i) - tlair(i) - tlcut(i)
        Next i
        Print #f, ""
        Print #f, "The estimated Cycle time is : " + CStr(Format(cytime, "0.00")) + " min"
        Print #f,"======================================"

        'Rearrange original data
        For i = lnth - 2 To 0 Step -1
           For j = i + 1 To lnth - 1 Step 1
              If tlnum(i) = tlnum(j) Then
                tlair(i) = tlair(i) + tlair(j)
                tlcut(i) = tlcut(i) + tlcut(j)
                tltm(i) = tltm(i) + tltm(j)
                tllf(i) = tllf(i)
                tledge(i) = tledge(i)            'or teeth for milling
                tlresharpen(i) = tlresharpen(i)
                tlcost(i) = tlcost(i)
                tldesc(i) = tldesc(i)
              For k = j + 1 To lnth - 1 Step 1
                tlnum(k - 1) = tlnum(k)
                tlair(k - 1) = tlair(k)
                tlcut(k - 1) = tlcut(k)
                tltm(k - 1) = tltm(k)
                tllf(k - 1) = tllf(k)
                tledge(k - 1) = tledge(k)                'or teeth for milling
                tlresharpen(k - 1) = tlresharpen(k)
                tlcost(k - 1) = tlcost(k)
                tldesc(k - 1) = tldesc(k)
              Next k
              lnth = lnth - 1
       ReDim Preserve tlnum(lnth), tlair(lnth), tlcut(lnth), tltm(lnth), Cytm(lnth), tllf(lnth),
       tledge(lnth), tlresharpen(lnth), tlcost(lnth), tldesc(lnth)
              Else
                tlair(i) = tlair(i)
                tlcut(i) = tlcut(i)
                tltm(i) = tltm(i)
                tllf(i) = tllf(i)
                tledge(i) = tledge(i)            'or teeth for milling
                tlresharpen(i) = tlresharpen(i)
                tlcost(i) = tlcost(i)
                tldesc(i) = tldesc(i)

              End If
           Next j
        Next i
        alnth = lnth
    maxid = 0
    chooseprocess (evaFilerec(2).Caption)
    frmcost.SetFocus
```

```
calcost (lnth)
    Print #f, "Process File : " + procfile
    Print #f, "Template File: " + tempfile
    Print #f, "Machine Power: " + CStr(HP) + " HP"
    Print #f, "Machine Efficiency: " + CStr(efficiency)
    Print #f, "Rapid Traverse Feed: " + CStr(rapids) + " ipm"
    Print #f, "Maximum Speed: " + CStr(maxsp) + " rpm"
    Print #f, "Maximum Feed: " + CStr(maxfd) + " ipm"
    Print #f, "Maximum Depth of Cut: " + CStr(maxdoc) + " inch"
    Print #f, "Operation Cost: $" + CStr(opcost) + "/hr"
    Print #f, "Load/Unload Time: " + CStr(loadtime) + " min"
    Print #f, "Tool changing Time: " + CStr(chngtime) + " min"
    Print #f, "Tool reindex  Time: " + CStr(reindextm) + " min"
    Print #f, "Machine setup Time: " + CStr(setuptm) + " min"
    Print #f, "Lot Size per batch: " + CStr(lotsize) + " piece"
    Print #f, "Work Material Name: " & workmaterialname & " ;  Code: " &
workmaterialcode
    Print #f, "Work Material Hardness: " & workhardness & " BHN"
  If turn Then
    Print #f, "Workpiece original diameter: " & CStr(workmaterialdia) + """"
  Else
  End If
    Print #f, "Number of tools used in process: " & CStr(no_of_tool)
    Print #f, ""
    Print #f, "============Tool Information=============="
  If mill Then
    Print #f, " Tool   Tool      Tool     Code  Diameter  Teeth  Resharpen  Cost"
    Print #f, "Number  Type     Material  (AISI)  (inch)   (#)   (times)    ($)"
    Print #f, "______ _____ _______ ______ _______ _____ ________ ____"
   For i = 0 To arraysize
    Print #f, Spc(2); CStr(toolnum(i)); Tab(9); tooltype(i); Tab(20); toolmaterialname(i);
Tab(36); toolmaterialcode(i); Tab(43); CStr(tooldia(i)); Tab(54); CStr(toolteeth(i)); Tab(62);
CStr(toolresharpen(i)); Tab(71); CStr(toolcost(i))
   Next i
  End If

  If turn Then
    Print #f, " Tool   Tool      Tool     Code  Diameter  Edges  Resharpen  Cost"
    Print #f, "Number  Type     Material  (AISI)  (inch)   (#)   (times)    ($)"
    Print #f, "______ _______ _______ ______ _______ _____ ________ ____"
   For i = 0 To arraysize
    Print #f, Spc(2); CStr(toolnum(i)); Tab(9); tooltype(i); Tab(20); toolmaterialname(i);
Tab(36); toolmaterialcode(i); Tab(43); CStr(tooldia(i)); Tab(54); CStr(tooledge(i)); Tab(62);
CStr(toolresharpen(i)); Tab(71); CStr(toolcost(i))
   Next i
  End If
    Print #f, "===================================="
    Print #f, ""
    Print #f, "========= Revised Simulation Result ============"
    Print #f, ""
    Print #f, "      Air-   Cut-   Tool    Tool   Tool_change"
    Print #f, "Tool#  Time   Time   Time    Life   frequency   Tool Description"
    Print #f, "      (min)  (min)  (min)   (min)   (part)"
    Print #f, "____ _____ _____ _____ _____ _________ _______________"
```

```
        For i = 0 To lnth - 1
        Print #f, "  " + CStr(tlnum(i)); Tab(9); CStr(Format(tlair(i), "0.00")); Tab(18);
CStr(Format(tlcut(i), "0.00")); Tab(27); CStr(Format(tltm(i), "0.00")); Tab(35);
CStr(Format(tllf(i), "0.0")); Tab(47); CStr(Format(tllf(i) / tlcut(i), "0")); Tab(58); tldesc(i)
        Next i
        Print #f, ""
        Print #f, "The estimated Cycle time is : " + CStr(Format(cytime, "0.00")) + " min"
        Print #f, "===================================="
        Print #f, ""
        If turn Then
        Print #f, "===========Evaluation Report for Turning========="
        ElseIf mill Then
        Print #f, "===========Evaluation Report for Milling=========="
        End If
        Print #f, "Tool#    Rapid Trav   Feed    Toolchange  Tool Depr    Sub Total "
        Print #f, "          Cost($)     Cost($)   Cost($)     Cost($)      Cost($)"
        Print #f, "_____   __________  _______  _________  ________  ___________"
        ReDim subtotalV(5)
        ReDim subtotalH(lnth)
        For i = 0 To lnth - 1
        subtotalH(i) = rapidcost(i) + feedcost(i) + tlchngcost(i) + tldeprcost(i)
        Print #f, "  " + CStr(tlnum(i)); Tab(14); CStr(Format(rapidcost(i), "0.0000"));
Tab(26); CStr(Format(feedcost(i), "0.0000")); Tab(37); CStr(Format(tlchngcost(i),
"0.0000")); Tab(49); CStr(Format(tldeprcost(i), "0.0000")); Tab(62);
CStr(Format(subtotalH(i), "0.0000"))
            subtotalV(0) = subtotalV(0) + rapidcost(i)
            subtotalV(1) = subtotalV(1) + feedcost(i)
            subtotalV(2) = subtotalV(2) + tlchngcost(i)
            subtotalV(3) = subtotalV(3) + tldeprcost(i)
            subtotalV(4) = subtotalV(4) + subtotalH(i)
        Next i
        Print #f, "______________________________________________________________"
        Print #f, "Sub Total:"; Tab(14); CStr(Format(subtotalV(0), "0.0000")); Tab(26);
CStr(Format(subtotalV(1), "0.0000")); Tab(37); CStr(Format(subtotalV(2), "0.0000"));
Tab(49); CStr(Format(subtotalV(3), "0.0000")); Tab(62); CStr(Format(subtotalV(4),
"0.0000"))

        Print #f, "The Load/Unload cost :" + CStr(Format(loadcost, "0.0000")) + " $/pc"
        Print #f, "The Setup cost      :" + CStr(Format(setupcost, "0.0000")) + " $/pc"
        Print #f, "__________________________________________________"
        Print #f, "The Total Processing cost per piece : " + CStr(Format(loadcost +
setupcost + subtotalV(4), "0.0000")) + " $/pc"
        Print #f, "===================================="
        Close f                                ' Close file.
        f = FreeFile
        Open evaFilerec(3).Caption For Input As f
        txtCost.Text = Input$(LOF(f), f)
        Close f                                ' Close file.
    Case 2
        Y = Shell("calc.exe", 1)
End Select

End Sub
```

```
Sub Form_MouseDown (button As Integer, Shift As Integer, X As Single, Y As Single)

If button = 2 Then
z = Shell("calc.exe", 1)
End If

End Sub

Sub Form_QueryUnload (cancel As Integer, unloadmode As Integer)

If unloadmode = 0 Then
      Dim msg, response, title
      msg = "Are you sure to exit from this Evaluation?"
      title = "Caution!"
      response = MsgBox(msg, 52, title)
      If response = 6 Then
        Unload Me
      ElseIf response = 7 Then
      cancel = 1
      End If
End If

End Sub

Sub Form_Resize ()

On Error Resume Next
txtCost.Width = main.ScaleWidth
txtCost.Height = main.ScaleHeight - fraCost.Height
txtCost.Move 0, 0
fraCost.Move 0, txtCost.Height, txtCost.Width

End Sub

Sub txtCost_MouseDown (button As Integer, Shift As Integer, X As Single, Y As Single)

If button = 2 Then
z = Shell("calc.exe", 1)
End If

End Sub

Sub txtEvafile_GotFocus (indexevafile As Integer)
Select Case indexevafile
     Case 0
         Evatag = "sm6"
     Case 1
         Evatag = "prc"
     Case 2
         Evatag = "evl"
End Select
cmdeva(0).Value = 1
End Sub
```

<u>FRMMDI.FRM</u> (System Main Menu)

```
Sub Ed_Click (Indexed As Integer)

Select Case Indexed
      Case 1
       v = Shell("notepad.exe", 3)
End Select

End Sub

Sub environ_Click (Index As Integer)

Select Case Index
Case 0
x = Shell("printman.exe", 1)
Case 1
y = Shell("control.exe", 1)
End Select

End Sub

Sub file_Click (Index As Integer)

Select Case Index
    Case 0 'environment setup
    Case 1
      y = Shell("dosprmpt.pif", 1)
    Case 2
      y = Shell("winfile.exe", 1)
    Case 3
      Dim msg, response, title
      msg = "Are you sure to exit from this system?"

      title = "Caution!"
      response = MsgBox(msg, 52, title)
      If response = 6 Then
         End
      ElseIf response = 7 Then
      End If
End Select

End Sub

Sub main3_Click (Index As Integer)

   Select Case Index
   Case 0
   Case 1
   Case 2
      Set main = frmSetup
      loadform (5)
   Case 3
      Set main = frmCam
```

```
      loadform (5)
   Case 4
      Set main = frmCost
      loadform (5)
   Case 5

