AN ABSTRACT OF THE THESIS OF

Hsueh-Jen Chen for the degree of Master of Science in Industrial Engineering presented on April 21, 1992.

Title: Design and Implementation of a System for Integrating Material and Process Selection in Automated Manufacturing

Redacted for Privacy

Abstract approved: _

Sabah Randhawa

Today's manufacturing environment is characterized by competition and continuous change in product and process requirements. The concept of "design for manufacturability" integrates product specifications with manufacturing capabilities by considering the design and manufacturing phases as an integrated system, evaluating the combined system during the design phase of a product , and adjusting the design for maximum efficiency and production economics.

This research focuses on one aspect of design for manufacturability, that of process technology evaluation for a specified product design. The objective of the proposed system developed in this study is to evaluate technology alternatives for manufacturing a specified part design and to identify the best combination of product-process characteristics that would minimize production costs within the constraints set by the product's functional requirements and available processing technology.

The research objectives are accomplished by developing a simulation-

based analysis system. The user inputs product specifications through structural screens. The system maintains data bases of work and tool materials, and machining operations. Based on user input, the system then extracts appropriate information from these data bases, and analyzes of the production system in terms of production economics, and other operational measures such as throughput times and work-in-process inventories. Sensitivity analysis may then be performed to explore tradeoffs in design and production parameters. The system is completely integrated, and a user with no prior experience of either simulation or data base technology can use the system effectively.

Design and Implementation of a System for Integrating Material and Process Selection in Automated Manufacturing

by

Hsueh-Jen Chen

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed April 21, 1992

Commencement June 1992

APPROVED:

Redacted for Privacy

Professor of Industrial Engineering in charge of major

Redacted for Privacy

Head of Department of Industrial Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 21, 1992

Typed by Hsueh-Jen Chen for Hsueh-Jen Chen

ACKNOWLEDGEMENTS

I would like to express my thanks to individuals whom I consulted during the course of this study. Special thanks are due to Dr. Sabah Randhawa, my major professor, for his professional counsel as well as encouragement throughout this project. I also want to thank my family members for their continuous support and patience.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

<u>Table</u>

Design and Implementation of a System for Integrating Material and Process Selection in Automated Manufacturing

1. Introduction

There is a growing concern that U.S. Manufacturing is no longer competitive with many other industrialized nations (Whitney et al, 1988). The half life of many products has decreased to the point that 50 percent of their sales occur within the first three years (Meredith and Hill, 1987). This has resulted in continuous design and manufacture of new products, and in the need of flexible manufacturing systems that are economic at low volumes. Economic survival in this environment requires complete integration of all engineering and manufacturing functions, particularly design and production.

The primary objective of the design-manufacturing process is to consider manufacturing issues early enough to shorten product development time and time-to-market, and to reduce the costs resulting from segregation of design and production.

1.1 Concept of "Design for Manufacturability"

Specifically, design for manufacturability focuses on the following issues (Veilleux and Petro, 1988):

1. Understanding how the process by which a product is designed interacts with other components of the manufacturing system and using this understanding to design better quality products that can be marketed more quickly.
2. Understanding how the physical design of the product itself interacts with the components of the manufacturing system and using this understanding to define product design alternatives

that help facilitate "global" optimization of the manufacturing system as a whole.

This concept places emphasis upon the organizational and procedural issues in integration of product and process design. This is because of the complexity of the product specifications that are required and desired, and because of the complexity of the manufacturing systems that are needed to produce these products. The challenge of actually integrating product and process selection within the constraints imposed by the organizational structure and procedural processes requires input from as many design and manufacturing activities as possible. This step should be as early in the design process as possible, to maximize the quality of early design decisions and minimize the amount of engineering changes introduced in the product's life cycle.

Concerns from the concept of design for manufacturability imply cost reduction, quality, and productivity. Product designs that are based on design concepts selected for their inherent ease of manufacture, composed of components designed to promote ease of fixturing, handling, and assembling, and carefully matched process and material selection to the available manufacturing technology naturally result in productivity improvement. Along with lower cost and enhanced productivity comes quality improvements. The ability to consider manufacturing early in design phase means less quality risks and deviations from design intent.

The concept of design for manufacturability advocates flexibility in the new manufacturing system dimensions which significantly affect the design process. The availability of appropriate information bases, together with electronic computing capabilities increases flexibility in manufacturing by analyzing the production system to changes in product or production conditions and/or requirements.

Design flexibility implies more than the ability to manufacture to customer order in any sequence and lot sizes. It also implies the ability to maintain product and process options over time. Maintaining product and process options requires that the engineer and designer look forward in time to anticipate what changes are likely to occur or be required, and then to consciously plan for these changes in the design.

The process-driven design is embedded in the concept of design for manufacturing to ensure product conformance with processing requirement and constraints. In process-driven design, the manufacturing process plan is developed prior to performing the product design. The process-driven product design methodology (model) together with a viable means for electronically transferring design and process planning information between the computer-aided design (CAD) environment in which the product is designed and the computer-integrated manufacturing (CIM) environment in which it is manufactured, assembled, and tested is essential to the promise of a flexible computer-based manufacturing.

1.2 Importance of Integrating Material and Process in Design Phase

Traditionally, the design-manufacturing process consists of three rather independent components -- conception, design, and production (West, Randhawa, and Brings, 1989). The product is conceptualized based on market analysis. The design engineers produce product specification based primarily on input from sales and marketing as to what is desired on the market place without any consideration of the available processes. On the other hand, production engineers often spend inordinate amount of resources handling product tolerances and quality problems when a minor change in the product design specifications could have eliminated a number of these problems.

The importance of integrating design and manufacturing becomes apparent on observing the costs associated with a product's life cycle (Figure 1.1). The changes in the design phase are easier to handle and less costly to implement. The design phase determines 80 percent of the cost of the product (Anderson, 1990). This is because at this stage production specifications are not fixed, fewer people are involved, and hardware and production constraints have not been definitely specified. In the later stages of the product, change costs are high because of time delays and larger number of personnel and manufacturing components involved.

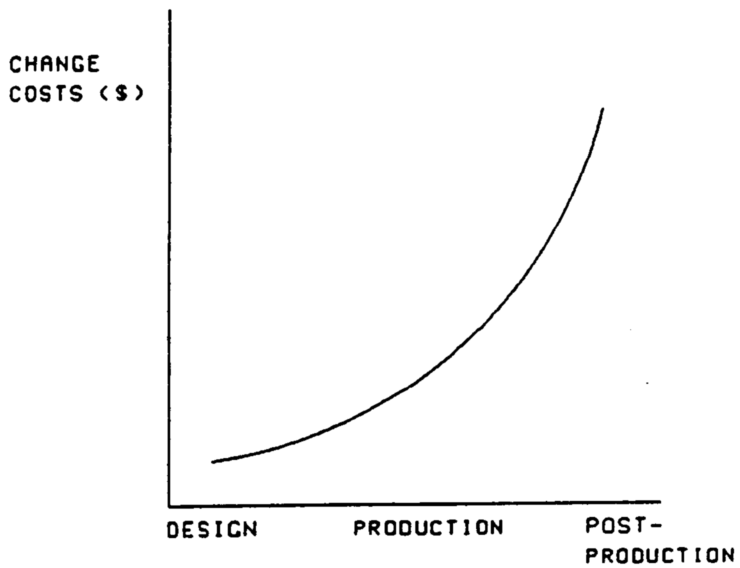

Figure 1.1: Variation in Change Costs over Product's Life Cycle (Reference: West, Randhawa, and Brings, 1989)

1.3 Research Objectives

The complete design-manufacturing process (Figure 1.2) is complex, involving numerous considerations such as market analysis, concept design, material and process selection, optimization, information and process control, costing, and manufacturing. The problem definition phase in Figure 1.2 defines the need requirements for a product. These requirements are then translated into product specifications. The specifications define the characteristics for 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.

The next phase is the evaluation phase where 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 also indicate reconfiguration of product requirements and specifications. The design and production parameters identified at this stage are used in the physical manufacturing process.

Recently, much has been written about the importance of the "total design" (Anderson, 1990; Ettlie and Stoll, 1990; Pugh, 1990). However, there is a lack of analytical tools available in published literature that can aid in the product design and evaluation processes. The system described in this thesis focuses on the design-manufacturing evaluation phase (Figure 1.1) and on integrating the analysis procedure with appropriate information bases. Development work on this system has included:

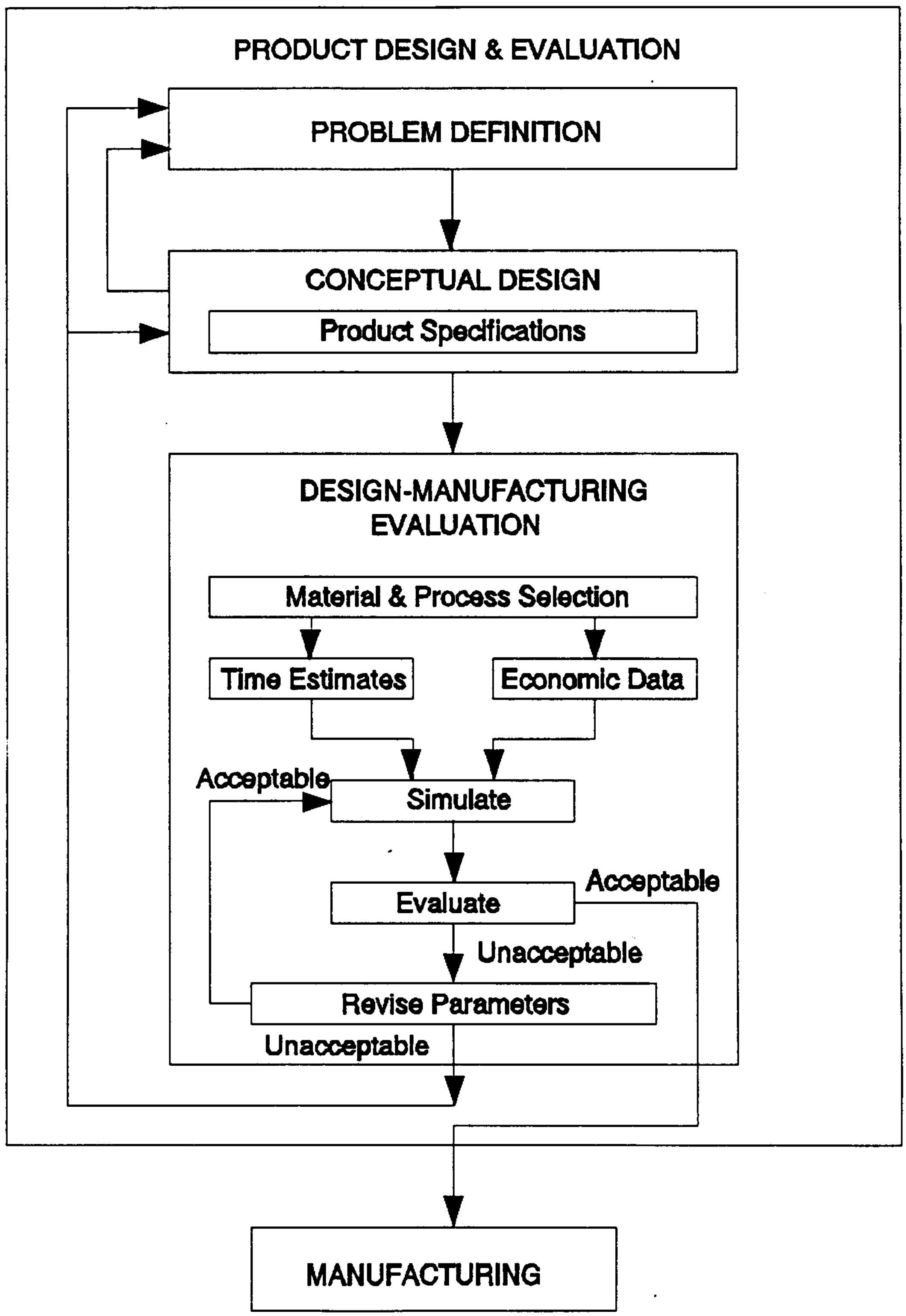

Figure 1.2: Design-Manufacturing Process

1. Creation of material and process databases that characterize information on tool and work materials, and different processes used in machining and metal removal.
2. Modeling and simulation of the manufacturing process for analyzing the system in terms of production economics and technical factors (such as throughput time and work-in-process inventories), and exploring tradeoffs through sensitivity analysis.
3. Development of user interfaces to provide front-end (data input) and back-end (output analysis) system interaction with the user.
4. Control structure for integrating the databases and simulation, and user interface modules.
5. Testing and evaluation of the system.

1.4 Organization of the Thesis

This thesis consists of six chapters. This introductory chapter focused on importance of this research and outlined the research objectives. Chapter two provides a brief introduction to the manufacturing processes being modeled. Chapter three describes the system developed in this research: its components and communication between these components. Chapter four describes the implementation of system components: data design, simulation module, user interfaces, and integration of individual components. Chapter five illustrates system implementation through application examples; this includes description of input data, output interpretation, and sensitivity analysis. The final chapter summarizes this research effort and provides directions for future research.

2. Background

2.1 Computer-Aided Process Planning

Computer-aided process planning (CAPP) refers to an interactive computer system that automates some of the work involved in preparing a process plan (Bedworth, Henderson, and Wolfe, 1991). Among the previous research concerning CAPP methodology, issues related to information processing and decision making have been addressed. A resourceful CAPP system must have access to a tremendous amount of information which includes facts about machines in a manufacturing system, rules about selecting and sequencing machining operations, and all necessary machining parameters. Furthermore, the system should be flexible because facts and rules in the data base require constant updating. As new technology, equipment, and processes become available, effective procedures to manufacture a particular part may also change. Also, integration of an analysis model and the process planning system is considered to be a useful aid in evaluating system performance.

Another growing concern in CAPP system is design data input. Two approaches have been explored. The first approach is direct interface with a CAD (Computer-Aided Design) system (Gupta, 1990). This effort includes the use of commercially available or specially designed 2D/3D CAD packages for preparing detail drawings and for making necessary information available to a computer-aided process planning module. The second approach uses a descriptive language (Gupta, 1990). This scheme uses the designated syntax to describe each individual attribute of the feature. Both dimensional and geometrical relations between any two feature of the part are also described by translating its drawings to functional literal representing these relations. This research uses the second approach for design data input.

2.2 Manufacturing Processes

The focus of this research is discrete parts manufacturing. There are a wide range of processes available in discrete manufacturing. The system developed here focused on the more important processes of turning, milling, drilling, grinding, cutoff, and shaping. For a description of these processes, see DeGarmo, Black, and Kosher, 1988; Doyle et al, 1985; and Lindberg, 1990.

2.2.1 Turning

Turning is the process of machining external cylindrical and conical surfaces. The basic machine tool on which turning operations are performed is the lathe. The turning operation may be divided into two classes: those done with the workpiece between centers and those done with the workpiece chucked or gripped at one end with or without support at the outer end.

2.2.2 Milling

Milling is a basic machining process by which a surface is generated progressively by the removal of chips from a workpiece fed into a rotating cutter in a direction perpendicular to the axis of the cutter. Milling requires a rotating, multi-edged cutting tool plus some mechanical means for guiding a workpiece while in contact with the cutter.

2.2.3 Drilling

Drilling is the most common way of producing a hole in metal or other material. Most of drilling is done with a tool having two cutting edges. These edges are at the end of a relatively flexible tool. Cutting action takes place inside the workpiece. The only exit for the chips is the hole that is filled by the drill. Drilling may be accompanied by other related hole-making processes such as boring and reaming.

2.2.4 Grinding

Grinding is done on surfaces of almost all conceivable shapes and materials of all kinds to provide a surface of desired finish quality. This operation is a high-energy process. Grinding together with other abrasive-finishing processes such as lapping, honing, and super-finishing have changed a strictly metal-finishing operation to a competitive metal-removal method.

2.2.5 Cutoff

By using a tool having a specific form or shape and feeding it inward against the work, external cylindrical, conical and irregular surfaces of limited length can also be turned. The shape of the resulting surface is determined by the shape and size of the cutting tool. Such machining is called form turning. If the tool is fed to the axis of the workpiece, it will be cut in two. This is called parting or cutoff and a simple, thin tool is used.

2.2.6 Shaping

In metal-cutting operations, planes may be generated by a series of straight cuts, without turning the workpiece. If the tool is reciprocated and the workpiece is moved a crosswise increment at each stroke, the operation is called shaping.

2.3 Materials Property Databases

The operational characteristics and economics of discrete parts manufacturing depend on the combination of workpiece material and tooling material. To be effective, a wide range of engineering materials need to be available in the system data base. These might include cast irons, steels, aluminums, nonferrous metals, plastics, rubber and elastomers, ceramics, and composites. This research focuses on five materials specific to the metal working industry. These include three types of carbon steel (low, medium, and high carbon), gray cast iron, and aluminum.

In carbon steels, carbon is the alloying element that essentially controls the properties of the alloys, and in which the amount of manganese cannot exceed 1.65%, and the copper and silicon contents must each be less than 0.60%. Carbon steel materials can be subdivided into those containing between 0.08 and 0.35% carbon, those containing between 0.35 and 0.5% carbon, and those containing more than 0.5% carbon. These are known as low-carbon, medium-carbon, and high-carbon steels, respectively (Doyle et al, 1985).

Low-carbon steels are relatively soft and ductile and cannot be hardened appreciably by heat treatment. It represents the largest tonnage of all steel produced. Cold finishing improves the surface finish, mechanical properties, and machinability of these compositions. Medium-carbon steels can be hardened by heat treatment, but cannot be through-hardened in sections whose thickness is greater than about one half inch. Finally, due to the comparatively higher percentage of carbon in high-carbon steels, the resulting hardness makes the high-carbon steel tools among the most useful general-purpose tools for applications.

Gray iron is one of the most common types of cast irons. Cast irons usually contain from 2.0 to about 4.5% carbon (Doyle et al, 1985). The excess carbon is the basis for many of the good properties of gray iron, such as high fluidity, high damping capacity, low notch sensitivity, and good machinability. Gray cast iron is comparatively soft, of low tensile strength, easily machined, and is stronger than many steels in compression. Strength is not always the major criterion for selection of a material. For instance, gray cast irons of the weaker grades have superior qualities for such applications as resistance to heat shock and dampening of vibrations in machine tool members.

Pure aluminum is known for excellent electrical and thermal conductivity, corrosion resistance, non-toxicity, light reflectivity, low specific gravity, and softness and ductility (Doyle et al, 1985). Aluminum can be hardened by solid-solution hardening, by cold working, and by precipitation hardening. The room-temperature mechanical properties of aluminum alloys are, in general, inferior to those of steel, almost equal to those of copper alloys, and superior to those of magnesium alloys. In addition to mechanical properties, the specific strength shows that most, but not all, aluminum alloys are superior to other steel compositions. Generally, aluminum alloys are considered easily machined, though in some cases good surface finish is difficult to obtain (Doyle et al, 1985).

In all machining processes, success in metal cutting also depends upon the selection of the proper cutting tool (material and geometry) for a given work material. A wide range of cutting tool materials is available with a variety of properties, performance capabilities, and cost. High-speed steels and cemented carbides are currently the most extensively used tool materials. These are currently included in the proposed system.

Compared with tool steel, high-speed steels can operate at about double the cutting speed with equal life, resulting in its name, often abbreviated HSS (DeGarmo, Black, and Kosher, 1988). Currently, high-speed steel is widely used for drills and many types of general-purpose milling cutters and in single-point tools used in general machining.