   Case 6
   End Select

End Sub

Sub MDIForm_Load ()

left = 0
top = 0
Height = screen.Height
Width = screen.Width
frmPost.Show

End Sub

Sub MDIForm_QueryUnload (cancel As Integer, unloadmode As Integer)

If unloadmode = 0 Then
      Dim msg, response, title
      msg = "Are you sure to exit from this system?"

      title = "Caution!"
      response = MsgBox(msg, 52, title)
      If response = 6 Then
         End
      ElseIf response = 7 Then
      cancel = 1
      End If
End If

End Sub
```

FRMSETUP.FRM (Process Set-up)

```
Dim maxid As Integer
Dim recomfile As String

Sub cmbDOC_Click (index As Integer)

If cmbDOC(index).Text = CStr(D(index, 0)) Then
  txtspeed(index).Text = CStr(S(index, 0))
  txtfeed(index).Text = CStr(F(index, 0))
  txtlife(index).Text = CStr(tlife(index, 0))
ElseIf cmbDOC(index).Text = CStr(D(index, 1)) Then
  txtspeed(index).Text = CStr(S(index, 1))
  txtfeed(index).Text = CStr(F(index, 1))
  txtlife(index).Text = CStr(tlife(index, 1))
ElseIf cmbDOC(index).Text = CStr(D(index, 2)) Then
  txtspeed(index).Text = CStr(S(index, 2))
  txtfeed(index).Text = CStr(F(index, 2))
  txtlife(index).Text = CStr(tlife(index, 2))
ElseIf cmbDOC(index).Text = CStr(D(index, 3)) Then
  txtspeed(index).Text = CStr(S(index, 3))
  txtfeed(index).Text = CStr(F(index, 3))
  txtlife(index).Text = CStr(tlife(index, 3))
End If

End Sub

Sub cmbDOC_GotFocus (index As Integer)

cmbDOC(index).Clear
cmbDOC(index).AddItem CStr(D(index, 0))
cmbDOC(index).AddItem CStr(D(index, 1))
cmbDOC(index).AddItem CStr(D(index, 2))
cmbDOC(index).AddItem CStr(D(index, 3))

 ok = 0

'Machining conditions selection criterion

 For j = 0 To 3
   If hpu(index, j) <> 0 And hpu(index, j) <= HP Then
       If      S(index, j) <> 0 And espeed(index) < S(index, j) And (S(index, j) <
               mspeed(index) Or S(index, j) > mspeed(index)) Then
           ok = ok + 1
           ReDim recomdoc(index, ok) As Single
           ReDim recomspeed(index, ok) As Integer
           ReDim recomfeed(index, ok) As Single

           recomdoc(index, j) = D(index, j)
           recomspeed(index, j) = S(index, j)
           recomfeed(index, j) = F(index, j)
           cmbDOC(index).Text = CStr(recomdoc(index, j))
           txtspeed(index).Text = CStr(recomspeed(index, j))
           txtfeed(index).Text = CStr(recomfeed(index, j))
```

```
        End If
    End If
  Next j

frapara.Refresh

End Sub

Sub cmbtoolcode_GotFocus (index As Integer)

Dim sql As String
sql ="select distinct [toolmaterial code] ,[toolmaterial name] from " + datasource
sql = sql + " where [combination #] LIKE '" + Left$(cmbtooltype(index).Tag, 1) +
combo1.Text + Right$(combo2.Text, 3) + "*'"

datasetup.RecordSource = sql

datasetup.Refresh
cmbtoolcode(index).Clear

Do While Not datasetup.Recordset.EOF
    cmbtoolcode(index).AddItem datasetup.Recordset("toolmaterial code")
    datasetup.Recordset.MoveNext
Loop
frmtool.Refresh

End Sub

Sub cmbtooldia_GotFocus (index As Integer)

   Dim sql As String
      sql = "select distinct [tool data] from " + datasource
      sql = sql + " where [tool type]= '" + cmbtooltype(index).Tag + "'"
   datasetup.RecordSource = sql
   datasetup.Refresh
   cmbtooldia(index).Clear
      Do While Not datasetup.Recordset.EOF
         cmbtooldia(index).AddItem datasetup.Recordset("tool data")
         datasetup.Recordset.MoveNext
      Loop
   frmtool.Refresh

End Sub
```

```
Sub cmbtooltype_Click (index As Integer)

TTX(index) = cmbtooltype(index).Text

   'Turning process      Milling process

   '"Face"        "Spot Drill"
   '"Turn"        "Drill"
   '"Bore"        "Reamer"
   '"Face Grv"    "Tap"
   '"OD Grv"      "End Mill"
   '"ID Grv"      "Ball Mill"
   '"OD Thrd"     "Face Mill"
   '"ID Thrd"     "Draft Mill"
   '"Drill"       "C'Bore"
   '"Tap"         "C'Sink"

If TTX(index) = "Turn" Or TTX(index) = "Face" Then
   TTG(index) = "TURN"
ElseIf TTX(index) = "Face Grv" Or TTX(index) = "OD Grv" Or TTX(index) = "ID Grv" Then
   TTG(index) = "CUTOFF"
ElseIf TTX(index) = "Drill" Or TTX(index) = "Tap" Or TTX(index) = "Spot Drill" Then
   TTG(index) = "DRILL"
ElseIf TTX(index) = "Bore" Or TTX(index) = "OD Thrd" Or TTX(index) = "ID Thrd" Then
   TTG(index) = "BORING"
ElseIf TTX(index) = "Reamer" Or TTX(index) = "C'Bore" Or TTX(index) = "C'Sink" Then
   TTG(index) = "REAMING"
ElseIf TTX(index) = "End Mill" Or TTX(index) = "Ball Mill" Or TTX(index) = "Face Mill" Or
TTX(index) = "Draft Mill" Then
   TTG(index) = "MILL"

frmtool.Refresh
End If
cmbtooltype(index).Tag = TTG(index)
frmtool.Refresh

End Sub

Sub cmbtooltype_GotFocus (index As Integer)

cmbtooltype(index).Clear
If mill Then
  cmbtooltype(index).AddItem "Spot Drill"
  cmbtooltype(index).AddItem "Drill"
  cmbtooltype(index).AddItem "Reamer"
  cmbtooltype(index).AddItem "Tap"
  cmbtooltype(index).AddItem "End Mill"
  cmbtooltype(index).AddItem "Ball Mill"
  cmbtooltype(index).AddItem "Face Mill"
  cmbtooltype(index).AddItem "Draft Mill"
  cmbtooltype(index).AddItem "C'Bore"
  cmbtooltype(index).AddItem "C'Sink"

ElseIf turn Then
```

```
    cmbtooltype(index).AddItem "Face"
    cmbtooltype(index).AddItem "Turn"
    cmbtooltype(index).AddItem "Bore"
    cmbtooltype(index).AddItem "Face Grv"
    cmbtooltype(index).AddItem "OD Grv"
    cmbtooltype(index).AddItem "ID Grv"
    cmbtooltype(index).AddItem "OD Thrd"
    cmbtooltype(index).AddItem "ID Thrd"
    cmbtooltype(index).AddItem "Drill"
    cmbtooltype(index).AddItem "Tap"
End If
frmtool.Refresh

End Sub

Sub cmdSet_Click ()

     workmaterialname = combo3.Text
     workhardness = combo2.Text
     workmaterialcode = combo1.Text
     no_of_tool = CInt(tool.Text)
    If turn Then workmaterialdia = CSng(workdia.Text)

 'Initialize the number of tools
    If tool.Text = "" Or tool.Text = "0" Then
       arraysize = 1
    ElseIf CInt(tool.Text) > 13 Then
       MsgBox " Sorry, you can only use at most 13 tools!"
       tool.Text = "13"
       arraysize = 12
    Else
       arraysize = CInt(tool.Text) - 1
    End If

 'Set data variables array size
    redimension

 'Initial the controls for tools

 If maxid = 0 Then
       maxid = 0
    Else
    Do Until maxid <= 0
      Unload lblnum(maxid)
      Unload txttoolNum(maxid)
      Unload cmbtooltype(maxid)
      Unload txttoolmaterial(maxid)
      Unload cmbtoolcode(maxid)
      Unload cmbtooldia(maxid)
      Unload txttoolteeth(maxid)
      Unload txtresharpen(maxid)
      Unload txttoolcost(maxid)
      Unload cmbDOC(maxid)
      Unload txtspeed(maxid)
```

```
        Unload txtfeed(maxid)
        Unload txtlife(maxid)
        maxid = maxid - 1
      Loop
    End If

    'Create the controls for tools

    If tool.Text <> "" Then
    Do Until maxid = arraysize
     maxid = maxid + 1
     Load lblnum(maxid): lblnum(maxid).Top = lblnum(maxid - 1).Top + 350
          lblnum(maxid).Visible = True: lblnum(maxid).Caption = maxid + 1
     Load txttoolNum(maxid): txttoolNum(maxid).Top = txttoolNum(maxid - 1).Top + 350
          txttoolNum(maxid).Visible = True
     Load cmbtooltype(maxid): cmbtooltype(maxid).Top = cmbtooltype(maxid - 1).Top + 350
            cmbtooltype(maxid).Visible = True
     Load txttoolmaterial(maxid): txttoolmaterial(maxid).Top = txttoolmaterial(maxid - 1).Top +
350
         txttoolmaterial(maxid).Visible = True
     Load cmbtoolcode(maxid): cmbtoolcode(maxid).Top = cmbtoolcode(maxid - 1).Top + 350
         cmbtoolcode(maxid).Visible = True
     Load cmbtooldia(maxid): cmbtooldia(maxid).Top = cmbtooldia(maxid - 1).Top + 350
         cmbtooldia(maxid).Visible = True
     Load txttoolteeth(maxid): txttoolteeth(maxid).Top = txttoolteeth(maxid - 1).Top + 350
         txttoolteeth(maxid).Visible = True
     Load txtresharpen(maxid): txtresharpen(maxid).Top = txtresharpen(maxid - 1).Top + 350
         txtresharpen(maxid).Visible = True
     Load txttoolcost(maxid): txttoolcost(maxid).Top = txttoolcost(maxid - 1).Top + 350
         txttoolcost(maxid).Visible = True
     Load cmbDOC(maxid): cmbDOC(maxid).Top = cmbDOC(maxid - 1).Top + 350
         cmbDOC(maxid).Visible = True
     Load txtspeed(maxid): txtspeed(maxid).Top = txtspeed(maxid - 1).Top + 350
         txtspeed(maxid).Visible = True
     Load txtfeed(maxid): txtfeed(maxid).Top = txtfeed(maxid - 1).Top + 350
         txtfeed(maxid).Visible = True
     Load txtlife(maxid): txtlife(maxid).Top = txtlife(maxid - 1).Top + 350
         txtlife(maxid).Visible = True
      Loop
    End If

If turn And combo3.Text <> "" And combo1.Text <> "" And combo2.Text <> "" And
tool.Text <> "" And workdia.Text <> "" Then
frmtool.Enabled = True
ElseIf mill And combo3.Text <> "" And combo1.Text <> "" And combo2.Text <> ""
And tool.Text <> "" Then

frmtool.Enabled = True

End If

End Sub
```

Sub Combo1_GotFocus ()

```
    Dim sql As String
        sql = "select distinct [workmaterial code] from " + datasource
        sql = sql + " where [Workmaterial name] = '" + combo3.Text + "'"
    datasetup.RecordSource = sql
    datasetup.Refresh
    combo1.Clear
        Do While Not datasetup.Recordset.EOF
            combo1.AddItem datasetup.Recordset("Workmaterial code")
            datasetup.Recordset.MoveNext
        Loop
    frame3d1.Refresh
```

End Sub

Sub Combo2_GotFocus ()

```
    Dim sql As String
        sql = "select distinct [Lower hardness],[Upper hardness] from " + datasource
        sql = sql + " where [Workmaterial code] = '" + combo1.Text + "'"
    datasetup.RecordSource = sql
    datasetup.Refresh
    combo2.Clear
        Do While Not datasetup.Recordset.EOF
          combo2.AddItem CStr(datasetup.Recordset("Lower hardness")) + "-" +
          CStr(datasetup.Recordset("Upper hardness"))
           datasetup.Recordset.MoveNext
        Loop
    frame3d1.Refresh
```

End Sub

Sub Combo3_GotFocus ()

```
    Dim sql As String
        sql = "select distinct [workmaterial name] from " + datasource
    datasetup.RecordSource = sql
    datasetup.Refresh
    combo3.Clear
        Do While Not datasetup.Recordset.EOF
            combo3.AddItem datasetup.Recordset("Workmaterial name")
            datasetup.Recordset.MoveNext
        Loop
    frame3d1.Refresh
```

End Sub

Sub Command1_Click ()

```
If txtSetup(0).Text <> "" And txtSetup(1).Text <> "" And txtSetup(2).Text <> "" And
txtSetup(3).Text <> "" And txtSetup(4).Text <> "" And txtSetup(5).Text <> "" And
txtSetup(6).Text <> "" And txtSetup(7).Text <> "" And txtSetup(8).Text <> "" And
txtSetup(9).Text <> "" And txtSetup(10).Text <> "" Then
```

```
Else
   MsgBox "Please complete the necessary information"
   Exit Sub
End If

procfile = txtSetup(0).Text
tempfile = txtSetup(1).Text
recomfile = Left$(txtSetup(1).Text, Len(txtSetup(1).Text) - 4) + ".REM"
HP = CInt(txtSetup(2).Text)
efficiency = CSng(txtSetup(3).Text)
rapids = CSng(txtSetup(4).Text)
maxsp = CInt(txtSetup(5).Text)
maxfd = CSng(txtSetup(6).Text)
maxdoc = CSng(txtSetup(7).Text)
opcost = CSng(txtSetup(8).Text)
loadtime = CSng(txtSetup(9).Text)
chngtime = CSng(txtSetup(10).Text)
reindextm = CSng(txtSetup(11).Text)
setuptm = CSng(txtSetup(12).Text)
lotsize = CInt(txtSetup(13).Text)

frame3d1.Enabled = True

End Sub

Sub databs_click (indexbs As Integer)

Dim DK, SK, FK As String

Select Case indexbs

     Case 0
      dbase.Show 1

     Case 1
         Dim pi As Single
         Dim msg As String
         Dim nl
         ReDim criteria(arraysize) As String
         nl = Chr(10) & Chr(13)
         pi = 3.1415926

For index = 0 To arraysize

   If    cmbtooltype(index).Text <> "" And combo1.Text <> "" And combo2.Text <>
         "" And cmbtoolcode(index).Text <> "" And cmbtooldia(index) <> "" Then

         combinationsetup(index) = Left$(cmbtooltype(index).Tag, 1) + combo1.Text &
         Right$(combo2.Text, 3) & cmbtoolcode(index).Text &
         CStr(CSng(cmbtooldia(index).Text) * 1000)
   Else

         MsgBox "Please complete necessary input!"
```

```
        Exit Sub

    End If

    criteria(index) = "[Combination #] =" + " '" + combinationsetup(index) + "'"
    datasetup.RecordSource = "select * from " + datasource
    datasetup.Refresh
    datasetup.Recordset.FindFirst criteria(index)

      If datasetup.Recordset.NoMatch Then

        MsgBox "There is no such combination in database for tool #" +
txttoolNum(index).Text

      End If

    workmaterialname = combo3.Text
    workhardness = combo2.Text
    workmaterialcode = combo1.Text
    no_of_tool = CInt(tool.Text)

    If turn Then workmaterialdia = CSng(workdia.Text)

    toolnum(index) = CInt(txttoolNum(index).Text)
    tooltype(index) = cmbtooltype(index).Text
    toolmaterialname(index) = txttoolmaterial(index).Text
    toolmaterialcode(index) = cmbtoolcode(index).Text

    If cmbtooldia(index).Text = "" Then
      tooldia(index) = 1
    Else
      tooldia(index) = CSng(cmbtooldia(index).Text)
    End If

      toolresharpen(index) = CInt(txtresharpen(index).Text)
      toolcost(index) = CSng(txttoolcost(index).Text)

    Select Case TTG(index)

        Case "TURN", "CUTOFF", "BORING"
            If txttoolteeth(index).Text = "" Then
              tooledge(index) = 1
            Else
              tooledge(index) = CInt(txttoolteeth(index).Text)
            End If
        Case "MILL"
              toolteeth(index) = CInt(txttoolteeth(index).Text)

    End Select

  For k = 0 To 3

    DK = "d" + CStr(k + 1)
    SK = "s" + CStr(k + 1)
```

```
    FK = "f" + CStr(k + 1)

If datasetup.Recordset(DK) <> 0 Then
        D(index, k) = datasetup.Recordset(DK)

  Select Case TTG(index)
  Case "TURN", "CUTOFF", "BORING"
        SFM(index, k) = datasetup.Recordset(SK)
        S(index, k) = CInt(datasetup.Recordset(SK) * 12 / (pi * workdia)) 'RPM
  Case "MILL", "DRILL", "REAMING"
        SFM(index, k) = datasetup.Recordset(SK)
        S(index, k) = CInt(datasetup.Recordset(SK) * 12 / (pi * tooldia(index))) 'RPM

  End Select

  Select Case TTG(index)

  Case "TURN", "CUTOFF", "BORING", "DRILL", "REAMING"
        FED(index, k) = datasetup.Recordset(FK)
        F(index, k) = FED(index, k) * S(index, k) 'IPM=IPR x RPM
  Case "MILL"
        FED(index, k) = datasetup.Recordset(FK)
        F(index, k) = FED(index, k) * toolteeth(index) * S(index, k) 'IPM=IPT x teeth x RPM

  End Select

Else
        D(index, k) = 0
End If

  n(index) = datasetup.Recordset("n")
  c(index) = datasetup.Recordset("c")
  pu(index) = datasetup.Recordset("Pu")

  Select Case TTG(index)

  Case "CUTOFF"
    hpu(index, k) = F(index, k) * tooldia(index) * pu(index) / efficiency * pi * workdia
  Case "TURN", "BORING"
    hpu(index, k) = F(index, k) * D(index, k) * pu(index) / efficiency * pi * workdia
  Case "MILL"
    hpu(index, k) = F(index, k) * D(index, k) * pu(index) / efficiency * pi * tooldia(index)
  Case "DRILL", "REAMING"
    hpu(index, k) = F(index, k) * pu(index) / efficiency * pi * (tooldia(index) / 2) ^ 2

  End Select

  If datasetup.Recordset(SK) <> 0 Then
    tlife(index, k) = (c(index) / CInt(datasetup.Recordset(SK))) ^ (1 / n(index))
  Else
    tlife(index, k) = 0
  End If

Next k
```

```
   Select Case TTG(index)

   Case "TURN", "CUTOFF", "BORING"
      tooltotalcost(index) = opcost / 60 * chngtime + toolcost(index) / tooledge(index)
      Elife(index) = (1 / n(index) - 1) * tooltotalcost(index) / (opcost / 60)
      espeed(index) = CInt((c(index) / Elife(index) ^ n(index)) * 12 / pi / workdia)
      Mlife(index) = (1 / n(index) - 1) * chngtime
      mspeed(index) = CInt((c(index) / Mlife(index) ^ n(index)) * 12 / pi / workdia)
   Case "MILL", "DRILL", "REAMING"
      tooltotalcost(index) = opcost / 60 * chngtime + toolcost(index)
      Elife(index) = (1 / n(index) - 1) * tooltotalcost(index) / (opcost / 60)
      espeed(index) = CInt((c(index) / Elife(index) ^ n(index)) * 12 / pi / tooldia(index))
      Mlife(index) = (1 / n(index) - 1) * chngtime
      mspeed(index) = CInt((c(index) / Mlife(index) ^ n(index)) * 12 / pi / tooldia(index))

   End Select

   datasetup.Refresh

'Parameter recommendations

cmbDOC(index).Clear
cmbDOC(index).AddItem CStr(D(index, 0))
cmbDOC(index).AddItem CStr(D(index, 1))
cmbDOC(index).AddItem CStr(D(index, 2))
cmbDOC(index).AddItem CStr(D(index, 3))


 ok = 0
 For j = 0 To 3

   If hpu(index, 0) > HP Then
      MsgBox " The Machine power required is too large that there is no available parameter
in database!"
      cmbDOC(index).Text = CStr(recomdoc(index, 0))
      txtspeed(index).Text = CStr(recomspeed(index, 0))
      txtfeed(index).Text = CStr(recomfeed(index, 0))
      txtlife(index).Text = CStr(recomlife(index, 0))
   ElseIf hpu(index, j) <> 0 And hpu(index, j) <= HP Then

        If espeed(index) > S(index, 0) Then

            MsgBox "There is no available parameter in database! "
            cmbDOC(index).Text = CStr(D(index, 0))
            txtspeed(index).Text = CStr(S(index, 0))
            txtfeed(index).Text = CStr(F(index, 0))
            txtlife(index).Text = CStr(tlife(index, 0))

        ElseIf  S(index, j) <> 0 And espeed(index) < S(index, j) And (S(index, j) <
                mspeed(index) Or S(index, j) > mspeed(index)) Then

            ok = ok + 1
            ReDim recomdoc(index, ok) As Single
```

```
        ReDim recomspeed(index, ok) As Integer
        ReDim recomfeed(index, ok) As Single
        ReDim recomlife(index, ok) As Single
        recomdoc(index, j) = D(index, j)
        recomspeed(index, j) = S(index, j)
        recomfeed(index, j) = F(index, j)
        recomlife(index, j) = tlife(index, j)
        cmbDOC(index).Text = CStr(recomdoc(index, j))

        oldlife(index) = CSng(recomlife(index, j))
        oldspeed(index) = CInt(recomspeed(index, j))
        oldfeed(index) = CSng(recomfeed(index, j))
        olddoc(index) = CSng(recomdoc(index, j))

        cmbDOC(index).Text = CStr(recomdoc(index, j))
        txtspeed(index).Text = CStr(recomspeed(index, j))
        txtfeed(index).Text = CStr(recomfeed(index, j))
        txtlife(index).Text = CStr(recomlife(index, j))