Cemented carbides are nonferrous alloys manufactured by powder metallurgy techniques. These tool materials are much harder, and chemically more stable; they have better hot hardness, high stiffness, and lower friction; and they operate at higher cutting speed than HSS (DeGarmo, Black, and Kosher, 1988).

This research uses some of the concepts of computer-aided process planning for obtaining design data input from the users, selecting appropriate manufacturing processes and materials, and analyzing the resulting system for operational economics and efficiency.

3. Approach

This chapter describes the model that has been developed in this research for the design-manufacturing evaluation phase. The system components include the functional modules, the input and output components, and the connection interfaces. Since the evaluation framework is based on minimizing the production cost, the economic models used in this research will also be discussed.

The system is designed primarily for flexible manufacturing, and in particular, for metal removal processes. The machining processes currently included in the system are turning, milling, drilling, grinding, cutoff and shaping. A set of these processes performed in sequence remove the unwanted metal from the workpiece so as to obtain a finished product of desired size, shape, and finish. Each process may be considered as an individual work station. The assignment of work stations to establish a production route depends on the product features to be produced.

3.1 Economic Model

The selection of design and production parameters is basically an economic decision. The parameters of machining processes greatly influence production costs in different ways. For example, the cutting speed has higher influence on the tool life compared to feed rate or the depth of cut, although increases of either cutting speed, depth of cut, or feed results in decreased tool life. As cutting speed is increased, the machining time decreases but tools wear out faster and must be changed more often. Therefore, to minimize the total production cost, these machining parameters need to be balanced.

The production cost can be decomposed into four basic components: machining costs, tool costs, tool changing costs, and the workpiece handling costs. The total processing costs for any machining process is the sum of these four components. Variations in these cost components as a function of cutting speed are shown in Figure 3.1. As can be seen from Figure 3.1, the minimum production costs represent a balance among the four cost components.

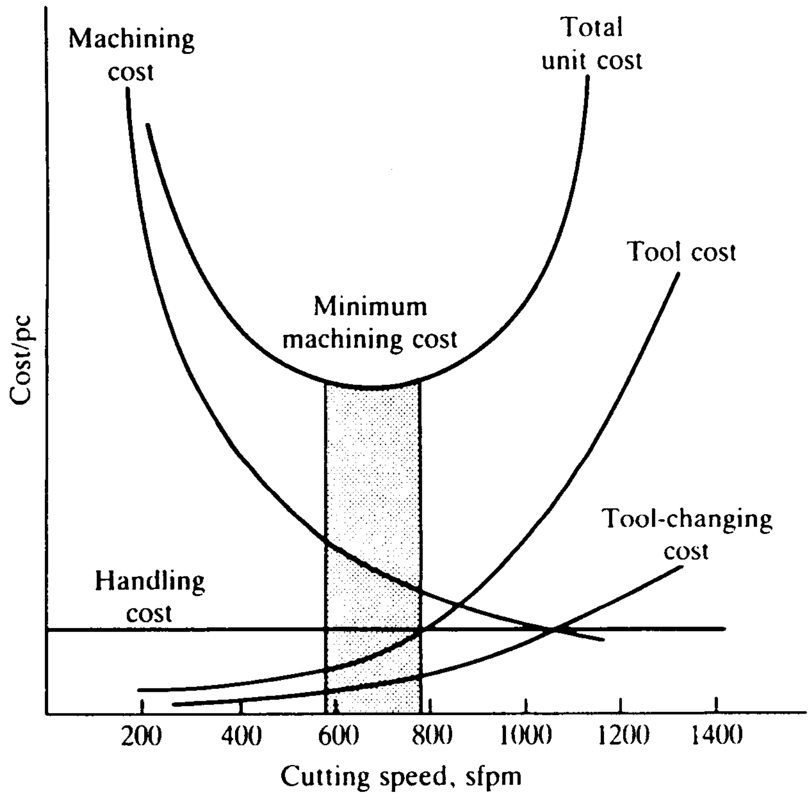

Figure 3.1: Cost Components as a Function of Cutting Speed (Reference: Lindberg, 1990)

The individual cost components, on per piece basis, are defined as follows (DeGarmo, Black, and Kosher, 1988):

Machining costs	=	$(C_0\ t)$
Tooling costs	=	$(C_t\ t)\ /\ T$
Tool changing costs	=	$(C_0\ t_c\ t)\ /\ T$
Handling costs	=	$(C_0\ t_l)$

where C_0 = operating labor cost,

C_t = initial tool cost,

t = cutting time per piece,

t_c = time to change tool or reindexing time,

t_l = workpiece load and unload time, and

T = tool life, given by Taylor's tool life equation.

The machining cost depends on machine cutting time (min/piece). The value for C_0, the operating labor cost, includes both the direct labor and associated overhead expense.

Besides the cutting time, the other factor that needs to be computed for computing the processing costs is tool life (T). Tools undergo gradual wear over time due to friction between tool surface and workpiece surface. Different models for tool wear have been proposed. The relationship for computing tool life here is given by (Lindberg, 1990):

$$VT^n = C$$

where n and C are constants whose values depend on tool and workpiece materials, and are given in a number of references (Lindberg, 1990).

Each time when the machine tool needs to be changed or reindexed, tooling and tool changing costs are encountered. These two costs can be figured from initial tool cost per piece, and the operating expenses for the time required to change tools. However, these costs are for a group of products rather than for each product. In order to allocate these costs per work piece, tool life information is required. In other words, proportion of tool life used for each work piece (machining cutting time versus tool life) must be included in estimating tooling and tool changing costs. The initial tool cost (C_t) is based on three factors: purchased price for an individual tool piece, resharpening expense, and average times of resharpening before discarding the tool piece. To compute tool changing cost to change or reindex a tool, only the operation cost (C_0) and the time used to change tool (t_o) are required.

The handling cost represents the operating costs while the part is being loaded and unloaded. It is proportional to workpiece loading and unloading time, and is independent of cutting speed.

In any machining operation, metal cutting parameters such as cutting speed, feed rate, and metal removal rate, are the major independent variables that determine machining costs by influencing cutting time and tool life. Other independent variables are work and tool materials, and geometry of cutting tool. Expressions for computing metal cutting parameters of all five machining operations modeled in this system are given in Table 3.1.

	Process				
Parameters	**Turning**	**Milling**	**Drilling**	**Shaping**	**Grinding**
Cutting Speed, V ft/min	$\pi \times D \times N$	$\pi \times d \times n$	$\pi \times d \times n$	Specified	$\pi \times d \times N$
Feed Rate, F in/min	$f_r \times N$	$f_t \times n_t \times n$	$f_r \times n$	$f_s \times n_s$	$f_r \times N$
Cutting Time, t min	$\frac{L}{F}$	$\frac{L}{F}$	$\frac{L}{F}$	$\frac{L}{F}$	$\frac{L}{F}$
Metal Removal Rate, Q in^3/min	$\pi \times D \times d_c \times F$	$w \times d_c \times F$	$\frac{\pi}{4} \times d^2 \times F$	$L \times d_c \times F$	$w \times d_c \times F$
Horsepower (spindle), HP_s	$Q \times HP_{unit}$	$Q \times HP_{unit}$	$Q \times HP_{unit}$	$Q \times HP_{unit}$	$Q \times HP_{unit}$
Horsepower (motor), HP_m	$\frac{HP_s}{E}$	$\frac{HP_s}{E}$	$\frac{HP_s}{E}$	$\frac{HP_s}{E}$	$\frac{HP_s}{E}$

Table 3.1: Machining Parameters for Different Processes

Symbol	Parameter
D	Part diameter (Ft)
d	Tool diameter (ft)
F	Feed rate (inch/min)
f_r	Feed rate (inch/rev)
F_t	Feed rate (inch/tooth)
f_s	Feed rate (inch/stroke)
N	RPM (rev/min)
n_t	Number of teeth per revolution
n_s	Strokes per minute
L	Length of cut (inch)
V	Cutting speed (ft/min)
t	Cutting time (second)
W	Width of cut or part (inch)
Q	Metal removal rate ($inch^3$/min)
d_o	Depth of cut (inch)
w	Width of cut, cutter, length of cutter diameter (inch)
HP_s	Horsepower (spindle)
HP_m	Horsepower (motor)
HP_{unit}	Unit of horsepower (hp/$inch^3$/min)
E	Motor efficiency
n	Tool rpm (rev/min)

Table 3.1 (Continued): Machining Parameters for Different Processes

The depth of cut, feed, and cutting speed are machine settings that must be established in any metal-cutting operation. They all affect the forces, the power, and the rate of metal removal. They can be defined by comparing them to the needle and record of a phonograph. The cutting speed (V) is represented by the velocity of the record surface relative to the needle in the tone arm at any instant. Feed is represented by the advance of the needle radially inward per revolution, or is the difference in position between two adjacent grooves. The depth of cut is the penetration of the needle into the record or the depth of the grooves. A workpiece or cutter must be revolved at the number of revolutions per minute (rpm), designated by N, that will give the required surface speed. If the tool diameter is d, in inches, then in one revolution a point on the periphery travels a distance of (πd)/12 ft. The rate of rotation is then given by (Doyle et al, 1985):

$$N=\frac{V}{\pi d/12}$$

Material cut is an important factor in determining tool forces and power. Ductility, hardness (or strength), coefficient of friction, and work hardening ability all have definite effects on the tool forces. Higher hardness causes higher forces and power. The coefficient of friction may be reduced by additives such as sulfur or lead, resulting in a lowering of forces. Materials that harden considerably, such as austenitic steels, require higher forces and unit power.

It is also important to be able to calculate the total cutting horsepower required. The power required at the tool point is calculated as the product of the unit horsepower and metal removal rate.

The unit horsepower for various materials is based on empirical results. The cutting horsepower can then be computed from (Niebel, Draper, and Wysk, 1989):

$$HP_s = HP_{unit} \cdot Q$$

Machine tools inherently contain inefficiency resulting from friction and other factors. The gross motor horsepower (HP_m) is, therefore, given by (Niebel, Draper, and Wysk, 1989):

$$HP_m = \frac{HP_s}{E}$$

where E represents the efficiency of the machine tool.

In general, the metal removal rate is calculated as the product of the area machined times the feed rate perpendicular to area machined. For all the machine processes listed in Table 3.1, the time to machine is calculated by dividing the distance to be traveled by the tool by the feed rate. The distance traveled in drilling process is the thickness of the workpiece plus an allowance owing to the drill angle. An allowance of 1/2D is typically used, since a drill angle of 118 degree is common. For turning, shaping, and grinding, the length to be traversed is increased by adding an allowance A, in addition to the length of workpiece, since the tool must start and stop at some distance from the workpiece to account for the cutting tool. In milling, the calculation of allowance A depends on the type of milling operation.

3.2 Evaluation Framework

The essential modules along with the information flows for the system are shown in Figure 3.2. The boxes in Figure 3.2 represent either a functional or an information object. The arcs represent the connections between objects; arrows show direction of flow. Information along each arc is identified in the figure.

There are two input types, technical and informative. Technical input represents the information base and supplies the system with up-to-date data. This type of input includes the manufacturing materials and current processing technology. Input and modification of technical information requires understanding of the specific knowledge and the relationships between its components. Also, the authority to access and protect information is required to ensure the accuracy of system performance. Manipulating technical information requires appropriate data base tools.

Compared with the technical input, the informative input is just the specification of the product to be produced. The dimensions of the product, the material properties for the features to be produced, and the characteristics of the processes for the product are the major elements of this specification. For informative input, a structured and user-friendly interface is required to aid the user in interacting with the system.

The evaluation process is iterative in nature involving multiple evaluations and adjustments. The central control unit of the system is

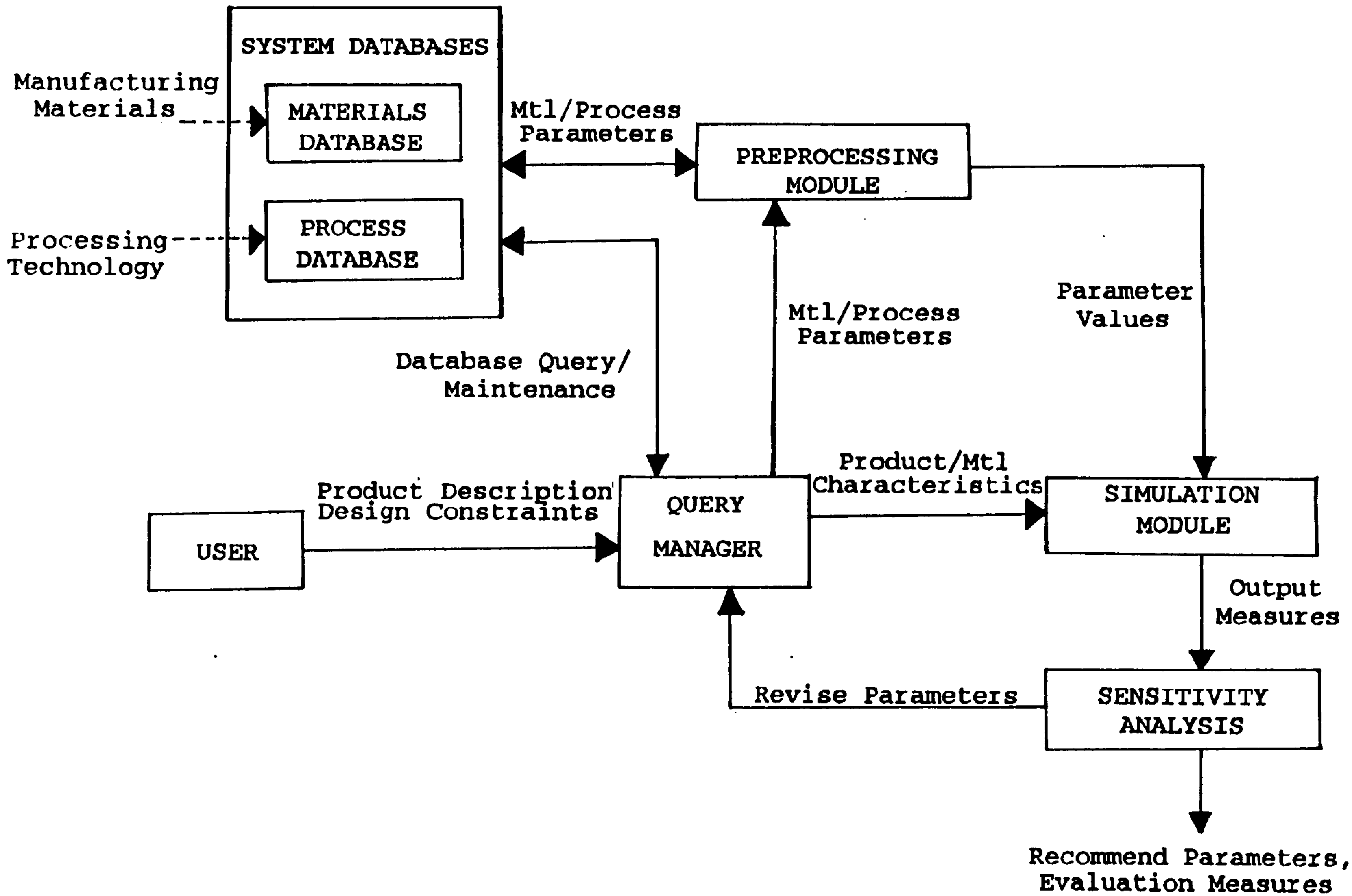

Figure 3.2: Design-Manufacturing Framework

the query manager. This controls the overall evaluation process and also keeps the connections to the functional modules which perform the basic match, selection, and logical inductions to support central decision-making. The access of the information databases is also through the central control unit.

The analysis is based on a simulation model built for the specified manufacturing process. The system outputs consist of final reports and on-line displays. The output consists of estimates of operational performance measures (such as production times and inventories) and production costs.

3.3 System Components

The specific components in the system are the system databases, the query manager, and two major functional modules which are the preprocessing and simulation modules. Each component is briefly described below based on its functions, major characteristics, as well as the position relative to other components in the system framework.

3.3.1 Material Databases

The material databases are categorized into two primary groups, tool material databases and the part material databases. Each group of data has identical attributes for the material's physical properties. The attributes generally belong to one of the three essential types: material identification, informative descriptions, and numeric parameters. The identification and description of the material are used as the definition and conditions related to the general categories specified by the American Society for Metals. The third type of attributes store the material-related constants required to estimate secondary set of material parameters. Examples include Taylor's constants for computing tool life in the tool material database and the material removal rates classified by unit horse power in the part material database.

The database information may be updated for maintenance by the central control interface and used for query selections and reviews. The other interaction involving material databases occurs during the construction of the simulation model, where mapping to the specified manufacturing process is accomplished.

3.3.2 Process Databases

There is one database for each process included in the system. The structure of a process database depends on the characteristics of that process. Since different processes may have different requirements, database structure may also differ. However, there are some common attributes such as kinds of tool material and part material to be used for the specific process. The other essential parameters for each process depend on the minimum elements required to fully characterize this process. Examples of such parameters include cutter diameter, depth of cut, and/or feed rate.

The process databases have the same relationships with other components as the material databases, and the functional impact is similar. The additional effect involved with process databases is caused by two-way information flow due to the creation of new entries that are appended to the process databases in order to complement the product features and process environment. Specific cases may arise, as for example, in drilling operation, where it is required to consider the possibility of changing cutting speed or feed rate for specific diameter drilling.

3.3.3 Query Manager

The query manager is the front-end to the entire system. This front-end performs two basic functions, input and editing, and an output function. As an input and editing tool, the query manager is used to obtain the product specifications from the user. The features to store and retrieve information as well as to move around the data bases is essential to the flexibility of the system. Once the connections to all the supplementary modules are complete and simulation executed, the query manager directly supports the viewing of output and results of sensitivity analysis.

For the input of product specification, a set of process-independent screens are provided for the user. The other information required for the design phase such as manufacturing constraints (for example, the allocated resources for manufacturing) and product constraints (for example, production throughput) can also be specified through this front-end.

The query manager is positioned as a central control unit in the evaluation framework. It directly connects all major modules and manages the evaluation process cycle.

3.3.4 Preprocessing Module

The preprocessing module handles the coordination between design and manufacturing phases and the calculation of some of the process parameters. Strategies for material transformation include built-in criteria for each process addressing specific concerns of each manufacturing process. Parameters like tool life and metal removal rate are calculated by using attribute values extracted from the material and process databases. The primary objective of the preprocessing module is to estimate some of the parameters used in the simulation process to reduce the simulation time for complex processes.

3.3.5 Simulation Module

The simulation module analyzes the processing sequence for production economics and system dynamics. An analysis of the initial system specifications provides the users with a baseline performance model. Improvements in the system may be made by changing product or process specification, as for example, tolerances, tooling material or workpiece material as long as product requirements are satisfied within the range specified. The flexibility that a generalized simulation model can be used for evaluating a wide range of alternatives is another important characteristics of the system developed in this research.

The simulation module receives the transformed parameters for the manufacturing process from the preprocessing module. Combined with the specified request for simulation runs from the user, the module can then be used to perform sensitivity analysis in order to aid the user in exploring tradeoffs between product design and process operations. The basic information from simulation runs is output system economics and performance.

4. Implementation

The key to the integration of the design-manufacturing system is the ability to easily maintain information of material and process databases, to link different modules and to transfer information among the different modules efficiently. This chapter discusses the implementation tool, basic structure, and the content (or examples) of each component.

The system has been implemented for a microcomputer environment. The basic hardware requirement is INTEL 80286-based or higher IBM compatible microcomputer. The software modules are developed using two commercial packages, PARADOX 3.0 (Paradox 3.0, 1988) and SIMAN (Pedgen, Shannon, and Sadowski, 1990). The application programs are developed to take advantage of the PARADOX's excellent features in handling data processing, and SIMAN's ability to simulate the operational behavior of discrete automated manufacturing processes. All the connections between the two software modules are manipulated under the DOS environment.

4.1 Database Design

This section discusses the implementation and structure of databases included in the system.

4.1.1 Implementation Tool

Developing a conceptual and logical database model is a complex process. The organization of data information is complex. An information management tool, such as a database system, is designed to conduct such complicated tasks. The data information is stored according to the file structure of the design tool. The change of design can be made independent of the information contents. PARADOX 3.0 from Borland International is an information management software package.

The structured format in PARADOX provides easy and efficient retrieval capability for large and complex databases. The relational database system helps inquiry of large database tables; the relational structure also reduces the usage of space for data information by minimizing redundancy among data items. Generally, the format of database reflects the characteristics of the information; however, the design of the database influences the performance of data management.

4.1.2 Organization and Structure of Database

The database system controls two groups of data information: material information and process information. The organization of each type of information is different due to differences in attributes; consequently, the structure of each database is different.

4.1.2.1 Materials Information

The material information includes properties of materials used for manufacturing the desired products. Based on user specification, properties of selected material are extracted from this database, and analyzed in conjunction with the process characteristics (from the process database) to determine the most economic material-process combination for the desired product in the simulation phase.

There are generally two material types that need to be specified in a production operation - work material and tool material. The processing time, feed rates, and other operation characteristics depend on both these parameters. Work materials currently included in the database are low carbon steel, medium carbon steel, high carbon steel, gray cast iron, and aluminum. There are a number of cutting tools available, but high speed steel and sintered carbide do the bulk of metal cutting; these two are included in the tool material database.

Material properties appear in tabulated form in handbooks published by technical societies such as SME, ASM, ASME, and SAE. The properties included in the materials database are taken from these publications (Metals Handbook, 1989, Machining Data Handbook, 1980, and Pollack, 1976).

4.1.2.2 Process Information

This database information details the processes that may be required for the creation of product features. Information includes specific process parameters such as restrictions on equipment capabilities and shapes that each operation is capable of producing.

The parameters stored in the database are different for each operation, being a function of the specific operation. For example, the primary specifications in turning are the depth of cut, cutting speed, and feed rate; in contrast, the primary considerations in drilling are the feed rate, tool peripheral speed, and drill diameter. In accordance, a separate data structure is defined for each operation, and values of individual parameters are stored for different combinations of the work and tool materials specified.

Some temporary databases may also be created during system execution.

Examples include processing sequence and input costs. However, these data bases primarily aid in analyzing the production of a specific part.

4.1.3 Database Contents

The primary components of each data base are briefly defined in Tables 4.1, 4.2, and 4.3. Table 4.1 shows the structure of tool material database, Table 4.2 of work piece material database, and Table 4.3 of process database. Note that there is a separate process database for each of the six processes included in the system.

Field Name	Description
ID #	Code used within the system, e.g. 'TM0100' where first two positions (TM) stands for Tool Material and positions 3-6 are for identifying different tool materials.
Tool Material	Name of tool material.
Taylor's C	Constant C for Taylor tool life equation, being a function of tool material used. Constant n for Taylor's tool life equation is specified in a separate data file since it depends on combination of work and tool materials.
ISO grade	Standard ISO grading for tool material.
AISI grade	Standard AISI grading for tool material.
Turning	Logical field to show the availability of a specific tool material for this process.
Milling	Same as turning.
Drilling	Same as turning.
Cutoff	Same as turning.
Shaping	Same as turning.
Grinding	Same as turning.

Table 4.1: Structure of Tool Material Database (Tool.db)

Field Name	Description
ID #	Code defined for stored work material used within the system, e.g. 'WM0010' where first two positions (WM) stands for Work Material, and position 3-6 are for identifying different work material.
Work material	Name of work material.
Hardness low	Lower bound of the hardness range for the defined work material.
Hardness high	Upper bound of hardness range.
Unit hp	Unit horse power which depends on the work material.
Condition	Available information on how and what temperature environment the work material is formed.

Table 4.2: Structure of Workpiece Material Database (Work.db)

Field Name	Description
Combination #	Code for different combination of parameters used in this operation.
Work material	Work material code defined in work.db and used by this operation.
Tool material	Tool material code defined in tool.db and used by this operation.
Tool parameters	Tool parameters are defined for individual processes: Turning: Not applicable. Milling: Diameter of cutter. Drilling: Diameter of cutter. Cutoff: Not applicable. Shaping: Not applicable. Grinding: Ratio of wheel width for maximum cross feed.
Work parameters	Work parameters are defined for individual processes: Turning: Depth of cut, cutting speed, and feed rate. Milling: Depth of cut, cutting speed, and feed rate. Drilling: Cutting speed and feed rate. Cutoff: Cutting speed and feed rate. Shaping: Depth of cut, cutting speed, and feed rate. Grinding: Down feed, table feed, and cross feed.

Table 4.3: Structure of Process Databases

4.2 Simulation Module

The simulation module is used to analyze the material and process combination in terms of technical and economic performance measures. Technical measures include throughput, work-in-process inventories and machine utilizations; economic measures include production costs. The use of simulation in this context has several advantages over analytical solutions. Deriving a feasible solution with analytical models usually requires greater simplifying assumptions. A simulation model can better represent the manufacturing environment because fewer restrictive assumption are required. In addition, simulation can provide solutions when analytical models fail to do so. Although simulation is not an optimization tool, it provides a method for exhaustively exploring many possible solutions when no simple analytical search for the optimum is available.

4.2.1 Implementation Tool

The simulation model is developed in SIMAN simulation language. SIMAN is an integrated simulation language that includes three different modeling views: network or process, discrete and continuous. In addition, it provides powerful statistical capabilities, and the ability to interact with external programs (such as PARADOX), and higher level languages, such as FORTRAN and C. A SIMAN program separates a simulation model file into two components: a model file that defines the process flow, and an experimental file that describes simulation parameters including processing times and their distributions, random number streams, and simulation time or the number of units to be produced. By separating the model structure and the experimental frame into two distinct elements, different simulation experimental conditions for executing the model can be run by only changing the experimental frame. The system model developed for a manufacturing environment remains the same. This feature helps to simplify the integrated system by only interacting with the experimental frame for evaluating different modeling scenarios.

4.2.2 Structure

The simulation module consists of three main components:

1. A SIMAN network model for modeling production flow sequence and the processing at each operation.
2. A SIMAN experimental frame, generated automatically from user input during execution for specifying parameters of operations models in the network components.
3. FORTRAN-written subroutines for initiating the simulation process, for describing the statistical data sampling functions, and for computing and printing output results.

The SIMAN network model is used to represent the standard operational behavior to be performed in automated manufacturing production. The standard operations for each process include loading, unloading, queuing conditions, tool wear-out, tool changing, and operation delay.

The workpiece load/unload times and tool change times are specified by the user, and may be specified as statistical distributions. The selection of specific distribution is through a menu. The distributions available in the system are: constant (CO), uniform (UN), normal(RN), and triangular (TR). Each distribution requires specification of different parameters. Selecting a uniform distribution requires specification of two parameters (minimum and maximum) as does the normal distribution (mean and standard deviation), while triangular distribution requires three parameters (minimum, mod, and maximum). The appropriate parameters need to be specified after a distribution is selected.

Through the experimental frame, the process operations can be arranged in any desired order specified through the query manager. Each of the operations may employ different types of machines (e.g. turning and milling), or may use the same machine but different tools (e.g. turning and cutoff). For each machine, the user can specify a different set of parameters through the user interface. By combining specifications of the operational process and machining parameters, an experimental frame is defined for running the system model described in the network model.

Two subroutines (PRIME and WRAPUP) and one function (UF) in FORTRAN code form the discrete event model component of the simulation module. The SIMAN sub-program library provides standard event functions for user-written subroutines to enhance this capability. Subroutine PRIME, called only once at the beginning of simulation, retrieves information from the query manager and establishes the initial conditions for the operational parameters (such as tool changing time and tool life based on statistical sampling distributions), and system parameters (such as number of parts to be produced). The other subroutine, WRAPUP, is executed only at the end of simulation run to output formatted results (e.g. production cost) and for communicating with the query manager for sensitivity analysis. In addition, the user-specified function, UF, is built to assign one of several different set of information (such as process time, tool changing time, and tool cost) unique to a process for cost computations depending on the passed function parameters. Thus, each process operation can have individual production cost based on its unique process time, tool changing time, and tool cost.

4.2.3 Input Parameters

The user input, entered through the query manager, is automatically translated into input parameters used by the simulation model. During the translation process, the system automatically checks for consistency in configuration and prompts the user if a mismatch occurs. This eliminates the possibility of crashing the program during input or editing.

In the simulation module, input parameters are required both for specifying manufacturing conditions and process operations. Information for manufacturing conditions include simulated processes, sequence of processes, and number of parts to be manufactured; input parameters for process operations include process times, tool life, tool cost, operating costs, and statistical distributions for tool changing time, and loading and unloading times.

4.2.4 Simulation Output

The simulation model is executed to produce a specified number of units. Two types of simulation output are available to the user: a customized output report and SIMAN summary report. The customized report has been designed to compute and present results on production economics; display of this report is through the designed screen in PARADOX. The SIMAN summary report is produced using the report generating features built in SIMAN; the display can be through the default SIMAN screen and computer printout. The later report includes information on the throughput times, machine utilizations, and inventory levels. The user can either accept the results or conduct a sensitivity analysis.

4.2.5 Sensitivity Analysis

Once a production process has been identified, it is important that it satisfies the original objectives of the design program. Sensitivity analysis focuses on product design for ease of assembly and handling, and on the simplification of the product specifications to further promote ease of manufacturing, improve quality, and reduce manufacturing costs. For example, a minor change in one or more specifications may result in substantial cost reductions, simplifications in tooling, fixturing, and material handling required to support the process.

Sensitivity analysis involves repeated computations with different processing sequences or with the same processing sequence but with different production parameters. The overall objective of the design phase is to identify a combination of material and process attributes that lead to meeting the stated need. This process may necessitate tradeoffs that can be explored using sensitivity analysis.

Simulation is an extremely effective and perhaps the only feasible approach for analyzing complex production systems. This is because it can represent complex dependency structures, both analytic and non-analytic, and explore different scenarios with rapid computational speed. However, this type of a heuristic methodology implies that "optimum" solutions with boundary conditions cannot be obtained with simulation analysis. Boundary conditions for a given system may only be established through repeated simulations and analysis under different scenarios.

4.3 User Interface

The database environment is designed to not only store information on material and processes, but also to accommodate other user-defined information such as statistical distributions for different time elements, tool costs, and operating costs. In addition, there is another component of the database that maintains the output reports generated by the simulation module during the initial evaluation and sensitivity analysis. The other functions supported from database management are menu selections for accessing different sections of the system and the interface for sequencing the selected processes.

The system data manipulation component is developed under Paradox 3.0 environment by PAL (Personal Application language). The functions used in developing query manager, for accessing the basic data information, and for providing connections among tables of information, user interfaces, and calculations are developed as an application within Paradox 3.0.

4.3.1 System Interaction

The initial menu screen (Figure 4.1) displays five choices. These are: general information review; data input of the manufacturing process; executing simulation; viewing the simulation results and performing sensitivity analysis; file manipulation; and leaving the system.

The main menu shown in Figure 4.1 demonstrates the consistent style used throughout the system development. It represents a nested menu structure where selecting and an item from a menu may produce a display of the sub-menu associated with the selected item. For example, the menu screen shown in Figure 4.2 displays the available choices under **PROCESS**. Options included in this sub-menu include setting up sequence of process and editing the input information; these are the two major tasks required in the setup of specific design and manufacturing activity. The choice of Quit always gets the users back to higher level menu screen.

::::::: CHOOSE APPLICATION ::: MAIN MENU ::::::::::::::::::	
INFO	View the general information about the system.
PROCESS	Setup the manufacturing processes.
SIMULATE	Simulate one or a group of processes.
ANALYZE	View the simulation results or the sensitivity analysis.
FILE	Export, import, or initialize a process table.
QUIT	Quit the application and return to Paradox.

Figure 4.1: Main Menu

::::::: CHOOSE APPLICATION ::::::::::::::::::::::::::::::::::::::: PROCESS MENU :::::::::::::::	
INFO	View the general information about the section.
SETUP	Setup the sequence of processes (view, move, or delete).
EDIT	Edit the processes (view or modify).
QUIT	Quit and return to Main Menu.

Figure 4.2: Process Menu

4.3.2 Specifying Processing Sequence

The built-in flexibility of manufacturing design can be maintained by adding, removing, and changing sequence of manufacturing processes according to the result of the product design activity. In the screen shown in Figure 4.3, the candidate processes are all linked according to the specified sequence.

Manufacturing Process

Sequence	Process
1	TURNING
2	CUT-OFF
3	MILLING

Figure 4.3: Specifying Processing Sequence

4.3.3 Process Screens

The data input screens for each process are shown in Figures 4.4 - 4.9. The elements within each process differ according to the characteristics of the process. The organization of input information is based on the same general principle, that is grouping similar input in clusters. Overall, there are five input groups related to work and tool materials, work dimensions, machine specification, time distribution and cost parameters.

Three of the five groups, work/tool materials, time distributions and cost parameters, have identical elements for all processes included in system. For the first group, all processes need specifications of tool and workpiece materials. The workpiece material is same for all processes and is specified at the start of analysis. This parameter is automatically retrieved and displayed in the screens. The tool material may differ from one process to the next; the user has the option of

```
Function : TURNING                    Record 1
     Work material :                      (        ) <-Shift F3
     Tool material :                      (        ) <-Shift F4
WORK            Initial diameter :           inch
                Final diameter   :           inch
                Length of cut    :           inch
MACHINE         Horse power      :           motor
                Efficiency       :      .6
TIME        Distribution Parameter     Parameter     Parameter
                Shift F6     1             2             3
Load/Unload    : (    )            min,          min,          min
Tool changing  : (    )            min,          min,          min
Process time   :   CO              min
COST           Operating Cost :          $/hr
               Tooling Cost   :          $/tool
```

Figure 4.4: Turning Operation

```
Function : CUT-OFF                    Record 1
     Work material :                      (        ) <-Shift F3
     Tool material :                      (        ) <-Shift F4
WORK            Length of cut    :           inch
MACHINE         Horse power      :           motor
                Efficiency       :      .6
TIME        Distribution Parameter     Parameter     Parameter
                Shift F6     1             2             3
Load/Unload    : (    )            min,          min,          min
Tool changing  : (    )            min,          min,          min
Process time   :   CO              min
COST           Operating Cost :          $/hr
               Tooling Cost   :          $/tool
```

Figure 4.5: Cutoff Operation

```
======== Function : MILLING ================ Record  1 =========

     Work material :                      (        ) <-Shift F3
     Tool material :                      (        ) <-Shift F4
================================================================
WORK               Length of work  :          inch
                   Width of work   :          inch
                   Depth of work   :          inch
================================================================
MACHINE   Horse power   :          motor, Efficiency:      .6
& TOOL    Tool diameter:           inch, <-(Shift F5)
          No. of teeth/rev.:
================================================================
TIME         Distribution Parameter    Parameter     Parameter
                 Shift F6     1            2             3
Load/Unload    : (    )            min,         min,         min
Tool changing  : (    )            min,         min,         min
Process time   :   CO              min
================================================================
COST             Operating Cost :            $/hr
                 Tooling Cost   :            $/tool
```

Figure 4.6: Milling Operation

```
======== Function : SHAPING ================ Record  1 =========

     Work material :                      (        ) <-Shift F3
     Tool material :                      (        ) <-Shift F4
================================================================
WORK               Length of work  :          inch
                   Width of work   :          inch
                   Depth of work   :          inch
================================================================
MACHINE   Horse power   :          motor, Efficiency:      .6
================================================================
TIME         Distribution Parameter    Parameter     Parameter
                 Shift F6     1            2             3
Load/Unload    : (    )            min,         min,         min
Tool changing  : (    )            min,         min,         min
Process time   :   CO              min
================================================================
COST             Operating Cost :            $/hr
                 Tooling Cost   :            $/tool
```

Figure 4.7: Shaping Operation

```
═══════ Function : GRINDING ═══════════════ Record  1 ═══════
     Work material :                           (        ) <-Shift F3
     Tool material :                           (        ) <-Shift F4
═════════════════════════════════════════════════════════════
WORK              Length of work  :              inch
                  Width of work   :              inch
                  Depth of work   :              inch
═════════════════════════════════════════════════════════════
MACHINE   Horse power   :          motor, Efficiency:      .6
& TOOL    Width of wheel :           inch
═════════════════════════════════════════════════════════════
TIME        Distribution Parameter    Parameter    Parameter
                Shift F6     1            2            3
Load/Unload      : (    )          min,         min,         min
Tool changing    : (    )          min,         min,         min
Process time     :   CO            min
═════════════════════════════════════════════════════════════
COST             Operating Cost :          $/hr
                 Tooling Cost   :          $/tool
```

Figure 4.8: Grinding Operation

```
═══════ Function : DRILLING ═══════════════ Record  1 ═══════
     Work material :                           (        ) <-Shift F3
     Tool material :                           (        ) <-Shift F4
═════════════════════════════════════════════════════════════
WORK              Length of hole:             inch
═════════════════════════════════════════════════════════════
MACHINE           Horse power    :            motor
   &              Efficiency     :        .6
 TOOL             Drill diameter:             inch <-(Shift F5)
═════════════════════════════════════════════════════════════
TIME        Distribution Parameter    Parameter    Parameter
                Shift F6     1            2            3
Load/Unload      : (    )          min,         min,         min
Tool changing    : (    )          min,         min,         min
Process time     :   CO            min
═════════════════════════════════════════════════════════════
COST             Operating Cost :          $/hr
                 Tooling Cost   :          $/tool
```

Figure 4.9: Drilling Operation

selecting the desired tool material. The keys to the right of parameter specification in the figures displays a menu of selection. For example, pressing <Shift> and <F4> simultaneously would display the available tool material for a process from which selection can be made.

The time parameters' specification group is also common to all six processes. The process time for an operation represents the machining time and is calculated based on work and machine characteristics. Computation of processing time for each individual operation is explained in the next section.

The two primary cost parameters specified by the user are the operating costs (in $/hour) and tooling costs (in $/tool). Again, the same two parameters specification is required of all process screens though tooling costs may differ from one process to another.

Specification of work dimensions, and machine and tool characteristics are different for each process as can be seen from Figures 4.4 - 4.9. The use of this information in computing process times is explained below.

4.3.4 Computing Machine Process Times

The information from interface screens is then passed to the preprocessing module for calculation of process times. For each process, a working database is created based on the information specified in input screens. This working database also contains information extracted from system databases, primarily as a function of work and tool materials. For the turning process, each record has a different combination of feed rate and cutting speed associated with the depth of cut. However, selecting a depth of cut must consider the machine capability specified as machine horse power, machine tool efficiency, and the unit horsepower that is selected by matching work and tool materials. The metal removal rate calculated based on machine power is considered as the upper bound for the operation for each combination of depth of cut, cutting speed, and feed rate. Any set of machine parameters that results in a smaller metal removal rate qualifies as satisfying machine capacity. Of the potential candidates, the maximum depth of cut is first performed to remove maximum metal. (The depth of cut incrementally reduced as the amount of material to be removed decreases.) The iterations are repeated until the specified work dimension is reached. The process time is the cumulative time for all iterations required to reach specified dimension. The process time for each iteration is the product of the number of cuts times the cutting time for each cycle of turning based on the expressions given in Table 3.1. The tool life is also computed within each iteration, since the feed rate is a major element in Taylor's tool equation and changes as the depth of cut changes.