    End If

    If oldspeed(index) > maxsp Then

      newspeed(index) = maxsp 'RPM
      txtspeed(index).Text = CStr(maxsp)

        Select Case TTG(index)

        Case "TURN", "CUTOFF", "BORING", "DRILL", "REAMING"
           newfeed(index) = FED(index, j) * newspeed(index)
        Case "MILL"
           newfeed(index) = FED(index, j) * toolteeth(index) * newspeed(index)

        End Select

        txtfeed(index).Text = CStr(newfeed(index))

        If newfeed(index) > maxfd Then

           newfeed(index) = maxfd
           txtfeed(index).Text = CStr(maxfd)

        End If

      newlife(index) = oldlife(index) * (oldspeed(index) / newspeed(index)) ^ (1 /
n(index))
      txtlife(index).Text = CStr(newlife(index))

    End If

    If oldspeed(index) >= maxsp And newfeed(index) >= maxfd Then

    'Calculate for maximal depth of cut
```

```
Select Case TTG(index)

Case "CUTOFF"
    hptest(index) = maxfd * tooldia(index) * pu(index) / efficiency * pi * workdia
Case "TURN", "BORING"
    hptest(index) = maxfd * olddoc(index) * pu(index) / efficiency * pi * workdia
Case "MILL"
    hptest(index) = maxfd * olddoc(index) * pu(index) / efficiency * pi *
tooldia(index)
Case "DRILL", "REAMING"
    hptest(index) = maxfd * pu(index) / efficiency * pi * (tooldia(index) / 2) ^ 2

End Select

testdoc = olddoc(index)
If hptest(index) > HP Then
        If TTG(index) = "TURN" Or TTG(index) = "BORING" Or TTG(index) =
        "MILL" Then

' "Case 1: If HPtest(index) is greater than machine power limit., reduce the HPtest(index)
to the limit "

 Do Until hptest(index) <= HP * .8 Or testdoc <= 0   'Assume 80% of machine
power is allowed

            testdoc = testdoc - .01

        Select Case TTG(index)

            Case "TURN", "BORING"
       hptest(index) = maxfd * testdoc * pu(index) / efficiency * pi * workdia
            Case "MILL"
        hptest(index) = maxfd * testdoc * pu(index) / efficiency * pi * tooldia(index)

        End Select

   Loop

     End If

     cmbDOC(index).Text = CStr(testdoc)
Else

     If TTG(index) = "TURN" Or TTG(index) = "BORING" Or TTG(index) = "MILL"
Then

' "Case 2: If HPtest(index) is less than machine power limit, enlarge the Hptest(index) to
the limit "

   Do Until hptest(index) >= HP * .8 Or testdoc >= maxdoc

            testdoc = testdoc + .01

        Select Case TTG(index)
```

```
        Case "TURN", "BORING"
              hptest(index) = maxfd * testdoc * pu(index) / efficiency * pi * workdia
        Case "MILL"
              hptest(index) = maxfd * testdoc * pu(index) / efficiency * pi * tooldia(index)
              End Select

         Loop

           End If
           cmbDOC(index).Text = CStr(testdoc)

       End If

   Else                 'Calculate for maximal depth of cut

       Select Case TTG(index)

       Case "CUTOFF"
 hptest(index) = oldfeed(index) * tooldia(index) * pu(index) / efficiency * pi * workdia
       Case "TURN", "BORING"
 hptest(index) = oldfeed(index) * olddoc(index) * pu(index) / efficiency * pi * workdia
       Case "MILL"
 hptest(index) = oldfeed(index) * olddoc(index) * pu(index) / efficiency * pi * tooldia(index)
       Case "DRILL", "REAMING"
 hptest(index) = oldfeed(index) * pu(index) / efficiency * pi * (tooldia(index) / 2) ^ 2
       End Select

       testdoc = olddoc(index)

       If TTG(index) = "TURN" Or TTG(index) = "BORING" Or TTG(index) = "MILL" Then
  '"Case 3: Other than Case 1 or Case 2"
         Do Until hptest(index) >= HP * .8 Or testdoc >= maxdoc
                 testdoc = testdoc + .01
                 Select Case TTG(index)
                 Case "TURN", "BORING"
     hptest(index) = oldfeed(index) * testdoc * pu(index) / efficiency * pi * workdia
                 Case "MILL"
     hptest(index) = oldfeed(index) * testdoc * pu(index) / efficiency * pi * tooldia(index)

                 End Select
         Loop

           End If

           cmbDOC(index).Text = CStr(testdoc)

     End If
   End If
 Next j

Next index

MsgBox "Complete data check! "
```

```
dataitem(1).Enabled = 1
dataitem(2).Enabled = 1
frapara.Enabled = True

Case 2
   Y = Shell("calc.exe", 1)
End Select
```

End Sub

Sub dataitem_Click (k As Integer)

```
Select Case k

Case 0 'Database

Case 1 'Recommendation
        recommandation
        fraRecom.Visible = 1
        winview.Enabled = 1
        Dim file As Integer
      On Error GoTo ErrHandler
         file = FreeFile
         Open recomfile For Input As file   ' Open file selected on File Open About.
         txtrecom.Text = Input$(LOF(file), file)
         Close file                          ' Close file.
         Exit Sub

ErrHandler:
      MsgBox "Error encountered while trying to open file, please retry.", 48, "Text Editor"
      Close file
      Exit Sub

Case 2 'Create SmartCAM Template Programs

m = MsgBox("Do you want to write a SmartCAM template file?", 36, "Template File")
        If m = 6 Then
        If turn Then

Open tempfile For Output As #1  (SmartCAM Template File)

Print #1, "@START"
Print #1, "#EVAL(#S9=TOOL#TOOL)"
Print #1, "#EVAL(#FEED=#TABLE(#S9,1))"
Print #1, "#IF(#FEED>" + CStr(rapids) + ")<#EVAL(#FEED=" + CStr(rapids) + ")>"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)#EVAL(#V0=" + CStr(rapids) + ")"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)"
Print #1, "#NEXTPT"
Print #1, ""
Print #1, "@TLCHG"
Print #1, "#EVAL(#V3=(SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2))/#V0)"
Print #1, "#EVAL(#V4=(SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2))/#FEED)"
Print #1, "#EVAL(#V5=#V5+#V3)#EVAL(#V6=#V6+#V4)#EVAL(#V7=#V5+#V6+" +
CStr(reindextm) + ")#EVAL(#V8=#V8+#V7)"
```

```
Print #1,
"#LTOOL,#FMT(#V5,D2.4),#FMT(#V6,D2.4),#FMT(#V7,D2.4),#FMT(#V8,D2.4),#S4,#S5,#
S6,#S7,#S8,"
Print #1,
"#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)#UPDATE(#V3)#UPDATE(#V4)#EVAL(#V5=0)"
Print #1, "#EVAL(#V6=0)#UPDATE(#V8)#EVAL(#V7=0)"
Print #1, ""
Print #1, "@TOOLCHG"
Print #1, "#CALL(TLCHG)"
Print #1, "#UPDATE(#TOOL)"
Print #1, "#CALL(VAL)"
Print #1, "#EVAL(#S9=TOOL#TOOL)"
Print #1, "#EVAL(#FEED=#TABLE(#S9,1))"
Print #1, "#IF(#FEED>" + CStr(rapids) + ")<#EVAL(#FEED=" + CStr(rapids) + ")>"
Print #1, "#NEXTPT"
Print #1, ""
Print #1, "@END"
Print #1, "#EVAL(#V3=(SQR(ABS((#V1-#XHOME)/2)^2+ABS(#V2-#ZHOME)^2))/#V0)"
Print #1, "#EVAL(#V4=(SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2))/#FEED)"
Print #1, "#EVAL(#V5=#V5+#V3)#EVAL(#V6=#V6+#V4)#EVAL(#V7=#V5+#V6+" +
CStr(reindextm) + ")#EVAL(#V8=#V8+#V7)"
Print #1,
"#LTOOL,#FMT(#V5,D2.4),#FMT(#V6,D2.4),#FMT(#V7,D2.4),#FMT(#V8,D2.4),#S4,#S5,#
S6,#S7,#TDESC,"
Print #1, ""
Print #1, "@VAL"
Print #1, "#EVAL(#S4=#TABLE(#S9,4))" 'tool life
Print #1, "#EVAL(#S5=#TABLE(#S9,5))" 'tool edge
Print #1, "#EVAL(#S6=#TABLE(#S9,6))" 'tool resharpen
Print #1, "#EVAL(#S7=#TABLE(#S9,7))" 'tool cost
Print #1, "#EVAL(#S8=#TDESC)"        'tool description
Print #1, ""
Print #1, "@STPROF"
Print #1, "#CALL(VAL)"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@RAP"
Print #1, "#EVAL(#V9=SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2)"
Print #1, "#EVAL(#V3=(SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2))/#V0)"
Print #1, "#EVAL(#V5=#V5+#V3)"
Print #1, "#UPDATE(#V3)#UPDATE(#V5)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)"
Print #1, ""
Print #1, "@LINE"
Print #1, "#EVAL(#V9=SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2)"
Print #1, "#EVAL(#V4=(SQR(ABS((#V1-#XPOS)/2)^2+ABS(#V2-#ZPOS)^2))/#FEED)"
Print #1, "#EVAL(#V6=#V6+#V4)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)#UPDATE(#V4)#UPDATE(#V6)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#ZPOS)"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@ARC"
Print #1, "#EVAL(#TANG=ABS(#TANG*3.14159/360))"
Print #1, "#EVAL(#V9=#ARAD*#TANG)"
```

```
Print #1, "#EVAL(#V4=#V9/#FEED)"
Print #1, "#EVAL(#V6=#V6+#V4)"
Print #1, "#UPDATE(#V4)#UPDATE(#V6)"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@FXCYCLE1"
Print #1, "#EVAL(#V9=SQR(ABS((#V1-#XPASS)/2)^2+ABS(#V2-#ZPASS)^2)"
Print #1, "#EVAL(#V4=(SQR(ABS((#V1-#XPASS)/2)^2+ABS(#V2-#ZPASS)^2))/#FEED)"
Print #1, "#EVAL(#V6=#V6+#V4)"
Print #1, ""
Print #1, "@FXCYCLE2"
Print #1, "#UPDATE(#V4)#UPDATE(#V6)"
Print #1, "#EVAL(#V1=#XPASS)#EVAL(#V2=#ZPASS)#EVAL(#V0=26.5)"
Print #1, ""
Print #1, "@FXCYCLE3"
Print #1, "#EVAL(#V9=ABS(#XPOS-#XPASS)/2)#EVAL(#V4=#V9/#FEED)"
Print #1,
"#EVAL(#V6=#V6+#V4)#UPDATE(#V4)#UPDATE(#V6)#EVAL(#V9=ABS(#ZPOS-#ZPASS
))"
Print #1,
"#EVAL(#V4=#V9/#FEED)#EVAL(#V6=#V6+#V4)#UPDATE(#V4)#UPDATE(#V6)"
Print #1, "#EVAL(#V9=ABS(#XPOS-#XPASS)/2)#EVAL(#V3=#V9/#V0)"
Print #1,
"#EVAL(#V5=#V5+#V3)#UPDATE(#V3)#UPDATE(#V5)#EVAL(#V9=ABS(#ZPOS-#ZPASS
))"
Print #1, "#EVAL(#V3=#V9/#V0)#EVAL(#V5=#V5+#V3)#UPDATE(#V3)#UPDATE(#V5)"
Print #1, ""
Print #1, "@FXD1"
Print #1, "#CALL(FXCYCLE1)"
Print #1, "#CALL(FXCYCLE2)"
Print #1, ""
Print #1, "@FXD2"
Print #1, "#CALL(FXCYCLE3)"
Print #1, ""
Print #1, "@FXD3"
Print #1, "#CALL(FXCYCLE3)"
Print #1, ""
Print #1, "@FXD4"
Print #1, "#CALL(FXCYCLE1)"
Print #1, "#CALL(FXCYCLE2)"
Print #1, ""
Print #1, "@FXD5"
Print #1, "#CALL(FXCYCLE1)"
Print #1, "#CALL(FXCYCLE2)"
Print #1, ""
Print #1, "@FXD6"
Print #1, "#CALL(FXCYCLE1)"
Print #1, "#CALL(FXCYCLE2)"
Print #1, ""
Print #1, "@FXD7"
Print #1, "#CALL(FXCYCLE1)"
Print #1, "#CALL(FXCYCLE2)"
Print #1, ""
Print #1, "@"
```