The cutoff process has a different set of machining parameters from the turning process. Only the cutting speed and feed rate are machine parameters involved for cutoff. The considerations of machine capacity and tool life for turning is also applicable for cutoff.

In the milling process, the cut is moving along the flat surface to remove one layer of workpiece at a time. A large depth of cut is initially preferred like in the turning process. When matching the specifications from milling screen with the milling database, in addition to work and tool materials, the diameter of cutter is also the key to select the machining parameters. The diameter of cut will give the right corner angle in the specification of work dimension for certain types of milling processes. The other tool information such as number of teeth per revolution is also one of the factors in calculating metal removal rate in addition to feed rate, cutting speed, and depth of cut.

The shaping process has same required machine parameters as turning process and same concerns for machine capacity. The calculated tool life factor and processing time use same criteria as turning process in the preprocessing module.

In the drilling process, the diameter of cutter, just like the milling process, is considered in determining the machining parameters, in addition to work and tool materials. The machine capacity is evaluated through machine power at motor, efficiency and unit horsepower, and is compared with the metal removal rate calculated from matched cutting speed, feed rate, and diameter of cutter. The cutting speed and feed rate will take turns to be reduced by 5% in order to perform the drilling operation under the machine capacity. Hole drilling is a one-step process. The tool life factor and processing time can be determined once the level of feed rate and cutting speed is derived.

The operation in surface grinding involves wheel downfeed rate (inch/pass), table crossfeed rate with up to curtain ratio of wheel width, and table speed. The set of machining parameters is selected from grinding database by matching work and tool materials. The metal removal rate can be calculated by those parameters and compared with the maximum metal removal rate from machine power at motor, machine tool efficiency, and unit horsepower. The smaller downfeed is always used for finishing fine surfaces. The downfeed rate in grinding and depth of work removed from process are used to determine the number of cuts required to reach the specification. The criteria of cumulating processing time and tool life is same as turning and most of the other processes.

4.4 Integration of Individual Components

From an implementation perspective, the system components described in last three sections are completely integrated within the framework of Figure 3.2. The information translation between PARADOX and SIMAN is done by an imbedded FORTRAN program. The entire process from translating the user input into SIMAN constructs and extraction of information from the databases is internal and totally transparent to the user.

The component integration is controlled through one batch file which contains the necessary DOS commands. The batch file is brought up in PARADOX; while executing this file the procedures for PARADOX's applications are temporarily suspended. After the last command in batch file is performed, the control returns back to the PARADOX environment. The jobs scheduled in batch file are executed in sequence briefly described below. (Figure 4.10)

1. Call to FORTRAN program (SIMFILE.EXE): This program reads process and machining information from the formatted output generated in PARADOX environment, formats the information for the required parameters in the experimental frame, and then creates a file for use in SIMAN as experimental frame for simulation module.

2. Compilation of the experimental frame generated in Step 1. The

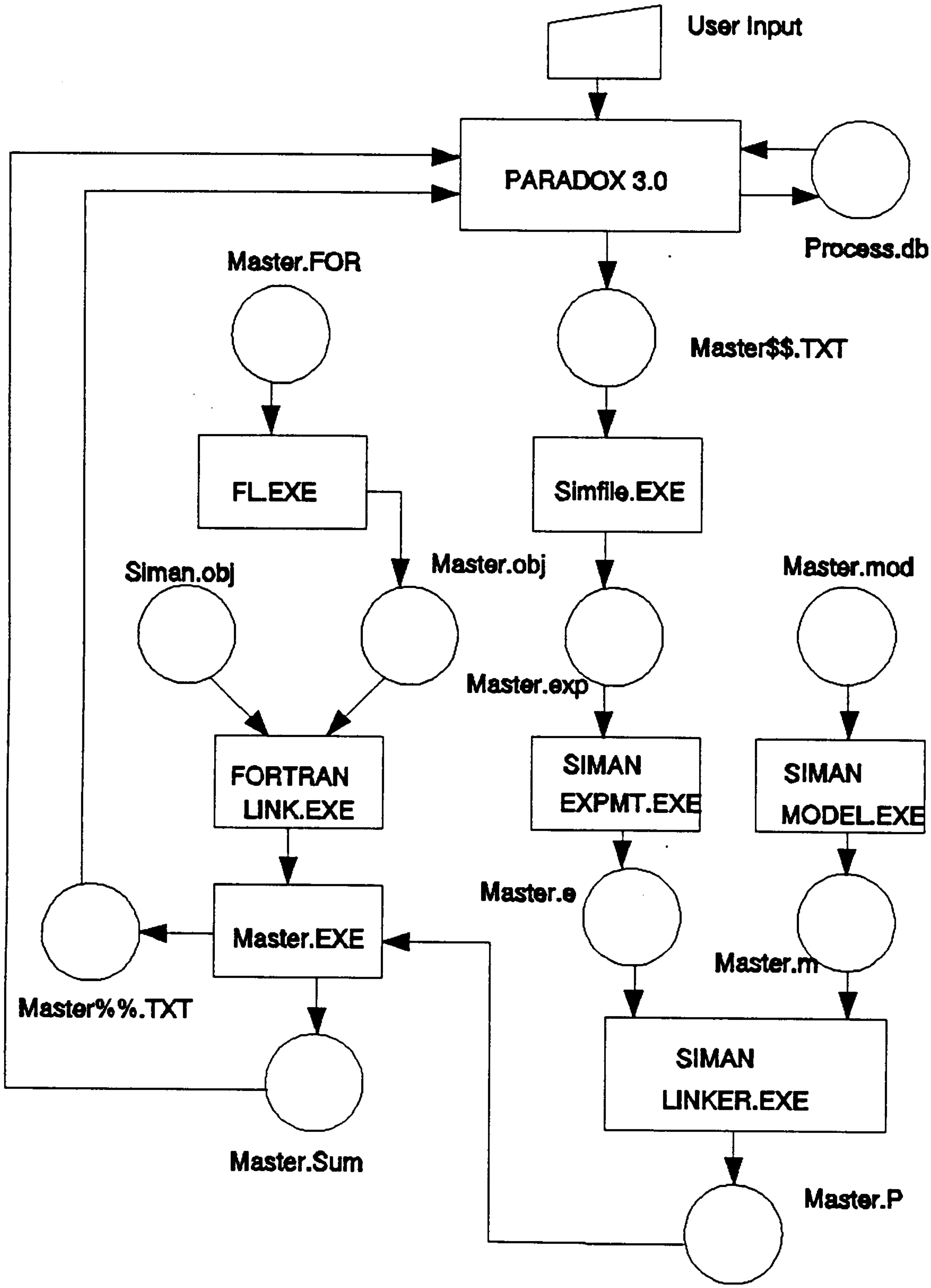

Figure 4.10: System Implementation Flow Chart

compiling program (EXPMT.EXE), a SIMAN module, reads the input statements for the experimental frame elements and generates an experiment listing and an experiment file as output.

3. Linking the experimental file with the model file. The model file is pre-compiled output from the simulation model frame. Since the model file **(master.mod)** contains the sequence of processing operations, this sequence is established based on user input. The SIMAN program (LINKER.EXE) links the model and experimental files into a single executable program file.

4. Call to the executable simulation program with the linked simulation model as the parameter. The simulation program is made by linking SIMAN object file and the compiled FORTRAN subroutines used for the discrete event modeling in simulation module. In this step, two outputs are generated: standard SIMAN output and the formatted file that maintains essential information (such as process sequence and production costs) for PARADOX module (e.g. query manager).

The connection between modules is critical due to the different files' integrity. The first step after PARADOX takes over the control is to translate and save simulation results stored in a file to database information format required under the PARADOX environment for sensitivity analysis. The user can thus view the results, make changes in input and rerun simulation for sensitivity analysis. The entire execution environment shown in Figure 4.10. is completely integrated. Thus, a user with little or no computer background can use the system effectively.

5. System Application Examples

System application is illustrated through two examples. The production of the first part requires three operations: turning, cutoff, and milling (Figure 5.1). The second part is again a relatively simple part (Figure 5.8), but it can be produced using three different sets of operations; the production economics using these different methods is compared.

5.1 Example One

The first example pertains to the production of part shown in Figure 5.1. The part has two inches diameter at on end and one and half inches at the other; a pocket of 0.75 inch by 0.75 inch with depth of 0.25 inch is carved from the surface at the narrow end.

To accomplish production of the part, three machining processes are required. These are cylindrical turning, cutoff, and pocket milling. The processing data for each operation is summarized in Figure 5.2. The initial choice of part material is low carbon steel (AISI 1020). The production of the part starts from Workstation 1 with turning machine. Two machining operations are performed at this workstation: cylindrical turning and cutoff. The tool material for both operations is high speed steel. The cylindrical turning operation is used for reducing the diameter from two inches to one and half inches through metal removal; the cutoff operation then cuts the entire 1 inch from edge to center. After the operation at the first workstation is completed , the part then starts the second process if Workstation 2 is idle. Otherwise, the part is placed in a waiting queue. The milling operation in Workstation 2 is required to produce the square pocket. The tool material for

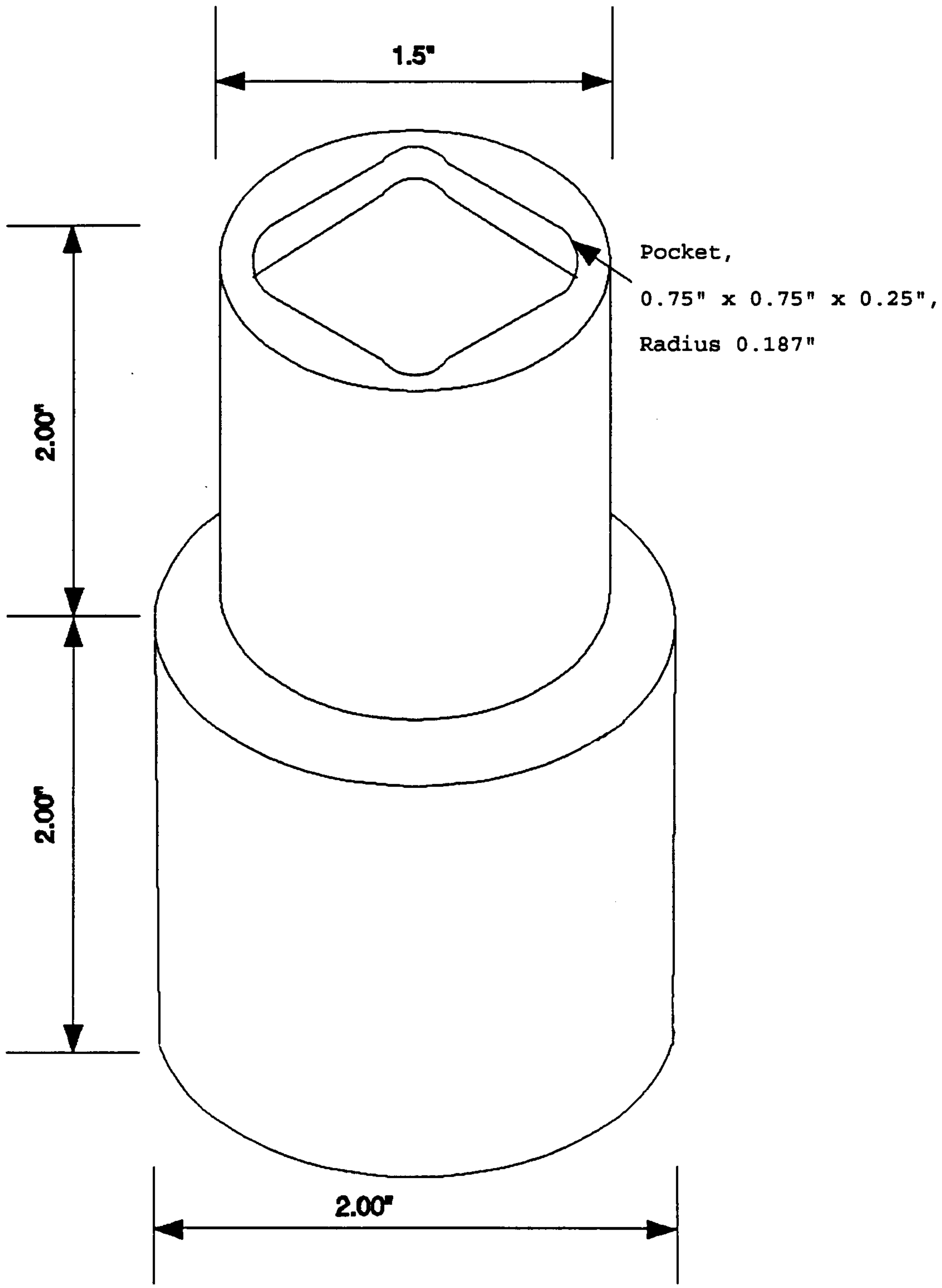

Figure 5.1: Part Diagram for Example One

milling is high speed steel. The length and width of the pocket determines the cutting distance; the depth of pocket determines the number of surface milling sweeps required to finish the work. It is assumed that during the simulation analysis, work material (raw material) at the first workstation is always available.

Part material: Gray Cast Iron

Workstation 1: Turning

Operation 1: Cylindrical Turning

Tool Material	=	High Speed Steel
Initial Diameter	=	2.00 inch
Final Diameter	=	1.5 inch
Length of cut	=	2.00 inch

Operation 2: Cutoff

Tool Material	=	High Speed Steel
Length of cut	=	1.00 inch

Workstation 2: Milling

Operation 1: Pocket Milling

Tool Material	=	High Speed Steel
Length of cut	=	0.75 inch
Width of cut	=	0.75 inch
Depth of cut	=	0.25 inch

Figure 5.2: Data Summary for Example One

The input screens for turning, cutoff, and milling are shown in Figures 5.3 through 5.5, respectively. The data input parameters were described in the previous chapter; that discussion is not repeated here.

The results based on the production of 200 parts are shown in Figure 5.6 and Figure 5.7. The results show that the unit production costs for the turning and milling workstations were $16.96 and $6.88, respectively. The SIMAN summary report shows that the average throughput time for a unit was 37.53 minutes and the utilization of the two workstations were

```
Function : TURNING ==================== Record 1 ====

     Work material : GRAY CAST IRON       ( WM0040 ) <-Shift F3
     Tool material : HIGH SPEED STEEL     ( TM0100 ) <-Shift F4
==============================================================
WORK              Initial diameter :      2.00  inch
                  Final diameter   :      1.50  inch
                  Length of cut    :      2.00  inch
==============================================================
MACHINE           Horse power      :     20.0   motor
                  Efficiency       :       .7
==============================================================
TIME         Distribution Parameter    Parameter     Parameter
                  Shift F6     1           2             3
Load/Unload     : ( UN )      1.5    min,  2.0   min,          min
Tool changing   : ( UN )      1.5    min,  2.0   min,          min
Process time    :   CO       7.21    min
==============================================================
COST             Operating Cost :    20.0   $/hr
                 Tooling Cost   :     4.0   $/tool
```

Figure 5.3: Turning Screen for Example One

```
Function : CUT-OFF ==================== Record 2 ====

     Work material : GRAY CAST IRON       ( WM0010 ) <-Shift F3
     Tool material : HIGH SPEED STEEL     ( TM0100 ) <-Shift F4
==============================================================
WORK              Length of cut    :      1.0   inch
==============================================================
MACHINE           Horse power      :     20.0   motor
                  Efficiency       :       .6
==============================================================
TIME         Distribution Parameter    Parameter     Parameter
                  Shift F6     1           2             3
Load/Unload     : ( UN )      1.5    min,  2.0   min,          min
Tool changing   : ( UN )      1.5    min,  2.0   min,          min
Process time    :   CO               min
==============================================================
COST             Operating Cost :    20.0   $/hr
                 Tooling Cost   :     4.0   $/tool
```

Figure 5.4: Cutoff Screen for Example One

```
═════════ Function : MILLING ══════════════ Record  3 ═════════

     Work material : GRAY CAST IRON        ( WM0040 ) <-Shift F3
     Tool material : HIGH SPEED STEEL      ( TM0100 ) <-Shift F4
═══════════════════════════════════════════════════════════════
WORK            Length of work  :      0.75   inch
                Width of work   :      0.75   inch
                Depth of work   :      0.25   inch
═══════════════════════════════════════════════════════════════
MACHINE  Horse power  :    20.0   motor, Efficiency:       .6
& TOOL   Tool diameter:    0.375  inch, <-(Shift F5)
         No. of teeth/rev.:  4
═══════════════════════════════════════════════════════════════
TIME        Distribution Parameter    Parameter   Parameter
                 Shift F6     1           2            3
Load/Unload    : ( UN )      1.5   min,  2.0   min,         min
Tool changing  : ( UN )      1.5   min,  2.0   min,         min
Process time   :   CO              min
═══════════════════════════════════════════════════════════════
COST            Operating Cost :    20.0   $/hr
                Tooling Cost   :     4.0   $/tool
```

Figure 5.5: Milling Screen for Example One

SIMAN Summary Report

Tally Variables

Number	Identifier	Average	Standard Deviation	Minimum Value	Maximum Value	Number of Obs
1	TIME IN SYSTEM	37.53	1.91	18.57	40.96	200

Discrete Change Variables

Number	Identifier	Average	Standard Deviation	Minimum Value	Maximum Value	Time Period
1	QUEUE 1	1.00	.04	.00	1.00	3227.53
2	QUEUE 2	.33	.48	.00	1.00	3227.53
3	MACHINE 1	.99	.08	.00	1.00	3227.53
4	MACHINE 2	.00	.00	.00	0.00	3227.53

Figure 5.6: SIMAN Output Report

Cost for Process

Sequence	Process	Cost
1	TURNING	11.27
2	CUT-OFF	5.69
3	MILLING	6.88

Figure 5.7: Summary of Economic Results

100% and 33%, respectively. On average, there was one part waiting before the turning workstation to be processed; the milling workstation was under-utilized.

Once the "base" analysis is complete, a number of different type of sensitivity analysis can be performed. Perhaps the most obvious is sensitivity to production parameters. For example, the output of the base model (Figure 5.6) shows that the utilization of the turning workstation is 100%, thus representing the bottleneck in the system. Increasing turning workstations available to the system, or increasing the current production rate of the turning workstation will decrease the average throughput time of the part product. However, increasing units of workstations may be an expensive undertaking, and increasing the production rates may be limited for a given piece of equipment.

Alternatively, one can investigate sensitivity to parameters specified at the design stage. For example, different combinations of work and tool materials can be used to produce the part shown in Figure 5.1 as long as the same functional properties are maintained. The results based on four different combinations of work material and tool material are shown in Table 5.1. The processing time required to produce a unit is a function of work and tool materials; in addition, parameters such as tooling costs may be different for different tool materials.