```
Print #1, "78901234567890123456789 0"
Print #1, ""
   For i = 0 To arraysize
Print #1, "@TOOL" + CStr(Format(toolnum(i), "000"))
Print #1, "1," + txtfeed(i).Text
Print #1, "2," + txtspeed(i).Text
Print #1, "3," + cmbDOC(i).Text
Print #1, "4," + Format(txtlife(i).Text, "0.00")
Print #1, "5," + txttoolteeth(i).Text
Print #1, "6," + txtresharpen(i).Text
Print #1, "7," + txttoolcost(i).Text
Print #1, ""
   Next i
Close

ElseIf mill Then

Open tempfile For Output As #1

Print #1, "@START"
Print #1, "#EVAL(#S9=TOOL#TOOL)"
Print #1, "#EVAL(#FEED=#TABLE(#S9,1))"
Print #1, "#IF(#FEED>" + CStr(rapids) + ")<#EVAL(#FEED=" + CStr(rapids) + ")>"
Print #1, "#EVAL(#V8=#TLEN)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1,
"#EVAL(#V0=0)#EVAL(#V4=0)#EVAL(#V5=0)#EVAL(#V6=0)#EVAL(#V7=0)#EVAL(#V9
=0)"
Print #1, ""
Print #1, "@TLCHG"
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
POS))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V4=#V9/" + CStr(rapids) + ")#EVAL(#V5=0)"
Print #1,
"#EVAL(#V6=#V6+#V4)#EVAL(#V7=#V7+#V5)#EVAL(#TLTIME=#V6+#V7+" +
CStr(reindextm) + ")#EXLN"
Print #1, "#EVAL(#V0=#V0+#TLTIME)"
Print #1,
"#LTOOL,#FMT(#V6,D2.4),#FMT(#V7,D2.4),#FMT(#TLTIME,D2.4),#FMT(#V0,D2.4),#S4,#
S5,#S6,#S7,#S8,"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V4)#UPDATE(#V5)#UPDATE(#V0)"
Print #1, "#EVAL(#V6=0)#EVAL(#V7=0)#EVAL(#TLTIME=0)"
Print #1, ""
Print #1, "@TOOLCHG"
Print #1, "#CALL(TLCHG)"
Print #1, "#EVAL(#S9=TOOL#TOOL)"
Print #1, "#EVAL(#FEED=#TABLE(#S9,1))"
Print #1, "#IF(#FEED>" + CStr(rapids) + ")<#EVAL(#FEED=" + CStr(rapids) + ")>"
Print #1, "#CALL(VAL)"
Print #1, ""
Print #1, "@END"
```

```
Print #1, "#CALL(TLCHG)"
Print #1, ""
Print #1, "@VAL"
Print #1, "#EVAL(#S4=#TABLE(#S9,4))" 'tool life
Print #1, "#EVAL(#S5=#TABLE(#S9,5))" 'tool teeth
Print #1, "#EVAL(#S6=#TABLE(#S9,6))" 'tool resharpen
Print #1, "#EVAL(#S7=#TABLE(#S9,7))" 'tool cost
Print #1, "#EVAL(#S8=#TDESC)"        'tool description
Print #1, ""
Print #1, "@STPROF"
Print #1, "#CALL(VAL)"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@ENDPROF"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@RAP"
Print #1, "< #ABSI >"
Print #1, "#EVAL(#S4=#TABLE(#S9,4))" 'tool life
Print #1, "#EVAL(#S5=#TABLE(#S9,5))" 'tool teeth
Print #1, "#EVAL(#S6=#TABLE(#S9,6))" 'tool resharpen
Print #1, "#EVAL(#S7=#TABLE(#S9,7))" 'tool cost
Print #1, "#EVAL(#S8=#TDESC)"        'tool description
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
POS))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V4=#V9/" + CStr(rapids) + ")"
Print #1, "#EVAL(#V6=#V6+#V4)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V4)#UPDATE(#V6)"
Print #1, ""
Print #1, "@LINE"
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
POS))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V5=#V9/#FEED)"
Print #1, "#EVAL(#V7=#V7+#V5)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V5)#UPDATE(#V7)"
Print #1, ""
Print #1, "@ARC"
Print #1, "#CALL(VAL)"
Print #1, "#IF(ABS(#V1-#XPOS)<.001,#OR
ABS(#V2-#YPOS)<.001)<#CALL(LINE)>#ELSE<"
Print #1, "#EVAL(#TANG=ABS(#TANG*3.14159/360))"
Print #1, "#EVAL(#V9=#ARAD*#TANG)#EVAL(#V5=#V9/#FEED)"
Print #1, "#EVAL(#V7=#V7+#V5)"
Print #1, "#UPDATE(#V5)#UPDATE(#V7)"
Print #1, ""
Print #1, "@ZCLRMV"
Print #1, "#CALL(RAP)"
Print #1, ""
```

```
Print #1, "@ZCHKMV"
Print #1, "#CALL(RAP)"
Print #1, ""
Print #1, "@ZDPTHMV"
Print #1, "#CALL(LINE)"
Print #1, ""
Print #1, "@FXCYCLE"
Print #1, "#EVAL(#U1=1)"
Print #1, "< #ABSI >"
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
POS))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V4=#V9/" + CStr(rapids) + ")"
Print #1, "#EVAL(#V6=#V6+#V4)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V4)#UPDATE(#V6)"
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
DPTH))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V5=#V9/#FEED)"
Print #1, "#EVAL(#V7=#V7+#V5)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V5)#UPDATE(#V7)"
Print #1, ""
Print #1, "@FXD1"
Print #1, "#IF(#TLTYPE=3)<#EVAL(#FXD=4)>#ELSE<#CALL(FXCYCLE)>"
Print #1, ""
Print #1, "@FXD2"
Print #1, "#EVAL(#U1=1)"
Print #1, "#CALL(RAP)"
Print #1,
"#EVAL(#V1=ABS(#V1-#XPOS))#EVAL(#V2=ABS(#V2-#YPOS))#EVAL(#V3=ABS(#V3-#Z
DPTH))"
Print #1, "#EVAL(#V9=SQR(#V1^2+#V2^2+#V3^2))"
Print #1, "#EVAL(#V5=#V9/#FEED+#DWELL/60)"
Print #1, "#EVAL(#V7=#V7+#V5)"
Print #1, "#EVAL(#V1=#XPOS)#EVAL(#V2=#YPOS)#EVAL(#V3=#ZPOS)"
Print #1, "#UPDATE(#V5)#UPDATE(#V7)"
Print #1, ""
Print #1, "@FXD3"
Print #1, "#EVAL(#S3=TAP)"
Print #1, "#CALL(FXCYCLE)"
Print #1, ""
Print #1, "@FXD4"
Print #1, "#CALL(FXCYCLE)"
Print #1, ""
Print #1, "@FXD5"
Print #1, "#CALL(FXCYCLE)"
Print #1, ""
Print #1, "@DWELL"
Print #1, "#EVAL(#V7=#V7+#DWELL/60)"
Print #1, "#UPDATE(#V7)"
```

```
Print #1, ""
Print #1, "@"
Print #1, "78901234567890 1234567890"
Print #1, ""
   For i = 0 To arraysize
Print #1, "@TOOL" + CStr(Format(toolnum(i), "000"))
Print #1, "1," + txtfeed(i).Text
Print #1, "2," + txtspeed(i).Text
Print #1, "3," + cmbDOC(i).Text
Print #1, "4," + Format(txtlife(i).Text, "0.00")
Print #1, "5," + txttoolteeth(i).Text
Print #1, "6," + txtresharpen(i).Text
Print #1, "7," + txttoolcost(i).Text
Print #1, ""
   Next i
Close #1
End If

MsgBox "SmartCAM template file has been writen to " + tempfile
ElseIf m = 7 Then
End If
End Select

End Sub

Sub Form_Unload (Cancel As Integer)

maxid = 0
End Sub

Sub frmtool_DragOver (Source As Control, X As Single, Y As Single, State As Integer)

frmtool.Refresh

End Sub

Sub getsdata ()
   Dim i As Integer

   'Setup data "sdata global variable"

   sara.profile = frmSetup.txtSetup(0).Text: sdata = sara.profile + ","
   sara.tmpfile = frmSetup.txtSetup(1).Text: sdata = sdata + sara.tmpfile + ","

   If IsNumeric(frmSetup.txtSetup(2).Text) Then
      sara.mpower = CSng(frmSetup.txtSetup(2).Text)
   Else
      sara.mpower = 0
   End If
   sdata = sdata + Str$(sara.mpower) + ","

   If IsNumeric(frmSetup.txtSetup(3).Text) Then
      sara.eff = CSng(frmSetup.txtSetup(3).Text)
   Else
```

```
    sara.eff = 1
End If
sdata = sdata + Str$(sara.eff) + ","

If IsNumeric(frmSetup.txtSetup(4).Text) Then
    sara.rapid = CSng(frmSetup.txtSetup(4).Text)
Else
    sara.rapid = 0
End If
sdata = sdata + Str$(sara.rapid) + ","

If IsNumeric(frmSetup.txtSetup(5).Text) Then
    sara.maxsp = CSng(frmSetup.txtSetup(5).Text)
Else
    sara.maxsp = 0
End If
sdata = sdata + Str$(sara.maxsp) + ","

If IsNumeric(frmSetup.txtSetup(6).Text) Then
    sara.maxfd = CSng(frmSetup.txtSetup(6).Text)
Else
    sara.maxfd = 0
End If
sdata = sdata + Str$(sara.maxfd) + ","

If IsNumeric(frmSetup.txtSetup(7).Text) Then
    sara.maxdoc = CSng(frmSetup.txtSetup(7).Text)
Else
    sara.maxdoc = 0
End If
sdata = sdata + Str$(sara.maxdoc) + ","

If IsNumeric(frmSetup.txtSetup(8).Text) Then
    sara.opcost = CSng(frmSetup.txtSetup(8).Text)
Else
    sara.opcost = 0
End If
sdata = sdata + Str$(sara.opcost) + ","

If IsNumeric(frmSetup.txtSetup(9).Text) Then
     sara.loadtime = CSng(frmSetup.txtSetup(9).Text)
Else
     sara.loadtime = 0
End If
sdata = sdata + Str$(sara.loadtime) + ","

If IsNumeric(frmSetup.txtSetup(10).Text) Then
     sara.changtime = CSng(frmSetup.txtSetup(10).Text)
Else
     sara.changtime = 0
End If
sdata = sdata + Str$(sara.changtime) + ","
If IsNumeric(frmSetup.txtSetup(11).Text) Then
     sara.reindex = CSng(frmSetup.txtSetup(11).Text)
```

```
    Else
        sara.reindex = 0
    End If
    sdata = sdata + Str$(sara.reindex) + ","
    If IsNumeric(frmSetup.txtSetup(12).Text) Then
        sara.setuptime = CSng(frmSetup.txtSetup(12).Text)
    Else
        sara.setuptime = 0
    End If
    sdata = sdata + Str$(sara.setuptime) + ","
    If IsNumeric(frmSetup.txtSetup(13).Text) Then
        sara.lotsize = CSng(frmSetup.txtSetup(13).Text)
    Else
        sara.lotsize = 1
    End If
    sdata = sdata + Str$(sara.lotsize) + ","
    sara.wmatl = combo3.Text: sdata = sdata + sara.wmatl + ","
    sara.wcode = combo1.Text: sdata = sdata + sara.wcode + ","
    sara.whard = combo2.Text: sdata = sdata + sara.whard + ","

    If IsNumeric(tool.Text) Then
        sara.tool = CInt(tool.Text)
    Else
        sara.tool = 1
    End If
    sdata = sdata + Str$(sara.tool) + ","

    If IsNumeric(workdia.Text) Then
        sara.wdia = CSng(workdia.Text)
    Else
        sara.wdia = 1
    End If
    sdata = sdata + Str$(sara.wdia) + ","

'Tool data "tdata global variable

tdata = ""    'initialization
For i = 0 To arraysize
        tdata = tdata + txttoolNum(i).Text + "," + cmbtooltype(i).Text + "," +
        cmbtoolcode(i).Text + "," + txttoolmaterial(i).Text + "," + cmbtooldia(i).Text +
        "," + txttoolteeth(i).Text + "," + txtresharpen(i).Text + "," + txttoolcost(i).Text
        + ","
Next i

End Sub

Sub optApp_Click (index As Integer, value As Integer)

   Select Case index

   Case 0
      turnON
      setupFileItem(3).Enabled = 1
      datasource = "datb"
```

```
   Case 1
      millON
      setupFileItem(3).Enabled = 1
      datasource = "datb"

   End Select

datasetup.Refresh

If mill Then
  panel3d4.Visible = False
  workdia.Visible = False
  lbltool(6).Caption = "Teeth"
ElseIf turn Then
  panel3d4.Enabled = True
  workdia.Enabled = True
  lbltool(6).Caption = "Edge"
End If
```

End Sub

Sub recommandation ()

```
Open recomfile For Output As #1

Print #1, "Process File : " + procfile
Print #1, "Template File: " + tempfile
Print #1, "Machine Power: " + CStr(HP) + " HP"
Print #1, "Machine Efficiency: " + CStr(efficiency)
Print #1, "Rapid Traverse Feed: " + CStr(rapids) + " ipm"
Print #1, "Maximum Speed: " + CStr(maxsp) + " rpm"
Print #1, "Maximum Feed: " + CStr(maxfd) + " ipm"
Print #1, "Maximum Depth of Cut: " + CStr(maxdoc) + " inch"
Print #1, "Operation Cost: $" + CStr(opcost) + "/hr"
Print #1, "Load/Unload Time: " + CStr(loadtime) + " min"
Print #1, "Tool Changing Time: " + CStr(chngtime) + "min"
Print #1, "Tool Reindex Time : " + CStr(reindextm) + "min"
Print #1, "Machine Setup Time:" + CStr(setuptm) + "min"
Print #1, "Lot Size per batch: " + CStr(lotsize) + "piece"
Print #1, "Work Material Name: " & workmaterialname & " ; Code: " & workmaterialcode
Print #1, "Work Material Hardness: " & workhardness & " BHN"

   If turn Then
      Print #1, "Workpiece original diameter: " & CStr(workmaterialdia) + """"
   Else
   End If
Print #1, "Number of tools used in process: " & CStr(no_of_tool)

Print #1, ""
Print #1, "==============Tool Information================="
   If mill Then
Print #1, " Tool    Tool      Tool      Code   Diameter  Teeth  Resharpen  Cost"
```

```
Print #1, "Number   Type     Material    (AISI)   (inch)   (#)   (times)   ($)"
Print #1, "_____ _________ ___________ _______ _______ _____ _________ ___"
        For i = 0 To arraysize
Print #1,       Spc(2); CStr(toolnum(i)); Tab(10); tooltype(i); Tab(22); toolmaterialname(i);
        Tab(39); toolmaterialcode(i); Tab(48); CStr(tooldia(i)); Tab(58); CStr(toolteeth(i));
        Tab(66); CStr(toolresharpen(i)); Tab(75); CStr(toolcost(i))
        Next i
   End If

   If turn Then
Print #1, " Tool    Tool      Tool      Code   Diameter  Edges  Resharpen  Cost"
Print #1, "Number   Type     Material    (AISI)   (inch)   (#)   (times)   ($)"
Print #1, "_____ _________ ___________ _______ _______ _____ _________ ___"
        For i = 0 To arraysize
Print #1,       Spc(2); CStr(toolnum(i)); Tab(10); tooltype(i); Tab(22); toolmaterialname(i);
        Tab(39); toolmaterialcode(i); Tab(48); CStr(tooldia(i)); Tab(58); CStr(tooledge(i));
        Tab(66); CStr(toolresharpen(i)); Tab(75); CStr(toolcost(i))
        Next i
    End If
Print #1, "======================================="
Print #1, ""

For i = 0 To arraysize
Print #1,       "Tool number: " & CStr(toolnum(i)) & "; n= " & n(i) & "; C= " & c(i) & "
        ;Pu= " & pu(i) & "HP/cubic in/min"
Print #1,       "Index"; Tab(8); "D.O.C(in)"; Tab(22); "Speed(rpm)"; Tab(36); "Feed(ipm)";
        Tab(50); "HPactual(HP)"; Tab(64); "Tool Life(min)"
Print #1,       Spc(2); "1"; Tab(9); CStr(D(i, 0)); Tab(23); CStr(S(i, 0)); Tab(37); CStr(F(i,
        0)); Tab(51); CStr(Format(hpu(i, 0), "0.0")); Tab(65); CStr(Format(tlife(i, 0), "0.0"))
Print #1,       Spc(2); "2"; Tab(9); CStr(D(i, 1)); Tab(23); CStr(S(i, 1)); Tab(37); CStr(F(i,
        1)); Tab(51); CStr(Format(hpu(i, 1), "0.0")); Tab(65); CStr(Format(tlife(i, 1), "0.0"))
Print #1,       Spc(2); "3"; Tab(9); CStr(D(i, 2)); Tab(23); CStr(S(i, 2)); Tab(37); CStr(F(i,
        2)); Tab(51); CStr(Format(hpu(i, 2), "0.0")); Tab(65); CStr(Format(tlife(i, 2), "0.0"))
Print #1,       Spc(2); "4"; Tab(9); CStr(D(i, 3)); Tab(23); CStr(S(i, 3)); Tab(37); CStr(F(i,
        3)); Tab(51); CStr(Format(hpu(i, 3), "0.0")); Tab(65); CStr(Format(tlife(i, 3), "0.0"))
Print #1, "Minimum cost Tool life (Te) is: " & CStr(Elife(i)) + " min; Speed (Ve) is: " &
CStr(espeed(i)) + " rpm"
Print #1,       "Maximum productive Tool life (Tm) is: " & CStr(Mlife(i)) + " min; Speed
        (Vm) is: " & CStr(mspeed(i)) + " rpm"
Print #1,       "Suggested D.O.C. : " + cmbDOC(i).Text + " inch;  Speed is: " +
        txtspeed(i).Text + " rpm; Feed : " + txtfeed(i).Text + " ipm"
Print #1,       "Suggested Tool Life : " + txtlife(i).Text + " min"
Print #1, "------------------------------------------------------------------------"
Next i

Close #1

End Sub

Sub redimension ()

   ReDim toolnum(arraysize)
   ReDim tooltype(arraysize)
   ReDim toolmaterialname(arraysize)
```

```
    ReDim toolmaterialcode(arraysize)
    ReDim tooldia(arraysize)
       If mill Then
          ReDim toolteeth(arraysize)
       ElseIf turn Then
          ReDim tooledge(arraysize)
       End If
    ReDim toolcost(arraysize)
    ReDim toolresharpen(arraysize)
    ReDim tooltotalcost(arraysize)
    ReDim Elife(arraysize)
    ReDim espeed(arraysize)
    ReDim Mlife(arraysize)
    ReDim mspeed(arraysize)
    ReDim combinationsetup(arraysize)
    ReDim D(arraysize, 4)
    ReDim S(arraysize, 4)   'RPM SPEED
    ReDim F(arraysize, 4)   'IPM FEED
    ReDim SFM(arraysize, 4) 'FPM SPEED
    ReDim FED(arraysize, 4) 'IPR OR IPT FEED
    ReDim TTX(arraysize)
    ReDim TTG(arraysize)
    ReDim newlife(arraysize)
    ReDim newspeed(arraysize)
    ReDim newfeed(arraysize)
    ReDim newdoc(arraysize)
    ReDim olddoc(arraysize)
    ReDim oldlife(arraysize)
    ReDim oldspeed(arraysize)
    ReDim oldfeed(arraysize)
    ReDim olddoc(arraysize)
    ReDim n(arraysize)
    ReDim c(arraysize)
    ReDim pu(arraysize)
    ReDim hpu(arraysize, 4)
    ReDim hptest(arraysize)
    ReDim tlife(arraysize, 4)

End Sub

Sub setupFileItem_Click (index As Integer)

Select Case index                              ' Check index value of selected menu item.