Work Material Tool Material	Gray Cast iron High Speed Steel	Gray Cast Iron Sintered Carbide	Low Carbon Steel High Speed Steel	Low Carbon Steel Sintered Carbide
Cost ($/unit)				
Turning	11.27	1.46	11.27	1.18
Cutoff	5.69	2.69	5.69	4.16
Milling	6.88	3.92	6.59	3.13
Total	23.84	8.07	23.55	8.47
Thruput Times (minute)	37.53	12.38	37.86	11.40
Utilization(%)				
Machine 1	100	100	100	100
Machine 2	33	55	35	50

Table 5.1 : Summary of Sensitivity Analysis for Example One

Table 5.1 shows that using high speed steel as tool material results in higher processing costs and higher throughput time as compared to using sintered carbide as tool material. Even though sintered carbide tooling is more expensive than high speed steel tools, particularly for cutoff and milling operations, it does result in higher cutting speeds and bigger depth of cut; both of these contribute in reducing the overall processing costs. Using sintered carbide tools with gray cast iron or low carbon steel work materials give comparative results for processing times and costs.

5.2 Example Two

The second example concerns the production of part shown in Figure 5.8. The drawing in Figure 5.8 shows the dimension of part. The part is 2" by 2" by 0.5"; two 0.5"-diameter holes are located 1.0" apart. The holes go all way through the 0.5" thickness.

Part production requires two major operations: surface work and hole drilling. The surface work or metal removal can be accomplished by using one of two operations: milling or shaping. The grinding operation is optional for producing more accurate and finer surface. Milling, shaping, and grinding are performed on separate workstations.

The part material selected in this example is low carbon steel (AISI 1020). The tool material for all processes but the grinding operation is high speed steel. The tool material for grinding is sintered carbide. The processing data for each operation is summarized in Figure 5.9. The results are based on production of 200 parts.

The production costs and operational results for all options are shown

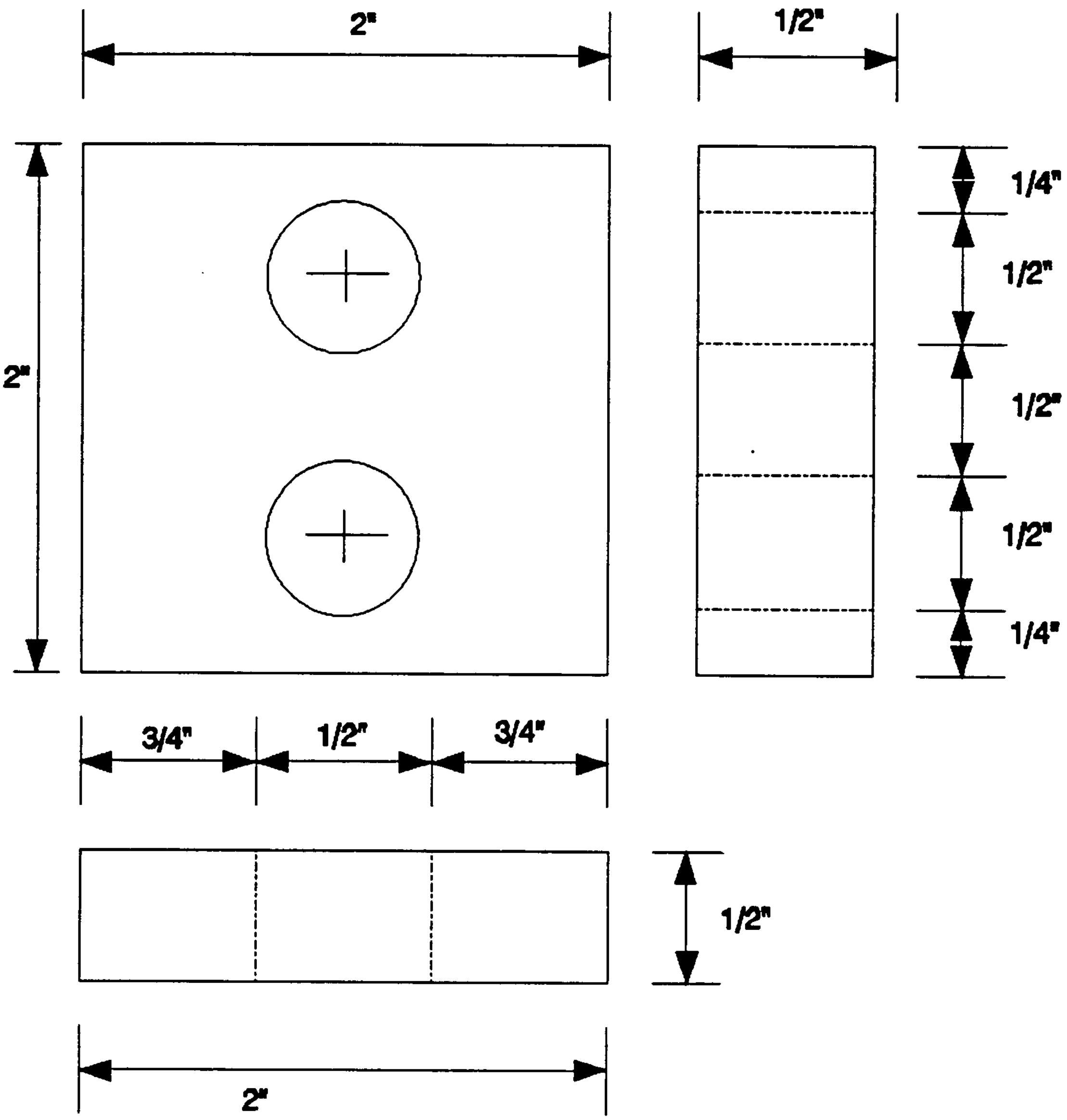

Figure 5.8: Part Diagram for Example Two

Part material: Low Carbon Steel (AISI 1020)

Workstation 1: Milling (surface work)

Tool Material	=	High Speed Steel
Length of cut	=	2.0 inch
Width of cut	=	2.0 inch
Depth of cut	=	0.05 inch

Workstation 2: Shaping (surface work)

Tool Material	=	High Speed Steel
Length of cut	=	2.0 inch
Width of cut	=	2.0 inch
Depth of cut	=	0.05 inch

Workstation 3: Grinding (surface finishing)

Tool Material	=	Sintered Carbide
Length of cut	=	2.0 inch
Width of cut	=	2.0 inch
Depth of cut	=	0.005 inch
Width of wheel	=	0.5 inch

Workstation 4: Hole Making

Operation 1: Drilling

Tool Material	=	High Speed Steel
Length of Hole	=	0.5 inch
Drill Diameter	=	0.5 inch

Operation 2: Drilling

Same as operation 1.

Figure 5.9 : Data Summary for Example Two

in Table 5.2. The results show that the production using milling operation results in higher processing costs and higher throughput times than using shaping operation for both cases of with and without the grinding operation for surface finish. Obviously, added grinding operation increases production costs and throughput times as compared with single operation on surface work. In general, a single milling operation can produce an acceptable surface finish. However, adding the grinding operation generates finer surface. The shaping operation uses a single point tool so that the produced surface has ridges. Adding the grinding operation improves the rough surface generated by the shaping operation. The least expensive method is to produce the part using the shaping operation if resulting surface finish meets the functional requirements of the product.

Processes	Milling + Drilling	Shaping + Drilling	Milling + Grinding + Drilling	Shaping + Grinding + Drilling
Cost ($/unit)				
Surface work	8.09	1.96	8.10	1.97
Surface finish			0.95	0.95
Drilling	0.90	0.90	0.90	0.90
Total	8.99	2.86	9.95	3.82
Throughput Time (minute)	38.39	12.97	40.77	15.35
Utilization (%)				
Workstation 1	100		100	
Workstation 2		100		100
Workstation 3			13	45
Workstation 4	14	47	14	47

Table 5.2 : Summary of Sensitivity Analysis for Example Two

6. Conclusions

6.1 Research Objectives

Integrated conceptual design of a product involves the selection of product material and process to produce the desired product characteristics, and analysis of the processing system in terms of operational parameters and production economics. This research developed a system that integrates these features. This was accomplished by combining database and simulation techniques with machining processes used in discrete manufacturing industry.

A "front end" user interface was designed to obtain the design information from the users; material and process characteristics are then extracted from system databases. This information along with product specifications was simulated to obtain system performance measures. The system integrates product characteristics and manufacturing operations since a product's status cannot be assessed independent of available technology. Finally, it allows analysis of a wide range of alternatives without prior commitment to any particular alternative.

6.2 Future Extensions

6.2.1 Expanding Processes and Material Databases

There are more machining processes such as sawing and broaching, and thread manufacturing processes are not currently included in the system. Expanding the processes' database to include these processes will improve the utility of the system.

Including more work materials will enable the designer to more efficiently survey and evaluate available candidate materials and processes for a given product. A broader range of engineering materials would need to be available in the database. In addition to the materials discussed in this research, other work materials include plastics, rubber and elastomers, ceramics and glass, composites, fibers, and coatings.

Success in metal cutting depends upon the selection of the proper cutting tool for a given work material. Tool material technology is advancing rapidly, enabling to manufacture difficult-to-machine materials to be machined at higher removal rates and/or cutting speed with greater performance reliability. A range of cutting tool materials is available with a variety of properties, performance capabilities, and costs. Other candidate tool materials include cast cobalt alloys, ceramics, cast carbides, and coated carbides.

6.2.2 CAD Connection

A front-end connection of the system discussed in this research with a computer-aided design system (such as AutoCAD) can provide product specification input directly from a CAD drawing. This would not only increase system usefulness, but would eliminate the time users have to spend providing input to the analysis system that can be directly read from a CAD drawing.

6.2.3 CAM Connection

Using the system presented in this research enables the user to evaluate and compare alternative manufacturing sequences and machining parameters with an objective to identify the most cost efficient alternative. However, the final objective is not analysis, but manufacturing of the actual product. The "best" alternative identified through this analysis can be selected, and its specifications translated into machining instructions for a computer-aided manufacturing (CAM) environment. Linking the analysis module with a CAD system at the front-end and with a CAM system at the back-end will truly provide a "design for manufacturing" environment. If desired, heuristic knowledge concerning materials and process interaction may also be included in this system through expert system technology.

REFERENCES

1. Allen, A.J. and Swift, K., "Manufacturing Process selection and Costing," Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture, 204, 143-148, 1990.
2. Anderson, D.M., Design for Manufacturability, CIM Press, Lafayette, CA, 1990.
3. Bedworth, D.B., Henderson, M.R., and Wolfe, P.M., Computer-Integrated Design and Manufacturing, McGraw-Hill, New York, NY, 1991.
4. Bernold, L., "Enhancing the Civil Engineering Experience Through the Use of Expert System and Robotics," ASEE Annual Conference Proceedings 3215, 1-9, 1988.
5. DeGarmo, E.P., Black, J.T., and Kosher, R.A., Materials and Processes in Manufacturing, Macmillan Publ. Co., New York, NY, 1988.
6. Doyle, L.E., Keyser, C.A., Leach, J.L., Schrader, G.F., and Singer, M.B., "Manufacturing Processes and Materials for Engineers," Prentice-Hall, Englewood Cliffs, NJ, 1985.
7. Encarnacao, J., "Product Data Interfaces in CAD/CAM Applications," IEEE Computer Graphics and Applications 7, 64-66, May 1987.
8. Ettlie, J.E. and Stoll, H.W., Managing the Design-Manufacturing Process, McGraw-Hill, Inc., New Your, NY, 1990.
9. Gal-Tzur, C., Shpitaini, M., and Malkin, S., "Design and Manufacturing Analysis for Integrated CAD/CAM of Cams," Journal of Engineering for Industry 111, 307-314, November 1989.
10. Girgis, M. and Ajayi-Majebvi, A., "Software for Material and Process Selection in Automated Manufacturing," Central State University, Wilberforce, OH, 1989.
11. Gupta, T., "An Expert System Approach in Process Planning: Current

Development and its Future," computers & Industrial Engineering 18, 69-80, 1990.

12. Jakobsom, G., Lafond, C., Nyberg, E., Piatetsky-Shapiro, G., "An Intelligent Database Assistant," IEEE Expert Conference 45, 65-78, Summer 1986.

13. Kern, F.W., "Integration of Numerical Simulation and Flow Visualization in the Design of Equipment of the Manufacture of Submicron Semiconductor Devices," The Journal of Environmental Sciences 32, 19-24, May/June 1989.

14. Knight, W.A., "Design for Manufacture Overview," Design and Manufacturing Conference 5, 1-6, February 1988.

15. Lindbeck, J.R., Williannms, M.W., and Wygant, R.M., Manufacturing Technology, Prentice Hall, Englewood Cliffs, NJ, 1990.

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

17. Machining Data Handbook, Vols. 1 & 2, Machinability Data Center, 3rd Ed., 1980.

18. Matsuhima, K., Olada, N., and Sata, T., "The Integration of CAD and CAM by Application of Artificial Intelligence Techniques," Annual CIPR 31, 329-332, 1982.

19. Meredith, J.R. and Hill, M.M., "Justifying New Manufacturing Systems: A Managerial Approach," Sloan Management Review, 49-61, Summer 1987.

20. Metals Handbook, Vol. 16 - Machining, American Society for Metals, 9th Ed., 1989.

21. Niebel, B.W., Draper, A.B., and Wysk, R.A., Modern Manufacturing Process Engineering, McGraw-Hill, New York, NY, 1989.

22. O'Connor, M.J., De La Garza, J., Ibbs, C. "An Expert System for construction Schedule Analysis," American Society of Civil Engineers 228, 67-77, 1986.

23. Paradox 3.0: Introduction to Paradox, Borland International, Scotts Valley, CA, 1988.

24. Pegden, C.D., Shannon, R.E., and Sadowski, R.P., Introduction to Simulation Using SIMAN, McGraw-Hill, Inc., New York, NY, 1990.

25. Pollack, H.W., Tool Design, Reston Publ. Co., reston, VA, 1976.

26. Pugh, S., Total Design, Addison-Wesley Publ. Co., Wokingham, England, 1990.

27. Tool & Manufacturing Engineers Handbook, Vol. 5 - Manufacturing Management, Society of manufacturing Engineers, 4th Ed., Ch. 13, 1-9, 1988.

28. West, T.M., Randhawa, S.U. and Brings, S.D., "The Role of Product Design In the Evaluation of New Manufacturing Technologies," Proc. 1989 International Industrial Engineering Conference, May 14-17, 1989, Toronto, Canada, 159-163.

APPENDICES

APPENDIX 1

MODEL Frame for Simulation Model

```
; SIMAN model frame: MASTER.MOD
;
; This is a model frame for the manufacturing simulation model.
; Ten work stations are simulated for a sequential production.  The
; eleventh work station in this model frame is used to tally statistics
; and release the entities for parts.
;
;
BEGIN;
      CREATE:,;                                   Parts are ready for processing
CREATE   COUNT:1,1:MARK(4);                       Initialize the total cost for this part
      ASSIGN:NS=1;                                Use the first process sequence
      ASSIGN:IS=0;                                Reset the index of sequence
      ROUTE:0,SEQ;                                Route to the first job of the seq.
;
;
      STATION,1-10;                               The 10 potential work stations
      QUEUE,M;                                    The store space for the work
      SEIZE,2:WORK(M);                            seize the machine for the work
      BRANCH:                                     !Create the next part available or not
       ALWAYS,START:                              !Continue the sequential work
       IF,M.EQ.1.AND.NC(1).LT.X(31),CREATE;       Generate the next part
;
;
START    ASSIGN:A(5)=A(1);                        Set the processing time.
      TALLY:M,UF(M);                              Tally the processing cost.
      DELAY:A(3)/2.;                              Delay the loading time.
      ASSIGN:X(M)=X(M)-X(M+10);                   Reduce the tool life for operation.
;
WEAROUT  BRANCH:
       ALWAYS,PROCESS:                            !Go on!
       IF,A(2).GE.0.0.AND.X(M).LT.0.0,PAUSE1:     !Change tool for 1st operation in station.
       IF,A(2).LT.0.0.AND.X(33).LT.0.0,PAUSE2;    Change tool for next operation in station, if any.
;
PAUSE1   DELAY:A(5)*(X(M)+X(M+10))/X(M+10);       Delay the tool life for regular machine operation.
CHNGTL1  QUEUE,M+10;                              Schedule the delay entity for worn-out tool.
      PREEMPT:WORK(M),1,ENTER1;                   Assign highest priority to use resource (machine).
      ASSIGN:X(M+20)=UF(M+10);                    Assign tool changing delay for machine.
      DELAY:X(M+20);                              Delay tool changing time.
      RELEASE:WORK(M):DISPOSE;                    Resume the processing.
;
ENTER1   QUEUE,M+20;                              Put part back for processing after changing tool.
      SEIZE,1:WORK(M);                            Higher priority than non-started ones.
      ASSIGN:X(M)=X(M)+1.0;                       Reset the tool life back to full charged.
      BRANCH:
       ALWAYS,PROCESS:                            !Go on.
       IF,X(M).LT.0.0,REENTER1;                   Tool changing needed again for same process.
;
REENTER1 DELAY:A(5)/X(M+10):NEXT(CHNGTL1);        Delay tool life before scheduling a delay entity.
;
;                                                 For continuous operation within a work station.
PAUSE2   DELAY:A(5)*(X(33)+X(34))/X(34);          Delay the tool life for regular machine operation.
CHNGTL2  QUEUE,M+30;                              Schedule the delay entity for worn-out tool.
      PREEMPT:WORK(M),1,ENTER2;                   Assign highest priority to use resource (machine).
      ASSIGN:X(32)=UF(M+10);                      Assign tool changing delay for machine.
```

```
       DELAY:X(32);                                  Delay tool changing time.
       RELEASE:WORK(M):DISPOSE;                      Resume the
;
ENTER2  QUEUE,M+31;                                  Put part back for processing after changing tool.
       SEIZE,1:WORK(M);                              Higher priority than non-started ones.
       ASSIGN:X(33)=X(33)+1.0;                       Reset the tool life back to full charged.
       BRANCH:
        ALWAYS,PROCESS:                              !Go on.
        IF,X(33).LT.0.0,REENTER2;                    Tool changing needed again for same process.
;
REENTER2 DELAY:A(5)/X(34):NEXT(CHNGTL2);             Delay tool life before scheduling a delay entity.
;
;
PROCESS  DELAY:A(1);                                 Delay the process time, defined in experiment.
       BRANCH,1:
        IF,A(2).GT.0.0,GO2ND:                        !Continue 2nd operation in same work station.
        ELSE,LEAVE;                                  Leave this work station.
;
GO2ND   DELAY:0.1;                                   Delay time for reindexing tool.
       ASSIGN:A(1)=A(2);                             Assign processing time from experimental frame.
       ASSIGN:A(5)=A(1);                             Reset the processing time for calculation.
       ASSIGN:A(2)=-1.0;                             Reset the flag for continuous operation.
       TALLY:11,UF(M);                               Tally processing cost for continuous operation.
       ASSIGN:X(33)=X(33)-X(34):NEXT(WEAROUT);       Reduce the tool life for operation.
;
LEAVE   DELAY:A(3)/2.;                               Delay the unloading time.
       RELEASE:WORK(M);                              Release the machine for the work
       ROUTE:0,SEQ;                                  Route to the next work station
;
;
       STATION,11;                                   The last station to finish the process
       TALLY:12,INT(4):DISPOSE;                      Tally the time in system and dispose parts
END;
```

APPENDIX 2

FORTRAN Codes for Simulation Model

```
C MS-FORTRAN 77: MASTER.FOR
C
C This file contains two subroutines and one user-defined function to
C schedule special events in the simulation process. The compiled output
C is linked with SIMAN system to perform the required functions for the
C designed model.
C
C Subroutine PRIME:
C        Before the simulation run starts, this procedure reads in the tool
C        life, tool changing time, loading/unloading time, and costs to
C        initialize the variables used and maintained within the simulation
C        run.
C
C Function UF:
C        Calculate the costs by parameters and the simulated variables.
C
C Subroutine WRAPUP:
C        Write out the simulated statistics to an ASCII file for use of
C        database applications.
C
C
C
      SUBROUTINE PRIME
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUM
C
      INTEGER PMAX
      PARAMETER (PMAX=10)

      INTEGER*2 IS, PARTCNT*4, CUTOFF
      CHARACTER*1 LASTPROC*8, PROCID*8, TOOLCHNG(PMAX+1)*2,
     +        LOADTIME(PMAX)*2

      REAL TOOLLIFE(PMAX+1), PROCTIME(PMAX+1),
     +    TOOLCHN1(PMAX+1), TOOLCHN2(PMAX+1), TOOLCHN3(PMAX+1),
     +    LOADTIM1(PMAX), LOADTIM2(PMAX), LOADTIM3(PMAX),
     +    OHCOST(PMAX), TOOLCOST(PMAX+1)

      COMMON/USER1/CUTOFF,IS,TOOLLIFE,PROCTIME
      COMMON/USER2/TOOLCHNG,TOOLCHN1,TOOLCHN2,TOOLCHN3
      COMMON/USER3/LOADTIME,LOADTIM1,LOADTIM2,LOADTIM3
      COMMON/USER4/OHCOST,TOOLCOST

      DATA LASTPROC, TOOLCHNG(11)/' ',' '/
      DATA IS, CUTOFF/2*0/
      DATA PROCTIME(11),TOOLLIFE(11),TOOLCHN1(11),TOOLCHN2(11),
     +    TOOLCHN3(11),TOOLCOST(11)/6*0.0/
C
C READ THE DATA FROM THE FILE MASTER.DAT
C
      OPEN (1,FILE='MASTER$$.TXT',FORM='FORMATTED')
      READ (1,'(I7)') PARTCNT
10    READ (1,'(A,F6.4,F6.1,A,3F6.1,A,3F6.1,2F8.3)',END=100) PROCID,
     +    TOOLLIFE(IS+1), PROCTIME(IS+1), TOOLCHNG(IS+1),
     +    TOOLCHN1(IS+1), TOOLCHN2(IS+1), TOOLCHN3(IS+1),
```

```
     +     LOADTIME(IS+1), LOADTIM1(IS+1), LOADTIM2(IS+1),
     +     LOADTIM3(IS+1), OHCOST(IS+1), TOOLCOST(IS+1)

      IF (PROCID .EQ. LASTPROC) THEN
        PROCTIME(IS) = PROCTIME(IS) + PROCTIME(IS+1)
        TOOLLIFE(IS) = TOOLLIFE(IS) + TOOLLIFE(IS+1)
      ELSE IF (PROCID.EQ.'CUT-OFF'.AND.LASTPROC.EQ.'TURNING') THEN
        CUTOFF = IS
        TOOLLIFE(11) = TOOLLIFE(IS+1)
        PROCTIME(11) = PROCTIME(IS+1)
        TOOLCHNG(11) = TOOLCHNG(IS+1)
        TOOLCHN1(11) = TOOLCHN1(IS+1)
        TOOLCHN2(11) = TOOLCHN2(IS+1)
        TOOLCHN3(11) = TOOLCHN3(IS+1)
        TOOLCOST(11) = TOOLCOST(IS+1)
      ELSE
        IS = IS + 1
        LASTPROC = PROCID
      ENDIF
      GOTO 10

100   CLOSE (1)

      DO 110 I=1,IS
        X(I) = 1.0
        X(I+10) = TOOLLIFE(I)
        IF (TOOLCHNG(I).EQ.'CO') X(I+20) = CO(IS+I)
        IF (TOOLCHNG(I).EQ.'EX') X(I+20) = EX(IS+I,11-I)
        IF (TOOLCHNG(I).EQ.'RN') X(I+20) = RN(IS+I,11-I)
        IF (TOOLCHNG(I).EQ.'TR') X(I+20) = TR(IS+I,11-I)
        IF (TOOLCHNG(I).EQ.'UN') X(I+20) = UN(IS+I,11-I)
110   CONTINUE

      IF (TOOLLIFE(11).NE.0.0) THEN
        X(33) = 1.0
        X(34) = TOOLLIFE(11)
        IF (TOOLCHNG(I).EQ.'CO') X(32) = CO(2*IS+1)
        IF (TOOLCHNG(I).EQ.'EX') X(32) = EX(2*IS+1,5)
        IF (TOOLCHNG(I).EQ.'RN') X(32) = RN(2*IS+1,5)
        IF (TOOLCHNG(I).EQ.'TR') X(32) = TR(2*IS+1,5)
        IF (TOOLCHNG(I).EQ.'UN') X(32) = UN(2*IS+1,5)
      ENDIF
      X(31) = PARTCNT
      RETURN
      END




      FUNCTION UF(JOB,N)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUM
C
      INTEGER PMAX
      PARAMETER (PMAX=10)

      INTEGER*2 IS, CUTOFF
      CHARACTER*1 TOOLCHNG(PMAX+1)*2, LOADTIME(PMAX)*2
```

```
      REAL TOOLLIFE(PMAX+1), PROCTIME(PMAX+1),
     +     TOOLCHN1(PMAX+1), TOOLCHN2(PMAX+1), TOOLCHN3(PMAX+1),
     +     LOADTIM1(PMAX), LOADTIM2(PMAX), LOADTIM3(PMAX),
     +     OHCOST(PMAX), TOOLCOST(PMAX+1)

      COMMON/USER1/CUTOFF,IS,TOOLLIFE,PROCTIME
      COMMON/USER2/TOOLCHNG,TOOLCHN1,TOOLCHN2,TOOLCHN3
      COMMON/USER3/LOADTIME,LOADTIM1,LOADTIM2,LOADTIM3
      COMMON/USER4/OHCOST,TOOLCOST

C     REAL Y,L,C
      INTEGER N
C      Y=A(JOB,5)
C      C=X(N+20)*X(N+10)
C      L=A(JOB,2)

      IF (N.LE.10) THEN
        IF (A(JOB,2).GE.0.0) THEN
          UF=A(JOB,5)*OHCOST(N)/60.0+X(N+20)*X(N+10)*OHCOST(N)/60.0+
     *       X(N+10)*TOOLCOST(N)+A(JOB,3)*OHCOST(N)/60.0
        ELSE
          UF=A(JOB,5)*OHCOST(N)/60.0+X(32)*X(34)*OHCOST(N)/60.0+
     *       X(34)*TOOLCOST(11)+A(JOB,3)*OHCOST(N)/60.0
        ENDIF
      ELSE
        IF (A(JOB,2).GE.0.0) THEN
          IF (TOOLCHNG(N-PMAX).EQ.'CO') UF = CO(N-PMAX+IS)
          IF (TOOLCHNG(N-PMAX).EQ.'EX') UF = EX(N-PMAX+IS,21-N)
          IF (TOOLCHNG(N-PMAX).EQ.'RN') UF = RN(N-PMAX+IS,21-N)
          IF (TOOLCHNG(N-PMAX).EQ.'TR') UF = TR(N-PMAX+IS,21-N)
          IF (TOOLCHNG(N-PMAX).EQ.'UN') UF = UN(N-PMAX+IS,21-N)
        ELSE
          IF (TOOLCHNG(11).EQ.'CO') UF = CO(2*IS+1)
          IF (TOOLCHNG(11).EQ.'EX') UF = EX(2*IS+1,5)
          IF (TOOLCHNG(11).EQ.'RN') UF = RN(2*IS+1,5)
          IF (TOOLCHNG(11).EQ.'TR') UF = TR(2*IS+1,5)
          IF (TOOLCHNG(11).EQ.'UN') UF = UN(2*IS+1,5)
        ENDIF
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE WRAPUP
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUM
      COMMON/USER1/CUTOFF,IS,TOOLLIFE,PROCTIME

      PARAMETER (PMAX=10)

      INTEGER*2 CUTOFF, IS
      REAL TOOLLIFE(PMAX+1), PROCTIME(PMAX+1)

      OPEN(1,FILE='MASTER%%.TXT',FORM='FORMATTED')
      DO 100 I=1, IS
```

```
      WRITE(1,'(F10.2)') TAVG(I)
      IF (I.EQ.CUTOFF) WRITE (1,'(F10.2)') TAVG(11)
100   CONTINUE
      CLOSE(1)
      RETURN
      END
```

APPENDIX 3

FORTRAN Codes for Linking Paradox with Simulation Model

```
C MS-FORTRAN 77: SIMFILE.FOR
C
C This file contains one program. The purpose of this program is to retrieve
C information from the file generated from database applications to produce a
C SIMAN experimental frame for the simulation model.
C
C
C
C CREATE THE SIMAN EXPERIMENTAL FRAME AND FORTRAN SUBROUTINE
C
      PROGRAM SIMFILE
C
      INTEGER PMAX
      PARAMETER (PMAX=10)

      LOGICAL*1 CUTOFF
      INTEGER*2 IS, PARTCNT*4
      CHARACTER*1 END, LASTPROC*8, PROCID*8, TOOLCHNG(PMAX+1)*2,
     +          LOADTIME(PMAX)*2, CFCHNG*2

      REAL TOOLLIFE(PMAX+1), PROCTIME(PMAX+1),
     +     TOOLCHN1(PMAX+1), TOOLCHN2(PMAX+1), TOOLCHN3(PMAX+1),
     +     LOADTIM1(PMAX), LOADTIM2(PMAX), LOADTIM3(PMAX),
     +     OHCOST(PMAX), TOOLCOST(PMAX+1), A2VALUE(PMAX), CFCHN1,
     +     CFCHN2, CFCHN3

      DATA END, LASTPROC/':',' '/
      DATA IS,CUTOFF/0,.FALSE./
      DATA A2VALUE /PMAX*0.0/
C
C READ THE DATA FROM THE FILE MASTER.DAT
C
      OPEN (1,FILE='MASTER$$.TXT',FORM='FORMATTED')
      READ (1,'(I7)') PARTCNT
10    READ (1,'(A,F6.4,F6.1,A,3F6.1,A,3F6.1,2F8.2)',END=100) PROCID,
     +     TOOLLIFE(IS+1), PROCTIME(IS+1), TOOLCHNG(IS+1),
     +     TOOLCHN1(IS+1), TOOLCHN2(IS+1), TOOLCHN3(IS+1),
     +     LOADTIME(IS+1), LOADTIM1(IS+1), LOADTIM2(IS+1),
     +     LOADTIM3(IS+1), OHCOST(IS+1), TOOLCOST(IS+1)

      IF (PROCID .EQ. LASTPROC) THEN
        PROCTIME(IS) = PROCTIME(IS) + PROCTIME(IS+1)
        TOOLLIFE(IS) = TOOLLIFE(IS) + TOOLLIFE(IS+1)
      ELSE IF (PROCID.EQ.'CUT-OFF'.AND.LASTPROC.EQ.'TURNING') THEN
        CUTOFF = .TRUE.
        A2VALUE(IS) = PROCTIME(IS+1)
        CFCHNG = TOOLCHNG(IS+1)
        CFCHN1 = TOOLCHN1(IS+1)
        CFCHN2 = TOOLCHN2(IS+1)
        CFCHN3 = TOOLCHN3(IS+1)
      ELSE
        IS = IS + 1
        LASTPROC = PROCID
      ENDIF
```

```
      GOTO 10

100   CLOSE (1)
C
C WRITE TO THE EXPERIMENTAL FRAME MASTER.EXP
C
      OPEN (3, FILE='MASTER.EXP',FORM='FORMATTED')
      WRITE (3,'(A)') 'BEGIN;'
      WRITE (3,'(A)') 'PROJECT,MANUFACTURING PROCESS,SHARON,1/3/90;'
      WRITE (3,'(A)') 'DISCRETE,130,5,41,11;'
      WRITE (3,'(A)') 'TALLIES:1,WORK 1 COST:2,WORK 2 COST:'
      WRITE (3,'(8X,A)') '3,WORK 3 COST:4,WORK 4 COST:5,WORK 5 COST:'
      WRITE (3,'(8X,A)') '6,WORK 6 COST:7,WORK 7 COST:8,WORK 8 COST:'
      WRITE (3,'(8X,A)') '9,WORK 9 COST:10,WORK 10 COST:11,CUT OFF COST:
     +'
      WRITE (3,'(8X,A)') '12,TIME IN SYSTEM;'
      WRITE (3,'(A)') 'DSTAT:'
      DO 15 I=1,IS
15       WRITE (3,'(6X,3(I2,A))') I,',NQ(',I,'),QUEUE ',I,':'
      DO 16 I=1,IS
        IF (I.EQ.IS) END = ';'
        WRITE (3,'(6X,3(I2,A))') IS+I,',NR(',I,'),MACHINE ',I,END
16    CONTINUE
      WRITE (3,'(A)') 'RESOURCES:1-10,WORK;'
      WRITE (3,'(A)') 'COUNTERS:1,PART;'
      WRITE (3,'(A)') 'PARAMETERS:'

      DO 20 I=1,IS
        IF ((LOADTIME(I) .EQ. 'EX') .OR. (LOADTIME(I) .EQ. 'CO'))
     +     WRITE (3,'(11X,I2,A,F5.1,A)') I, ',', LOADTIM1(I),':'
        IF ((LOADTIME(I) .EQ. 'RN') .OR. (LOADTIME(I) .EQ. 'UN'))
     +     WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A)') I,',',LOADTIM1(I),',',
     +                                  LOADTIM2(I),':'
        IF (LOADTIME(I) .EQ. 'TR')
     +     WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A,F5.1,A)') I,',',
     +          LOADTIM1(I),',',LOADTIM2(I),',',LOADTIM3(I),':'
20    CONTINUE

      END = ':'
      DO 25 I=1,IS
        IF (I.EQ.IS.AND..NOT.CUTOFF) END = ';'
        IF ((TOOLCHNG(I) .EQ. 'EX') .OR. (TOOLCHNG(I) .EQ. 'CO'))
     +     WRITE (3,'(11X,I2,A,F5.1,A)') IS+I, ',', TOOLCHN1(I),END
        IF ((TOOLCHNG(I) .EQ. 'RN') .OR. (TOOLCHNG(I) .EQ. 'UN'))
     +     WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A)') IS+I,',',TOOLCHN1(I),
     +          ',',TOOLCHN2(I),END
        IF (TOOLCHNG(I) .EQ. 'TR')
     +     WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A,F5.1,A)') IS+I,',',
     +          TOOLCHN1(I),',',TOOLCHN2(I),',',TOOLCHN3(I),END
25    CONTINUE

      IF (CUTOFF) THEN
        IF ((CFCHNG .EQ. 'EX') .OR. (CFCHNG .EQ. 'CO'))
     +     WRITE (3,'(11X,I2,A,F5.1,A)') 2*IS+1, ',', CFCHN1,';'
        IF ((CFCHNG .EQ. 'RN') .OR. (CFCHNG .EQ. 'UN'))
     +     WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A)') 2*IS+1,',',CFCHN1,
     +          ',',CFCHN2,';'
        IF (CFCHNG .EQ. 'TR')
     +      WRITE (3,'(11X,I2,A,F5.1,A,F5.1,A,F5.1,A)') 2*IS+1,',',
     +          CFCHN1,',',CFCHN2,',',CFCHN3,';'
```

```
      ENDIF

      WRITE (3,'(A)') 'SEQUENCES:1,'

      DO 30 I=1,IS
        IF (LOADTIME(I).EQ.'CO') THEN
          WRITE (3,'(I2,A,F6.2,A,F6.2,A,A,A,I2,A,I2,A,A,A,I2,A)')
     -     I,',',PROCTIME(I),',',A2VALUE(I),',',LOADTIME(I),'(',I,')/'
        ELSE
          WRITE (3,'(I2,A,F6.2,A,F6.2,A,A,A,I2,A,I2,A,A,A,I2,A,I2,A)')
     -     I,',',PROCTIME(I),',',A2VALUE(I),',',LOADTIME(I),'(',I,',',
     -     I,')/'
        ENDIF
30    CONTINUE

      WRITE (3,'(A)') '11,0;'
      WRITE (3,'(A)') 'REPLICATE,1;'
      WRITE (3,'(A)') 'END;'
      CLOSE (3)

      END
```

APPENDIX 4

DOS Batch File for Executing Programs

```
REM  MS-DOS batch file: SIMULATE.BAT
REM
REM  The batch file perform several tasks under DOS environment to
REM  execute the simulation run.
REM
REM  1)  Generate the experimental frame from database output for
REM      use of simulation runs.
REM  2)  Compile the experimental frame and link output with model
REM      frame from system.
REM  3)  Call system to execute the simulation.
REM
     @echo off
     if not exist master$$.TXT goto :end
REM
REM            Create master.exp
               simfile
REM
REM            Compile the source.
               expmt master.exp master.e
REM
REM            Route the next job.
               if not errorlevel 1 goto :ok2
REM
REM            Show the file with the error and leave.
               pause
               qedit master.exp
               goto :end
REM
     :ok2
REM
REM            Link the model and the experimental frames.
               linker master master.e master.p
REM
REM            Route the next job.
               if not errorlevel 1 goto :ok3
REM
REM            Leave due to the error while linking.
               goto :end
REM
     :ok3
REM
REM            Clean up files used for starting simulation run.
               if exist master.sum del master.sum
               cls
               echo Please hit return once to pass the initial banner.
               echo Please wait..
               master master.p > master.sum
REM
REM            Clean up files generated from execution process.
               if exist master.exp   del master.exp
               if exist master.e     del master.e
               if exist master.p     del master.p
               if exist errors.txt   del errors.txt
               if exist master$$.txt del master$$.txt
               if exist master.txt   del master.txt
```

```
:end
echo on
```

APPENDIX 5

DOS Batch File for Viewing Simulation Results

```
REM  MS-DOS batch file: VIEWRPT.BAT
REM
REM  The batch file perform one task under DOS environment to
REM  view output of the simulation run.
REM
REM
        @echo off
        cls
        if not exist master.sum goto no_file
                list master.sum
                goto end
        :no_file
                echo No simulation run has been executed.
                pause
        :end
        echo on
```