    Case 0                                  ' If index = 0, the user chose "new process file"
       txtSetup(0).Text = FileName + "*.prc" ' Set default text in File Name text box.
       txtSetup(0).SelLength = 1            ' Show default text selected.
       txtSetup(1).Text = FileName + "*.tmp"      ' Set default text in File Name text box.
       txtSetup(1).SelLength = 1            ' Show default text selected.
    Case 1                                  ' If index = 1, the user chose "open template file"
       txtSetup(0).Text = ""                ' Clear text box
       frmOpenSave.Caption = "Open Process"
       frmOpenSave.txtOpenSave.Text = "*.prc"     ' Set default text in File Name text box.
       frmOpenSave.txtOpenSave.SelLength = 5      ' Show default text selected.
```

```
      Set main = frmOpenSave
      loadform (3)
      txtSetup(0).Text = FileName
   Case 2                              ' If index = 2, the user chose "Save As..."
      frmOpenSave.Caption = "Save Process as" ' Set caption of Open/Save dialog to
"Save"
      frmOpenSave.txtOpenSave.Text = FileName ' Set default text in File Name text box.
      frmOpenSave.txtOpenSave.SelLength = Len(frmOpenSave!txtOpenSave.Text)
      getsdata
      Set main = frmOpenSave
      loadform (3)
   Case 3                              ' If index = 3, the user chose "Close"
        Dim response As Integer
 response = MsgBox("Do you want to leave without saving your works ?", 36, "Warning")
      Select Case response
        Case 7        'NO
 frmOpenSave.Caption = "Save Process as" ' Set caption of Open/Save dialog to "Save"
 frmOpenSave.txtOpenSave.Text = FileName ' Set default text in File Name text box.
          frmOpenSave.txtOpenSave.SelLength = Len(frmOpenSave!txtOpenSave.Text)
          Set main = frmOpenSave
          loadform (3)
        Case 6        'YES
          Unload Me ' End this application and return to the Windows operating system.

      End Select
        turn = 0
        mill = 0
        maxid = 0
        arraysize = 0
   Case 4              ' This menu item is a sesarator bar, no code needs to be written here

   Case 5              ' If index = 5, the user chose "print"

   End Select

End Sub

Sub txtRecom_MouseDown (button As Integer, Shift As Integer, X As Single, Y As Single)

If button = 2 Then
      z = Shell("calc.exe", 1)
End If

End Sub

Sub txtSetup_LostFocus (index As Integer)

Select Case index

Case 0
If Right$(txtSetup(0).Text, 4) <> ".prc" And Right$(txtSetup(0).Text, 4) <> ".PRC" Then
           txtSetup(0).Text = txtSetup(0).Text + ".prc"
      Else
           txtSetup(0).Text = txtSetup(0).Text
```

```
        End If
        FileName = txtSetup(0).Text
        On Error Resume Next
Case 1
        txtSetup(1).Text = Left$(FileName, Len(FileName) - 4) + ".tmp"
        recomfile = Left$(FileName, Len(FileName) - 4) + ".REM"
Case 3
        If IsNumeric(txtSetup(3).Text) Then
           If CSng(txtSetup(3).Text) > 1# Or CSng(txtSetup(3).Text) < 0# Then
              MsgBox "The effective input ranges from 0 to 1 and no string data"
              txtSetup(3).Text = "": txtSetup(3).SetFocus
           End If
        Else
              MsgBox "The effective input ranges from 0 to 1 and no string data"
              txtSetup(3).Text = "": txtSetup(3).SetFocus
        End If

End Select

End Sub

Sub txtSpeed_KeyDown (index As Integer, KeyCode As Integer, Shift As Integer)

If KeyCode = 13 Then
   If txtspeed(index).Text <> "" Then
     newspeed(index) = CInt(txtspeed(index).Text)

           Select Case TTG(index)

           Case "TURN", "CUTOFF", "BORING", "DRILL", "REAMING"
              newfeed(index) = oldfeed(index) / oldspeed(index) * newspeed(index)
           Case "MILL"
     newfeed(index) = oldfeed(index) / oldspeed(index) / toolteeth(index) * newspeed(index)

           End Select

     txtfeed(index).Text = CStr(newfeed(index))
     newlife(index) = oldlife(index) * (oldspeed(index) / newspeed(index)) ^ (1 / n(index))
     txtlife(index).Text = CStr(newlife(index))

   Else

     MsgBox "Please enter speed in RPM"

   End If

End If

End Sub

Sub txttoolmaterial_GotFocus (index As Integer)

Dim criteria As String
criteria = "[Toolmaterial code] =" + " '" + cmbtoolcode(index).Text + "'"
```

```
datasetup.RecordSource = "select * from " + datasource
datasetup.Refresh
txttoolmaterial(index).DataField = "[Toolmaterial name]"
datasetup.Recordset.FindFirst criteria

If datasetup.Recordset.NoMatch Then
        MsgBox "There is no such combination in database!"
End If
txttoolmaterial(index).DataField = ""

End Sub


Sub txttoolNum_Change (index As Integer)

Dim msg As String
   msg = "Duplicated tool number !!"
   For i = 0 To index - 1
      If txttoolNum(index).Text = txttoolNum(i).Text Then
      MsgBox msg
      End If
   Next i
   For i = index + 1 To arraysize
      If txttoolNum(index).Text = txttoolNum(i).Text Then
      MsgBox msg
      End If
   Next i

End Sub

Sub view_Click (index As Integer)

   Select Case index

         Case 0
            fraRecom.Visible = 1
            view(0).Checked = 1
            view(1).Checked = 0
         Case 1
            fraRecom.Visible = 0
            view(0).Checked = 0
            view(1).Checked = 1

   End Select

End Sub
```

APPENDIX B. SmartCAM Macro Files

This appendix contains the listing of the SmartCAM Macro code used to generate the example shape file shown in Chapter 4. **Use of the SAMPT.mcl macro needs to run Turning application in SmartCAM.**

```
// Turn32 V7.01 06/16/92 Macro File
// C:\CCSYS\EXAMPLE\SAMPT.mcl
// CREATED: 05/08/1994
READ[FN="C:\\CCSYS\\EXAMPLE\\SAMPT2", FT=1]
// Macro default UI script for the READ command.
//
open[FileRead]                  // Is "FileRead" up already?
BEGIN                           // If not, do this block
        select[FileMenuAction] // click on "FileMenuAction"
        select[FileMenu2]              // ... "FileMenu2"
END
update[]                               // update parameters (er, values) in panel
close[FileRead]                 // DIALOGS ONLY! Drops the dialog..
execute[]                              // do the action itself.
WITH_TOOL[TL=1, WP="ZX_PLANE", OF=2, LV=0, CL=0.1, PT=?]
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
        select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
WITH_TOOL[TL=1, WP="ZX_PLANE", OF=2, LV=0, CL=0.1, PT=?]
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
        select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
ELMT_SEQ[BA=1, EL="10", ME=0]
// Macro default UI script for the ELMT_SEQ command.
open[InsertControl]
BEGIN
        select[XC1]
        select[ElmtSeqBIN]
END
update[]
execute[]
ELMT_SEQ[BA=1, EL="10", ME=0]
// Macro default UI script for the ELMT_SEQ command.
open[InsertControl]
BEGIN
        select[XC1]
```

```
        select[ElmtSeqBIN]
END
#PZ=1000
#PD=1000
PROMPT[TX="ENTER TOOL CHANGE POINT (Z):",VN="PZ",PT=1]
PROMPT[TX="ENTER TOOL CHANGE POINT (X OR D):",VN="PD",PT=1]
update[]
execute[]
POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
        select[CreateMenuAction]
        select[CreateMenu1]
        select[GeoPointBIN]
END
#DOC=10000
PROMPT[TX="ENTER DEPTH OF CUT:",VN="DOC",PT=1]
update[]
execute[]
LINEAR_RGH[TY=0, ES="1", EE="2", XS=0.1, YS=1.6, XF=0, YF=0,
   DP=#DOC, CL=0, AN=180, AP=0, CP=0, OP=0]
// Macro default UI script for the Linear_Rgh command.
open[LrufUcut]
BEGIN
        select[ProcessMenuAction]
        select[ProcessMenu4]
        select[LatheRoughUcutBIN]
END
update[]
execute[]
CONTOUR_RGH[ES="6", EE="8", XS=-2.3, YS=1.1, XF=0, YF=0, DP=#DOC,
   CL=0]
// Macro default UI script for the Contour_Rgh command.
open[LrufCptrn]
BEGIN
        select[ProcessMenuAction]
        select[ProcessMenu4]
        select[LatheRoughCptrnBIN]
END
update[]
execute[]

POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
        select[CreateMenuAction]
        select[CreateMenu1]
        select[GeoPointBIN]
END
update[]
execute[]
WITH_TOOL[TL=1, WP="ZX_PLANE", OF=2, LV=0, CL=0, PT=?]
```

```
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
        select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
WITH_TOOL[TL=5, WP="ZX_PLANE", OF=2, LV=0, CL=0, PT=?]
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
        select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
        select[CreateMenuAction]
        select[CreateMenu1]
        select[GeoPointBIN]
END
update[]
execute[]
CONTOUR_RGH[ES="3", EE="5", XS=-3.7, YS=1.1, XF=0, YF=0, DP=#DOC,
   CL=0]

// Macro default UI script for the Contour_Rgh command.
open[LrufCptrn]
BEGIN
        select[ProcessMenuAction]
        select[ProcessMenu4]
        select[LatheRoughCptrnBIN]
END
update[]
execute[]

POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
        select[CreateMenuAction]
        select[CreateMenu1]
        select[GeoPointBIN]
END
update[]
execute[]
WITH_TOOL[TL=3, WP="ZX_PLANE", OF=2, LV=0, CL=0, PT=?]
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
```

```
      select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
      select[CreateMenuAction]
      select[CreateMenu1]
      select[GeoPointBIN]
END
update[]
execute[]
POINT[XE=-5.5, YE=1.6, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
      select[CreateMenuAction]
      select[CreateMenu1]
      select[GeoPointBIN]
END
update[]
execute[]
LINE[XS=-5.5, YS=1.6, ZS=0, XE=-5.5, YE=0, ZE=0, ZI=0, SE=2,
  SS=2, PK=0]
// Macro default UI script for the LINE command.
open[GeometryLine]
BEGIN
      select[CreateMenuAction]
      select[CreateMenu1]
      select[GeoLineBIN]
END
update[]
execute[]
POINT[XE=-5.5, YE=1.6, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
      select[CreateMenuAction]
      select[CreateMenu1]
      select[GeoPointBIN]
END
update[]
execute[]
POINT[XE=#PZ, YE=#PD, LV=0, CO=0, SE=2]
// Macro default UI script for the POINT command.
open[GeometryPoint]
BEGIN
      select[CreateMenuAction]
      select[CreateMenu1]
      select[GeoPointBIN]
END
```

Use of the SAMPM.mcl macro needs to run Milling application in SmartCAM.