APPENDIX 6

Database Program

```
; Paradox 3.0: CENTRAL.SC
;
; This program performs the functions required for query manager and
; preprocessing module in the designed system.  The user screens and
; data manipulation are implemented within these procedures.
;
;
;
;=====================================================================
PROC OnScreen(string)
;=====================================================================
;
;     Display the messages on screen for any required situations.
;
;  Private Y
   CANVAS OFF
   CLEAR
   PAINTCANVAS ATTRIBUTE 31 0,0,24,79
   STYLE ATTRIBUTE 30
   @8,0
   TEXT

   ENDTEXT
   @11, 20 ?? string
   STYLE BLINK
   STYLE ATTRIBUTE 30
   @11, 20+LEN(string) ?? "..."
   CANVAS ON
   STYLE
ENDPROC
;
;
;
;=====================================================================
PROC DistMenu()
;=====================================================================
;
;     Display the menu choices of five statistical distributions for
;     use of SIMAN.
;
   PRIVATE dist
   dist = "Esc"
   SHOWMENU "CO" : "Constant distribution",
            "EX" : "Exponential distribution",
            "RN" : "Normal distribution",
            "TR" : "Triangular distribution",
            "UN" : "Uniform distribution"
   TO dist
```

```
   RETURN dist
   RELEASE VARS dist
ENDPROC
;
;
;
;
;===========================================================================
PROC ShowRecord (file_name, field1, field2)
;===========================================================================
;     Given the file which contains the records for field1 and
;     field2 shown in the menu.
;     Return the choice1 or "esc" if "esc" is pressed to quit.
;
   PRIVATE choice1, rec, line1, line2, record
   VIEW file_name
   rec = NRECORDS(file_name)
   choice1 = "Esc"
   Array line1[rec]
   Array line2[rec]
   SCAN
      COPYTOARRAY record
      line1[RECNO()] = STRVAL(record[field1])
      IF NOT ISBLANK(field2)
         THEN line2[RECNO()] = record[field2]
         ELSE line2[RECNO()] = " "
      ENDIF
   ENDSCAN
   SHOWARRAY
      line1 line2
   TO choice1
   CLEARIMAGE
   RETURN choice1
ENDPROC
;
;
;
;
;===========================================================================
PROC ToolRecordMenu(machine)
;===========================================================================
;
;     Display the available tools for a given process.
;     Return the choice1 or "esc" if "esc" is pressed to
;         quit choosing.
;
   PRIVATE choice1, rec, line1, line2, record, avlrec
   VIEW "tool"
   rec = NRECORDS("tool")
   avlrec = 0
   choice1 = "Esc"
   SCAN
      COPYTOARRAY record
      IF record[machine] = 1
         THEN avlrec = avlrec + 1
      ENDIF
   ENDSCAN
   IF avlrec > 0
      THEN Array line1[avlrec]
           Array line2[avlrec]
```

```
    ENDIF
    avlrec = 0
    SCAN
       COPYTOARRAY record
       IF record[machine] = 1
          THEN avlrec = avlrec + 1
               line1[avlrec] = record["Id #"]
               line2[avlrec] = record["Tool material"]
       ENDIF
    ENDSCAN
    choice1 = "Esc"
    IF avlrec >= 1
       THEN SHOWARRAY
               line1 line2
            TO choice1
       ELSE MESSAGE "No tool on hand is available."
            SLEEP 3000
    ENDIF
    CLEARIMAGE
    RETURN choice1
ENDPROC
;
;
;
;
;==========================================================================
PROC TaylorC(tool)
;==========================================================================
;
;     Perform the query for the Taylor constant c which matches the tool
;     material.
;
   Query

    Tool  |  Id #   | Taylor's c |
          | ~tool   | Check      |
          |         |            |
          |         |            |

   Endquery
   DO_IT!
   UPIMAGE UPIMAGE CLEARIMAGE DOWNIMAGE
   IF (ISEMPTY("answer"))
      THEN RETURN " "
   ENDIF
   RETURN [Taylor's c]
ENDPROC
;
;
;
;
;==========================================================================
PROC TaylorN(tool,work)
;==========================================================================
;
;     Perform the query for Taylor constant n which matches the tool and
;     the work materials.
;
   Query
```

```
    Const_n | Tool material | Work material | Taylor's n |
            | ~tool         | ~work         | Check      |
            |               |               |            |
            |               |               |            |

   Endquery
   DO_IT!
   UPIMAGE UPIMAGE CLEARIMAGE DOWNIMAGE
   IF (ISEMPTY("answer"))
      THEN RETURN " "
   ENDIF
   RETURN [Taylor's n]
ENDPROC
;
;
;
;
;==========================================================================
PROC UnitHP(work)
;==========================================================================
;
;     Perform the query for unit horsepower which matches work material.
;
   Query

    Work | Id #  | Unit hp |
         | ~work | Check   |
         |       |         |
         |       |         |

   Endquery
   DO_IT!
   UPIMAGE UPIMAGE CLEARIMAGE DOWNIMAGE
   IF (ISEMPTY("answer"))
      THEN RETURN -1.0
   ENDIF
   RETURN [Unit hp]
ENDPROC
;
;
;
;
;==========================================================================
PROC Sel_Turning (hpU)
;==========================================================================
;
;     Calculate the processing times when turning process
;     is selected.
;
   PRIVATE depth, no_of_cut, cut_time, process
   PRIVATE wtemp, ttemp, q, time_per_rev, tool_life
;
   IF ISBLANK([Work material]) OR
      ISBLANK([Tool material]) OR
      ISBLANK([Machine power]) OR
      ISBLANK([Machine efficiency]) OR
      ISBLANK([Dimension 1]) OR
      ISBLANK([Dimension 2]) OR
      ISBLANK([Dimension 3])
      THEN RETURN
```

```
ENDIF
EDITKEY
COPYTOARRAY process
DO_IT!
wtemp = process["Work material"]
ttemp = process["Tool material"]
SETQUERYORDER IMAGEORDER
Query

 Turning | Work material  | Tool material  |
         | ~wtemp         | ~ttemp         |
         |                |                |
         |                |                |

 Turning | Depth of cut     | Cutting speed | Feed rate |
         | CheckDescending  | Check         | Check     |
         |                  |               |           |
         |                  |               |           |

 Turning | Combination #  |
         | Check          |
         |                |
         |                |

Endquery
DO_IT!
SETQUERYORDER TABLEORDER
CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
proctime = 0.0
procset = ""
tool_life = 0.0
IF NOT ISEMPTY("answer")
  THEN VIEW "answer"
       init_rad = (process["Dimension 2"] -
                   process["Dimension 3"])/2.
       q = process["Machine power"]*process["Machine efficiency"]/hpU
       WHILE NOT EOT()
         IF 12.0*[Depth of cut]*[Feed rate]*[Cutting speed] <= q
            THEN end_rad = MOD(init_rad, [Depth of cut])
                 no_of_cut = INT(init_rad/[Depth of cut])
                 IF ATLAST() AND end_rad > 0.0
                    THEN no_of_cut = no_of_cut + 1
                 ENDIF
                 IF no_of_cut >= 1
                    THEN cut_time = 1.25*(process["Dimension 1"]+0.5)
                                    *PI()*process["Dimension 2"]
                         time_per_rev = no_of_cut*cut_time /
                                         (12.0*[Feed rate]*
                                          [Cutting speed])
                         proctime = proctime + time_per_rev
                         procset = procset + [Combination #]
                         tool_life = tool_life + time_per_rev /
                                     ((LN(process["Taylor's param c"])
                                       - LN([Cutting speed])) /
                                       process["Taylor's param n"])
                 ENDIF
                 init_rad = end_rad
         ENDIF
         MESSAGE proctime
         SLEEP 2000
```

```
            SKIP 1
          ENDWHILE
   ENDIF
   CLEARIMAGE
   process["Process time"] = proctime
   process["Process set"] = procset
   process["Tool life"] = tool_life
   EDITKEY
   COPYFROMARRAY process
   DO_IT!
ENDPROC
;
;
;
;
;
;==================================================================
PROC Sel_Milling (hpU)
;==================================================================
;
;     Calculate the processing times when milling
;     process is selected.
;
   PRIVATE depth, no_of_cut, cut_time, process
   PRIVATE wtemp, ttemp, q, ptemp, time_per_rev, tool_life
   IF ISBLANK([Work material]) OR
      ISBLANK([Tool material]) OR
      ISBLANK([Machine power]) OR
      ISBLANK([Machine efficiency]) OR
      ISBLANK([Dimension 1]) OR
      ISBLANK([Dimension 2]) OR
      ISBLANK([Dimension 3]) OR
      ISBLANK([Tool parameter 1]) OR
      ISBLANK([Tool parameter 2])
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY process
   DO_IT!
   wtemp = process["Work material"]
   ttemp = process["Tool material"]
   ptemp = process["Tool parameter 1"]
   SETQUERYORDER IMAGEORDER
   Query

    Milling | Work material | Tool material | Diameter of cutter|
            | ~wtemp        | ~ttemp        | ~ptemp            |
            |               |               |                   |
            |               |               |                   |

    Milling | Depth of cut    | Cutting speed | Feed rate |
            | CheckDescending | Check         | Check     |
            |                 |               |           |
            |                 |               |           |

    Milling | Combination # |
            | Check         |
            |               |
            |               |
```

```
    Endquery
    DO_IT!
    SETQUERYORDER TABLEORDER
    CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
    proctime = 0.0
    procset = ""
    tool_life = 0.0
    IF NOT ISEMPTY("answer")
      THEN VIEW "answer"
           init_rad = process["Dimension 3"]
           q = process["Machine power"] * process["Machine efficiency"]
                / hpU
           WHILE NOT EOT()
             IF 12.0*[Depth of cut]*[Feed rate]*[Cutting speed] *
                process["Tool parameter 2"]/PI() <= q
                THEN end_rad = MOD(init_rad, [Depth of cut])
                     no_of_cut = INT(init_rad/[Depth of cut])
                     IF ATLAST() AND end_rad > 0.0
                        THEN no_of_cut = no_of_cut + 1
                     ENDIF
                     IF no_of_cut >= 1
                        THEN cut_time = ((2.0+process["Dimension 1"] /
                                       (0.75*process["Tool parameter 1"]))
                                        *process["Dimension 2"] +
                                        process["Tool parameter 1"])
                                       *PI()*process["Tool parameter 1"]
                             time_per_rev = no_of_cut*cut_time/
                                       (12.0*[Feed rate]*[Cutting speed]
                                            *process["Tool parameter 2"])
                             proctime = proctime + time_per_rev
                             procset = procset + [Combination #]
                             tool_life = tool_life + time_per_rev/
                                         ((LN(process["Taylor's param c"])
                                           - LN([Cutting speed]))/
                                          process["Taylor's param n"])
                     ENDIF
                     init_rad = end_rad
             ENDIF
             MESSAGE proctime
             SLEEP 2000
             SKIP 1
           ENDWHILE
      ELSE MESSAGE "No matched process"
           SLEEP 2000
    ENDIF
    CLEARIMAGE
    process["Process time"] = proctime
    process["Process set"] = procset
    process["Tool life"] = tool_life
    EDITKEY
    COPYFROMARRAY process
    DO_IT!
ENDPROC
;
;
;
;
;
;
;
```

```
;================================================================
PROC Sel_Drilling (hpU)
;================================================================
;
;     Calculate the processing times when drilling
;     process is selected.
;
   PRIVATE depth, no_of_cut, cut_time, process, wtemp, ttemp
   PRIVATE q, f_factor, c_factor, feed_rate, cutting_speed, gap
   IF ISBLANK([Work material]) OR
      ISBLANK([Tool material]) OR
      ISBLANK([Machine power]) OR
      ISBLANK([Machine efficiency]) OR
      ISBLANK([Dimension 1]) OR
      ISBLANK([Tool parameter 1])
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY process
   DO_IT!
   wtemp = process["Work material"]
   ttemp = process["Tool material"]
   ptemp = process["Tool parameter 1"]
   SETQUERYORDER IMAGEORDER


   Query

    Drilling | Work material | Tool material | Diameter of cutter|
             | ~wtemp        | ~ttemp        | ~ptemp            |
             |               |               |                   |
             |               |               |                   |

    Drilling | Cutting speed | Feed rate | Combination # |
             | Check         | Check     | Check         |
             |               |           |               |
             |               |           |               |

   Endquery
   DO_IT!
   SETQUERYORDER TABLEORDER
   CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
   proctime = 0.0
   procset = ""
   gap = process["Tool parameter 1"]/(2.0*TAN(PI()*118.0/360.0))
   IF NOT ISEMPTY("answer")
     THEN VIEW "answer"
          f_factor = 0
          c_factor = 0
          feed_rate = [Feed rate]
          cutting_speed = [Cutting speed]
          q =  process["Machine power"]*process["Machine efficiency"] /
               hpU
          WHILE NOT (f_factor > 5 AND c_factor >  5)
            IF 12.0*[Feed rate]*POW(0.95,f_factor)*[Cutting speed]
               *POW(0.95,c_factor)*process["Tool parameter 1"]/4.0 <= q
               THEN proctime = 1.25*(process["Dimension 1"] + gap + 0.5)
                               * PI() * process["Tool parameter 1"]/
                               (12.0*[Feed rate]*POW(0.95,f_factor) *
```

```
                                [Cutting speed]*POW(0.95,c_factor))
                        procset = [Combination #]
                        process["Process time"] = proctime
                        process["Process set"] = procset
                        process["Tool life"] = proctime /
                                       ((LN(process["Taylor's param c"])
                                         - LN([Cutting speed] *
                                       POW(0.95,c_factor))) /
                                       process["Taylor's param n"])
                        CLEARIMAGE
                        EDITKEY
                        COPYFROMARRAY process
                        DO_IT!
                        QUITLOOP
                    ELSE IF f_factor <= c_factor
                            THEN f_factor = f_factor + 1
                            ELSE c_factor = c_factor + 1
                        ENDIF
                ENDIF
            ENDWHILE
            IF f_factor > 5 AND c_factor > 5
               THEN MESSAGE
                    "Machine power is too small for the matched process."
                    SLEEP 3000
             ENDIF
        ELSE
            MESSAGE "No matched process."
            SLEEP 2000
   ENDIF
ENDPROC
;
;
;
;
;=================================================================
PROC Sel_Grinding (hpU)
;=================================================================
;
;    Calculate the processing times when grinding
;    process is selected.
;
   PRIVATE depth, no_of_cut, cut_time, process, wtemp, ttemp, q
   PRIVATE cfeed, air_time, time_per_rev, tool_life
   IF ISBLANK([Work material]) OR
      ISBLANK([Tool material]) OR
      ISBLANK([Machine power]) OR
      ISBLANK([Machine efficiency]) OR
      ISBLANK([Dimension 1]) OR
      ISBLANK([Dimension 2]) OR
      ISBLANK([Dimension 3]) OR
      ISBLANK([Tool parameter 1])
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY process
   DO_IT!
   wtemp = process["Work material"]
   ttemp = process["Tool material"]
   SETQUERYORDER IMAGEORDER
```

```
Query

 Grinding | Work material  | Tool material |
          | ~wtemp         | ~ttemp        |
          |                |               |
          |                |               |

 Grinding | Downfeed         | Table speed | Crossfeed |
          | CheckDescending  | Check       | Check     |
          |                  |             |           |
          |                  |             |           |

 Grinding | Rate of wheel | Combination # |
          | Check         | Check         |
          |               |               |
          |               |               |

Endquery
DO_IT!
SETQUERYORDER TABLEORDER
CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
proctime = 0.0
procset = ""
tool_life = 0.0
IF NOT ISEMPTY("answer")
  THEN VIEW "answer"
       init_rad = process["Dimension 3"]
       q =  process["Machine power"]*process["Machine efficiency"] /
            hpU
       WHILE NOT EOT()
         cfeed = MIN([Crossfeed],process["Tool parameter 1"]
                                 *[Rate of wheel])
         IF 12.0*[Downfeed]*[Table speed]*cfeed/6.0 <= q
            THEN end_rad = MOD(init_rad, [Downfeed])
                 no_of_cut = INT(init_rad/[Downfeed])
                 IF ATLAST() AND end_rad > 0.0
                    THEN no_of_cut = no_of_cut + 1
                 ENDIF
                 IF no_of_cut >= 1
                    THEN cut_time = (process["Dimension 2"] +
                                     process["Tool parameter 1"])
                                     * 2.0 * process["Dimension 1"] /
                                     (12.0*cfeed*[Table speed])
                         air_time = (process["Dimension 2"] +
                                     process["Tool parameter 1"]) /
                                    [Table speed]
                         time_per_rev = no_of_cut * cut_time
                         proctime = proctime +
                                    no_of_cut * (cut_time + air_time)
                         procset = procset + [Combination #]
                         tool_life = tool_life + time_per_rev /
                                   ((LN(process["Taylor's param c"])
                                     - LN([Table speed])) /
                                    process["Taylor's param n"])
                 ENDIF
                 init_rad = end_rad
         ENDIF
         MESSAGE proctime
         SLEEP 2000
         SKIP 1
```

```
          ENDWHILE
       ELSE MESSAGE "No matched process"
          SLEEP 2000
    ENDIF
    CLEARIMAGE
    process["Process time"] = proctime
    process["Process set"] = procset
    process["Tool life"] = tool_life
    EDITKEY
    COPYFROMARRAY process
    DO_IT!
ENDPROC
;
;
;
;
;==========================================================================
PROC Sel_Shaping (hpU)
;==========================================================================
;
;     Calculate the processing times when shaping process is selected.
;
    PRIVATE depth, no_of_cut, cut_time, process, wtemp, ttemp
    PRIVATE q, tool_life
    IF ISBLANK([Work material]) OR
       ISBLANK([Tool material]) OR
       ISBLANK([Machine power]) OR
       ISBLANK([Machine efficiency]) OR
       ISBLANK([Dimension 1]) OR
       ISBLANK([Dimension 2]) OR
       ISBLANK([Dimension 3])
       THEN RETURN
    ENDIF
    EDITKEY
    COPYTOARRAY process
    DO_IT!
    wtemp = process["Work material"]
    ttemp = process["Tool material"]
    SETQUERYORDER IMAGEORDER
    Query

     Shaping | Work material | Tool material |
             | ~wtemp        | ~ttemp        |
             |               |               |

     Shaping | Depth of cut    | Cutting speed | Feed rate |
             | CheckDescending | Check         | Check     |
             |                 |               |           |

     Shaping | Combination # |
             | Check         |
             |               |

    Endquery
    DO_IT!
    SETQUERYORDER TABLEORDER
    CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
```

```
    proctime = 0.0
    procset = ""
    tool_life = 0.0
    IF NOT ISEMPTY("answer")
      THEN VIEW "answer"
           init_rad = process["Dimension 3"]
           q =  process["Machine power"]*process["Machine efficiency"] /
                hpU
           WHILE NOT EOT()
             IF 12.0*[Depth of cut]*[Feed rate]*[Cutting speed] <= q
                THEN end_rad = MOD(init_rad, [Depth of cut])
                     no_of_cut = INT(init_rad/[Depth of cut])
                     IF ATLAST() AND end_rad > 0.0
                        THEN no_of_cut = no_of_cut + 1
                     ENDIF
                     IF no_of_cut >= 1
                        THEN cut_time = 1.25*(process["Dimension 1"] +
                                        0.5) * process["Dimension 2"] /
                                        ([Feed rate] * [Cutting speed])
                             proctime = proctime + no_of_cut*cut_time
                             procset = procset + [Combination #]
                             tool_life = tool_life+no_of_cut*cut_time/
                                       ((LN(process["Taylor's param c"])
                                        - LN([Cutting speed])) /
                                       process["Taylor's param n"])
                     ENDIF
                     init_rad = end_rad
             ENDIF
             MESSAGE proctime
             SLEEP 2000
             SKIP 1
           ENDWHILE
      ELSE MESSAGE "No matched process"
           SLEEP 2000
    ENDIF
    CLEARIMAGE
    process["Process time"] = proctime
    process["Process set"] = procset
    process["Tool life"] = tool_life
    EDITKEY
    COPYFROMARRAY process
    DO_IT!
ENDPROC
;
;
;
;
;===========================================================================
PROC Sel_Cut_Off (hpU)
;===========================================================================
;
;     Calculate the processing times when cut-off
;     process is selected.
;
   PRIVATE depth, no_of_cut, cut_time, process, wtemp, ttemp, q
   IF ISBLANK([Work material]) OR
      ISBLANK([Tool material]) OR
      ISBLANK([Machine power]) OR
      ISBLANK([Machine efficiency]) OR
      ISBLANK([Dimension 1])
```

```
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY process
   DO_IT!
   wtemp = process["Work material"]
   ttemp = process["Tool material"]
   SETQUERYORDER IMAGEORDER
   Query

    Cut-off | Work material | Tool material |
            | ~wtemp        | ~ttemp        |
            |               |               |

    Cut-off | Cutting speed | Feed rate | Combination # |
            | Check         | Check     | Check         |
            |               |           |               |

   Endquery
   DO_IT!
   SETQUERYORDER TABLEORDER
   CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
   proctime = 0.0
   procset = ""
   IF NOT ISEMPTY("answer")
     THEN VIEW "answer"
          cut_time = 1.25*(process["Dimension 1"]+0.5) * 2 *
                     PI()*process["Dimension 1"]
          cut_time = cut_time/(12.0*[Feed rate]*[Cutting speed])
          process["Process time"] = cut_time
          process["Process set"] = [Combination #]
          process["Tool life"] = cut_time /
                                 ((LN(process["Taylor's param c"]) -
                                   LN([Cutting speed])) /
                                  process["Taylor's param n"])
          MESSAGE cut_time
          SLEEP 2000
   ENDIF
   CLEARIMAGE
   EDITKEY
   COPYFROMARRAY process
   DO_IT!
ENDPROC
;
;
;
;
;
;==========================================================================
PROC DoEdit()
;==========================================================================
;
;     Display the proper screen for editing of the processes.
;
   PRIVATE lprocess, cprocess, wwtemp, tmtemp
   VIEW "Process"
   rec_no = 1
   WHILE(True)                   ; change to switch/case
```

```
IF [Process id] = "TURNING"
   THEN PICKFORM 3
ENDIF
IF [Process id] = "MILLING"
   THEN PICKFORM 4
ENDIF
IF [Process id] = "DRILLING"
   THEN PICKFORM 5
ENDIF
IF [Process id] = "SHAPING"
   THEN PICKFORM 6
ENDIF
IF [Process id] = "GRINDING"
   THEN PICKFORM 7
ENDIF
IF [Process id] = "CUT-OFF"
   THEN PICKFORM 8
ENDIF
WAIT RECORD
   PROMPT "[F9] to edit, <ESC> to quit",
          "[PgDn] to next process, [PgUp] to previous process"
UNTIL "F9","PGDN","PGUP","Esc"
SWITCH
   CASE retval = "Esc" : QUITLOOP
   CASE retval = "PGDN" :
        IF (ATLAST())
           THEN BEEP
           ELSE rec_no = rec_no + 1 PGDN
        ENDIF
   CASE retval = "PGUP" :
        IF (ATFIRST())
           THEN BEEP
           ELSE rec_no = rec_no - 1 PGUP
        ENDIF
   CASE retval = "F9" :
        COPYTOARRAY lprocess
        EDITKEY
        WHILE TRUE
           WAIT RECORD
              PROMPT
     "Press F2 to save the editing, Esc to quit without saving",
     "^XF3, ^XF4, and ^XF5 to pop up menu for choices"
           UNTIL "F2", "Esc", "F13", "F14", "F15", "F16"
           SWITCH
              CASE retval = "F16":
                   IF FIELD() = "Tool change dist" OR
                      FIELD() = "Loading dist"
                      THEN  result = DistMenu()
                            IF result <> "Esc"
                               THEN [] = result
                            ENDIF
                      ELSE BEEP MESSAGE
                              "Not Applicable!" SLEEP 2000
                   ENDIF
              CASE retval = "F15":
                   IF FIELD() = "Tool parameter 1"
                      THEN  wwtemp=[Work material]
                            tmtemp=[Tool material]
                            DO_IT!
                            Query
```

```
                  Milling | Work material   | Tool material |
                          | ~wwtemp         | ~tmtemp       |
                          |                 |               |
                          |                 |               |

                  MILLING | Diameter of cutter |
                          | Check COUNT>=1     |
                          |                    |
                          |                    |

                     Endquery
                     DO_IT!
                     CLEARIMAGE UPIMAGE UPIMAGE CLEARIMAGE
                     result = ShowRecord ("answer",
                              "Diameter of cutter","")
                     EDITKEY
                     IF result <> "Esc"
                        THEN [] = BLANKNUM()
                             TYPEIN result
                     ENDIF
                     DO_IT!
                     EDITKEY
                     IF [Process id] = "TURNING"
                        THEN PICKFORM 3
                     ENDIF
                     IF [Process id] = "MILLING"
                        THEN PICKFORM 4
                     ENDIF
                     IF [Process id] = "DRILLING"
                        THEN PICKFORM 5
                     ENDIF
                     IF [Process id] = "SHAPING"
                        THEN PICKFORM 6
                     ENDIF
                     IF [Process id] = "GRINDING"
                        THEN PICKFORM 7
                     ENDIF
                     IF [Process id] = "CUT-OFF"
                        THEN PICKFORM 8
                     ENDIF
                ELSE BEEP MESSAGE
                          "Not Applicable!" SLEEP 2000
            ENDIF
        CASE retval = "F14":
            IF FIELD() = "Tool material"
               THEN  DO_IT!
                     result = ToolRecordMenu([Process id])
                     EDITKEY
                     IF result <> "Esc"
                        THEN [] = result
                     ENDIF
                     DO_IT!
                     EDITKEY
                     IF [Process id] = "TURNING"
                        THEN PICKFORM 3
                     ENDIF
                     IF [Process id] = "MILLING"
                        THEN PICKFORM 4
                     ENDIF
                     IF [Process id] = "DRILLING"
```

```
                    THEN PICKFORM 5
                 ENDIF
                 IF [Process id] = "SHAPING"
                    THEN PICKFORM 6
                 ENDIF
                 IF [Process id] = "GRINDING"
                    THEN PICKFORM 7
                 ENDIF
                 IF [Process id] = "CUT-OFF"
                    THEN PICKFORM 8
                 ENDIF
           ELSE BEEP MESSAGE
                    "Not Applicable!" SLEEP 2000
      ENDIF
CASE retval = "F13":
      IF FIELD() = "Work material"
         THEN  DO_IT!
               result = ShowRecord ("work","Id #",
                         "Work material")
               EDITKEY
               IF result <> "Esc"
                  THEN [] = result
               ENDIF
               DO_IT!
               EDITKEY
               IF [Process id] = "TURNING"
                  THEN PICKFORM 3
               ENDIF
               IF [Process id] = "MILLING"
                  THEN PICKFORM 4
               ENDIF
               IF [Process id] = "DRILLING"
                  THEN PICKFORM 5
               ENDIF
               IF [Process id] = "SHAPING"
                  THEN PICKFORM 6
               ENDIF
               IF [Process id] = "GRINDING"
                  THEN PICKFORM 7
               ENDIF
               IF [Process id] = "CUT-OFF"
                  THEN PICKFORM 8
               ENDIF
         ELSE BEEP MESSAGE
                    "Not Applicable!" SLEEP 2000
      ENDIF
CASE retval = "Esc":
      COPYFROMARRAY lprocess
      DO_IT!
      QUITLOOP
CASE retval = "F2":
      COPYTOARRAY cprocess
      DO_IT!
      IF ISBLANK(cprocess["Taylor's param n"]) OR
         cprocess["Work material"] <>
         lprocess["Work material"] OR
         cprocess["Tool material"] <>
         lprocess["Tool material"]
         THEN cprocess["Taylor's param n"] =
              TaylorN(cprocess["Tool material"],
```

```
              cprocess["Work material"])
         CLEARIMAGE     ; clear table answer
ENDIF
IF ISBLANK(cprocess["Taylor's param c"]) OR
   cprocess["Tool material"] <>
   lprocess["Tool material"]
   THEN cprocess["Taylor's param c"] =
        TaylorC(cprocess["tool material"])
        CLEARIMAGE     ; clear table answer
ENDIF
EDITKEY
COPYFROMARRAY cprocess
DO_IT!
IF ISBLANK(cprocess["Process time"]) OR
   cprocess["Work material"] <>
   lprocess["Work material"] OR
   cprocess["Tool material"] <>
   lprocess["Tool material"] OR
   cprocess["Machine power"] <>
   lprocess["Machine power"] OR
   cprocess["Machine efficiency"] <>
   lprocess["Machine efficiency"] OR
   cprocess["Dimension 1"] <>
   lprocess["Dimension 1"] OR
   cprocess["Dimension 2"] <>
   lprocess["Dimension 2"] OR
   cprocess["Dimension 3"] <>
   lprocess["Dimension 3"] OR
   cprocess["Tool parameter 1"] <>
   lprocess["Tool parameter 1"] OR
   cprocess["Tool parameter 2"] <>
   lprocess["Tool parameter 2"]
   THEN hp_unit =
        UnitHP(cprocess["Work material"])
       CLEARIMAGE     ; clear table answer
       IF cprocess["Process id"]="TURNING" AND
          hp_unit > 0.0
          THEN Sel_Turning (hp_unit)
       ENDIF
       IF cprocess["Process id"]="MILLING" AND
          hp_unit > 0.0
          THEN Sel_Milling (hp_unit)
       ENDIF
       IF cprocess["Process id"]="DRILLING" AND
          hp_unit > 0.0
          THEN Sel_Drilling (hp_unit)
       ENDIF
       IF cprocess["Process id"]="GRINDING" AND
          hp_unit > 0.0
          THEN Sel_Grinding (hp_unit)
       ENDIF
       IF cprocess["Process id"]="SHAPING" AND
          hp_unit > 0.0
          THEN Sel_Shaping (hp_unit)
       ENDIF
       IF cprocess["Process id"]="CUT-OFF" AND
          hp_unit > 0.0
          THEN Sel_Cut_Off (hp_unit)
       ENDIF
ENDIF
```

```
                        QUITLOOP
                    ENDSWITCH
                ENDWHILE
      ENDSWITCH
   ENDWHILE
   RETURN
ENDPROC
;
;
;
;
;==============================================================================
PROC DoCopy()
;==============================================================================
;
;     Perform the function of copying existed process.
;
   PRIVATE temppro, tmprec
   WAIT RECORD
        PROMPT "Press RETURN to confirm the copying from this process",
               "<ESC> to quit without copying"
   UNTIL "Enter", "Esc"
   IF retval = "Esc"
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY tmppro
   @0,0 ?? "Enter record # the process is copying to"
   @1,0 ACCEPT "N" MIN 1 TO tmprec
   IF tmprec > NRECORDS("Process")
      THEN END ENTER
      ELSE MOVETO RECORD tmprec INS
   ENDIF
   COPYFROMARRAY tmppro
   DO_IT!
   RETURN
ENDPROC
;
;
;
;
;==============================================================================
PROC DoMove()
;==============================================================================
;
;     Perform the function of moving the existed process.
;
   PRIVATE tmppro, tmprec
   WAIT RECORD
        PROMPT "Press RETURN to confirm the moving for this process",
               "<ESC> to quit"
   UNTIL "Enter", "Esc"
   IF retval = "Esc"
      THEN RETURN
   ENDIF
   EDITKEY
   COPYTOARRAY tmppro
   DEL
   @0,0 ?? "Enter the record # the process is moving to"
   @1,0 ACCEPT "N" MIN 1 TO tmprec
```

```
   IF tmprec > NRECORDS("Process")
      THEN END ENTER
      ELSE MOVETO RECORD tmprec INS
   ENDIF
   COPYFROMARRAY tmppro
   DO_IT!
   RETURN
ENDPROC
;
;
;
;
;==========================================================================
PROC DoSetup()
;==========================================================================
;
;     Display the functions available in setup section.
;
   PRIVATE wmaterial, setdefa, change, machine_id
   CLEAR STYLE
   wmaterial = "Esc"
   WHILE TRUE
      STYLE ATTRIBUTE 31
      @0, 0 ?? "Setup work material for all processes:
                        "
      @1, 0 ?? "Press <Enter> to continue, <Esc> to bypass.
                        "
      @0, 40 ACCEPT "A5" TO setdefa
      SWITCH
         CASE retval = "Esc" : QUITLOOP
         CASE setdefa = "" :
              OnScreen("Choose the default work material")
              wmaterial = ShowRecord ("work","Id #","Work material")
              QUITLOOP
         OTHERWISE :
              BEEP BEEP
              MESSAGE "Unrecognized key!"
              SLEEP 2000
      ENDSWITCH
   ENDWHILE
   CLEAR STYLE
   IF ISEMPTY("process")
      THEN CANVAS OFF
           CLEARALL CLEAR
           PAINTCANVAS ATTRIBUTE 127 0,0,24,79
           STYLE ATTRIBUTE 112
           @5,0
;
           TEXT
                          ╔═ Manufacturing Process ═╗
                          ║                         ║
                          ║  Sequence     Process   ║
                          ║  --------    ---------  ║
                          ║      1                  ║
                          ║                         ║
                          ║                         ║
                          ║                         ║
                          ║                         ║
                          ╚═════════════════════════╝
```

```
            ENDTEXT
;           Paint "Manufacturing Process"
            PAINTCANVAS ATTRIBUTE 31 5,28,5,48
            CANVAS ON
            STYLE
            machine_id = ShowRecord ("Machine","Machine id", "")
            CLEAR
            IF machine_id <> "Esc"
               THEN EDIT "process"
                    [Process id] = machine_id
                    DO_IT!
                    CLEARIMAGE
               ELSE RETURN
            ENDIF
    ENDIF
    IF NOT ISEMPTY("process")
       THEN VIEW "process"
            MOVETO [Process id]
            PICKFORM 2
            WHILE (TRUE)
               WAIT RECORD

                    PROMPT
"[F9] to make change on this process, Esc to quit with records saved.",
"Arrow UP to move up, Arrow DOWN to move down one record."

               UNTIL "F9", "Esc", "UP", "DOWN"
               SWITCH
                  CASE retval = "Esc":
                    QUITLOOP
                  CASE retval = "UP":
                    IF (ATFIRST())
                       THEN BEEP
                       ELSE UP
                    ENDIF
                  CASE retval = "DOWN":
                    IF (ATLAST())
                    THEN BEEP
                       ELSE DOWN
                    ENDIF
                  CASE retval = "F9":
                    SHOWMENU
                       "View"   : "View the detail of this process",
                       "Insert" : "Insert one process before a process",
                       "Delete" : "Delete one process",
                       "Copy"   : "Copy one process",
                       "Move"   : "Move one process",
                       "Append" : "Append one process at the end",
                       "Quit"   : "Quit the change"
                    TO change

                    SWITCH
                       CASE change = "View":
                       CASE change = "Insert":
                            machine_id = ShowRecord ("machine",
                                                     "Machine id", "")
                            IF machine_id <> "Esc"
                               THEN EDITKEY INS
                                    [Process id] = machine_id
                                    DO_IT!
```

```
                              ENDIF
                         CASE change = "Delete":
;                             IF OnScreen() <> "Esc" THEN
                                 EDITKEY DEL DO_IT!
;                             ENDIF
                         CASE change = "Copy":
                              DoCopy()
                         CASE change = "Move":
                              DoMove()
                         CASE change = "Append":
                              machine_id = ShowRecord ("Machine",
                                                         "Machine id", "")
                              IF machine_id <> "Esc"
                                 THEN EDITKEY END DOWN
                                      [Process id] = machine_id
                                      DO_IT!
                              ENDIF
                    ENDSWITCH
                    PICKFORM 2
               ENDSWITCH
            ENDWHILE
   ENDIF
   CLEAR
   IF NOT ISEMPTY("process")
      THEN EDIT "process"
           SCAN
              IF wmaterial <> "Esc"
                 THEN [Work material] = wmaterial
              ENDIF
              [Machine efficiency] = 0.6
           ENDSCAN
           DO_IT!
   ENDIF
ENDPROC
;
;
;=====================================================================
PROC DoProcess()
;=====================================================================
;
;     Display the process menu with all the functions available.
;
   Private Y, Process
   CANVAS OFF
   CLEARALL CLEAR
   PAINTCANVAS ATTRIBUTE 31 0,0,24,79
   STYLE ATTRIBUTE 30
   @4,0
   TEXT
```