```
// Mill32 V7.00 11/19/91 Macro File
// C:\CCSYS\EXAMPLE\SAMPM.mcl
// CREATED: 02/08/1994
READ[FN="c:\\CCSYS\\EXAMPLE\\SAMPM2", FT=1]
// Macro default UI script for the READ command.
open[FileRead]                  // Is "FileRead" up already?
BEGIN                           // If not, do this block
        select[FileMenuAction] // click on "FileMenuAction"
        select[FileMenu2]               // ... "FileMenu2"
END
update[]                // update parameters (er, values) in panel
close[FileRead] // DIALOGS ONLY! Drops the dialog..
execute[]                       // do the action itself.
WITH_TOOL[TL=2, WP="XY_PLANE", OF=2, LV=0, CL=0.1, PT=?]
// Macro default UI script for the WITH_TOOL command.
open[InsertControl]
BEGIN
        select[XC1]
END
select[InsertWithToolBIN]
update[]
execute[]
ELMT_SEQ[BA=1, EL="21", ME=0]
// Macro default UI script for the ELMT_SEQ command.
open[InsertControl]
BEGIN
        select[XC1]
        select[ElmtSeqBIN]
END
update[]
execute[]
ELMT_SEQ[BA=1, EL="18", ME=0]
// Macro default UI script for the ELMT_SEQ command.
open[InsertControl]
BEGIN
        select[XC1]
        select[ElmtSeqBIN]
END
update[]
execute[]
#DOC=100000
PROMPT[TX="ENTER THE DEPTH OF CUT:",VN="DOC",PT=1]
UPDATE[]
EXECUTE[]
POCKET[TY=1, EL="14", FA=0, WC=0.1875, AN=0, XS=0.5+(0.375/2),
   YS=0, AR=45, DS=-#DOC, DC=#DOC, DE=-0.5, EP=0, IN=0, IP=0]
// Macro default UI script for the POCKET command.
//
open[TDRghPocket]
BEGIN
        select[ProcessMenuAction]
```

```
        select[ProcessMenu1]
        select[TDRghPckBIN]
END
update[]
execute[]
```

APPENDIX C. Recommendation Output

```
Template File: c:\ccsys\turn\ac.tmp
Machine Power: 4 HP
Machine Efficiency: .8
Rapid Traverse Feed: 26.5 ipm
Maximum Speed: 2000 rpm
Maximum Feed: 20 ipm
Maximum Depth of Cut: .5 inch
Operation Cost: $25/hr
Load/Unload Time: 1.5 min
Tool Changing Time: 3.3min
Tool Reindex Time : .25min
Machine Setup Time:15min
Lot Size per batch: 50piece
Work Material Name: Aluminum alloys ;  Code: alum
Work Material Hardness: 30-150 BHN
Workpiece original diameter: 3"
Number of tools used in process: 3
=========================Tool Information============================
Tool     Tool    Tool     Code     Diameter Edges  Resharpen  Cost
Number   Type   Material  (AISI)   (inch)    (#)    (times)    ($)
_____   _____  ________  ______   _______  _____  _________  ____
  1      Turn   Carbide    C3       1        4        1        25
  3   Face Grv  Carbide    C2       .125     1        1        19
  5      Turn   Carbide    C3       1        4        1        25
=====================================================================
Tool number: 1; n= .2; C= 3850 ;Pu= .25HP/cubic in/min
Index  D.O.C(in)    Speed(rpm)   Feed(ipm)   HPactual(HP)  ToolLife(min)
  1      .04          2546        25.46          3.0           26.4
  2      .15          2292        45.84         20.3           44.8
  3      .3           1528        61.12         54.0          339.9
  4      .625         1273       101.84        187.5          845.9
Minimum cost Tool life (Te) is: 73.2 min; Speed (Ve) is: 2077 rpm
Maximum productive Tool life (Tm) is: 13.2 min; Speed (Vm) is: 2926rpm
Suggested D.O.C. is: 0.054 inch;  Speed is: 2000 rpm; Feed is: 20 ipm
Suggested Tool Life:  88.4 min
--------------------------------------------------------------------
Tool number: 3; n= .288; C= 2390 ;Pu= .25HP/cubic in/min
Index  D.O.C(in)    Speed(rpm)   Feed(ipm)   HPactual(HP)  ToolLife(min)
  1     1            1019         3.057         1.1           44.7
  2     0               0         0             0.0            0.0
  3     0               0         0             0.0            0.0
  4     0               0         0             0.0            0.0
Minimum cost Tool life (Te) is: 120.8917 min; Speed (Ve) is: 765 rpm
Maximum productive Tool life (Tm) is: 8.15 min; Speed (Vm) is:1663 rpm
Suggested D.O.C. is: N/A inch;  Speed is: 1019 rpm; Feed is: 3.057 ipm
Suggested Tool Life: 44.7 min
--------------------------------------------------------------------
Tool number: 5; n= .2; C= 3850 ;Pu= .25HP/cubic in/min
Index  D.O.C(in)    Speed(rpm)   Feed(ipm)   HPactual(HP)  ToolLife(min)
  1      .04          2546        25.46          3.0           26.4
  2      .15          2292        45.84         20.3           44.8
  3      .3           1528        61.12         54.0          339.9
  4      .625         1273       101.84        187.5          845.9
Minimum cost Tool life (Te) is: 73.2 min; Speed (Ve) is: 2077 rpm
Maximum productive Tool life (Tm) is: 13.2 min; Speed (Vm) is: 2926rpm
Suggested D.O.C. is: 0.054 inch;  Speed is: 2000 rpm; Feed is: 20 ipm
Suggested Tool Life: 88.4 min
--------------------------------------------------------------------
```

APPENDIX D. Evaluation Summary Output

======================Original SmartCAM Result========================

Tool#	Air-Time (min)	Cut-Time (min)	Reindex Time (min)	Tool Time (min)	Tool Life (min)	Tool_change frequency (part)	Tool Dscrptn
1	2.44	3.31	0.25	6.00	88.4	27	0.0 RADIUS TURN
5	0.99	1.33	0.25	2.57	88.4	66	0.0 RADIUS TURN
1	1.06	1.71	0.25	3.01	88.4	52	0.0 RADIUS TURN
3	0.27	0.51	0.25	1.03	44.7	88	.12 WIDE FACE GRV

The estimated Cycle time : 12.61 min

==================Process Information Summary=======================

Process File : c:\ccsys\turn\ac.prc
Template File: c:\ccsys\turn\ac.tmp
Machine Power: 4 HP
Machine Efficiency: .8
Rapid Traverse Feed: 26.5 ipm
Maximum Speed: 2000 rpm
Maximum Feed: 20 ipm
Maximum Depth of Cut: .5 inch
Operation Cost: $25/hr
Load/Unload Time: 1.5 min
Tool changing Time: 3.3 min
Tool reindex Time: .25 min
Machine setup Time: 15 min
Lot Size per batch: 50 piece
Work Material Name: Aluminum alloys ; Code: alum
Work Material Hardness: 30-150 BHN
Workpiece original diameter: 3"
Number of tools used in process: 3

========================Tool Information============================

Tool Number	Tool Type	Tool Material	Code (AISI)	Diameter (inch)	Edges (#)	Resharpen (times)	Cost ($)
1	Turn	Carbide tool	C3	1	4	1	25
3	Face Grv	Carbide tool	C2	.125	1	1	19
5	Turn	Carbide tool	C3	1	4	1	25

================== Revised Simulation Result =======================

Tool#	Air-Time (min)	Cut-Time (min)	Tool Time (min)	Tool Life (min)	Tool_change frequency (part)	Tool Dscrptn
1	4.00	5.02	9.02	88.4	18	0.0 RADIUS TURN
5	1.24	1.33	2.57	88.4	66	0.0 RADIUS TURN
3	0.52	0.51	1.03	44.7	88	.12 WIDE FACE GRV

The estimated Cycle time : 12.61 min

==================Evaluation Report for Turning=====================

Tool#	Rapid Trav Cost($)	Feed Cost($)	Toolchange Cost($)	Tool Depr Cost($)	Sub Total Cost($)
1	1.6676	2.0903	0.0780	0.3547	4.1906
5	0.5155	0.5540	0.0207	0.0940	1.1841
3	0.2158	0.2113	0.0156	0.2155	0.6582
Sub Total:	2.3990	2.8555	0.1143	0.6642	6.0330

The Load/Unload cost :0.6250 $/pc
The Setup cost is :0.1250 $/pc
The Total Processing cost per piece : 6.7830 $/pc

===

APPENDIX E. NC Codes Generated by SmartCAM

Turning NC code

```
 N` G`   X `     Z `   F`   H
  0 92  3200   1800
  1 94
  2M03
  3M06     0      0  T0
  4 00  3200    100
  5 00  3200    100
  6 84  2956 - 5000  89    0
  7 84  2712 - 5000  89    0
  8 84  2468 - 5000  89    0
  9 84  2224 - 5000  89    0
 10 00  3444 - 4878
 11 00  3200    100
 12 00  2200 - 2300
 13 00  2200 - 3042
 14 01  1976 - 3177  89
 15 01  1976 - 4012  89
 16 01  2200 - 4012  89
 17 00  2200 - 2300
 18 00  2200 - 2852
 19 01  1732 - 3133  89
 20 01  1732 - 4134  89
 21 01  2200 - 4134  89
 22 00  2200 - 2300
 23 00  2200 - 2661
 24 01  1488 - 3088  89
 25 01  1488 - 4256  89
 26 01  2200 - 4256  89
 27 00  2200 - 2300
 28 00  2200 - 2471
 29 01  1244 - 3044  89
 30 01  1244 - 4378  89
 31 01  2200 - 4378  89
 32 00  2200 - 2300
 33 00  2000 - 2400
 34 01  1000 - 3000  89
 35 01  1000 - 4500  89
 36 01  2000 - 4500  89
 37 00  2000 - 2300
 38 00  2200 - 2300
 39 00  3200   1800
 40 00  3200   1800
 41 94
 42M05
 43M06 -  50 -  538  T4
 44M00
 45M03
 46 00  2200 - 3700
 47 00  2200 - 2958
 48 01  1976 - 2823  89
 49 01  1976 - 1988  89
 50 01  2200 - 1988  89
 51 00  2200 - 3700
 52 00  2200 - 3148
 53 01  1732 - 2867  89
 54 01  1732 - 1866  89
 55 01  2200 - 1866  89
 56 00  2200 - 3700
 57 00  2200 - 3339
 58 01  1488 - 2912  89
 59 01  1488 - 1744  89
 60 01  2200 - 1744  89
 61 00  2200 - 3700
 62 00  2200 - 3529
 63 01  1244 - 2956  89
 64 01  1244 - 1622  89
 65 01  2200 - 1622  89
 66 00  2200 - 3700
 67 00  2000 - 3600
 68 01  1000 - 3000  89
 69 01  1000 - 1500  89
 70 01  2000 - 1500  89
 71 00  2000 - 3700
 72 00  2200 - 3700
 73 00  3200   1800
 74 00  3200   1800
 75 94
 76M05
 77M06   453    377  T4
 78M00
 79M03
 80 00  3200 - 5500
 81 00  3200 - 5500
 82 01     0 - 5500  31
 83 00  3200 - 5500
 84 00  3200   1800
 85 00  3200   1800
 86M06    00     00  T4
 87M30
"
```

Milling NC code

```
 N' G'  X  '  Y '   Z ' F
  0 90
  1 92     0    0  2000
  2M06D    0S2000      0T 02
  3M03
  4 00   488    0  2000
  5 00   488    0     0
  6 00   488- 425-  500
  7 01  1013- 425-  500 120
  8 01  1013- 237-  500 120
  9 01   488- 237-  500 120
 10 01   488-  50-  500 120
 11 01  1013-  50-  500 120
 12 01  1013  138-  500 120
 13 01   488  138-  500 120
 14 01   488  325-  500 120
 15 01  1013  325-  500 120
 16 01  1013  513-  500 120
 17 01   488  513-  500 120
 18 00   488    0-  500
 19 00   488    0  2000
 20 00     0    0  2000
 21 00     0    0  2000
 22M06D    0S   0      0T 00
 23M30
"I
```