::::::: CHOOSE APPLICATION :: PROCESS MENU ::::::::::	
INFO	View the general information about the section.
SETUP	Setup the sequence of processes (view, move, or delete).
EDIT	Edit the processes (view or modify).
QUIT	Quit and return to Main Menu.

```
    ENDTEXT
    PAINTCANVAS ATTRIBUTE 78 5,7,5,24       ; Paint "Choose application"
    PAINTCANVAS ATTRIBUTE 46 5,56,5,67      ; Paint "Process Menu"
    CANVAS ON

    STYLE
      SHOWMENU "Info"  : "General information about this section",
               "Setup" : "View, move, or delete the processes",
               "Edit"  : "Edit the content of processes",
               "Quit"  : "Quit the section"
      TO Process
      SWITCH
        CASE (Process = "Quit" or Process = "Esc"):
             RETURN FALSE        ;stop execution of the main loop (below)
        CASE (Process = "Info"):
             MESSAGE "Enter info section"
             SLEEP 3000
;            DoInfo()
        CASE (Process = "Setup"):
             MESSAGE "Enter setup section"
             SLEEP 2000
             DoSetup()
        CASE (Process = "Edit"):
             MESSAGE "Enter edit section"
             SLEEP 2000
             DoEdit()
      ENDSWITCH
      WHILE (Charwaiting())     ; Get all extra characters that were press
        Y = Getchar()           ; before returning to selection menu
      ENDWHILE
      RETURN TRUE               ;show the Main menu again
ENDPROC
;
;
;
;
;===========================================================================
PROC DoExport()
;===========================================================================
;
;     Perform the function of exporting processes to an ASCII file.
;
   PRIVATE text_file, goon
   text_file = ""
   CLEAR
   STYLE ATTRIBUTE 31
   @ 0, 0 ?? " File name:
             "
   @ 1, 0 ?? " Enter name of ascii file to export
             "
   @ 0, 11 ACCEPT "A8" TO text_file
   STYLE
   CLEAR
   IF LEN(text_file) <= 0
      THEN MESSAGE "User cancelled."
           SLEEP 3000
           RETURN
   ENDIF
   IF ISFILE(text_file+".TXT")
      THEN SHOWMENU "Cancel"  : "Cancell the export",
```

```
                  "Replace" : "Replace the existed file"
             TO goon
             SWITCH
                CASE (goon = "Cancel" OR goon = "Esc"):
                     MESSAGE "User cancelled."
                     SLEEP 3000
                     RETURN
                CASE (goon = "Replace"):
                     MENU {Tools} {ExportImport} {Export} {Ascii}
                          {Delimited} {process}
                     TYPEIN text_file ENTER {Replace}
             ENDSWITCH
        ELSE
             MENU {Tools} {ExportImport} {Export}
                  {Ascii} {Delimited} {process}
             TYPEIN text_file ENTER
   ENDIF
   RETURN
ENDPROC
;
;
;
;===========================================================================
PROC DoImport()
;===========================================================================
;
;     Perform the function of importing processes
;     from an ASCII file.
;
   PRIVATE text_file, goon
   CLEAR
   STYLE ATTRIBUTE 31
   @ 0, 0 ?? " File name:
                                    "
   @ 1, 0 ?? " Enter name of ascii file to import
                                    "
   @ 0, 11 ACCEPT "A8" TO text_file
   STYLE
   CLEAR
   IF LEN(text_file) <= 0
      THEN MESSAGE "User cancelled."
           SLEEP 3000
           RETURN
   ENDIF
   IF ISFILE (text_file+".TXT")
      THEN SHOWMENU "Cancel"  : "Cancell the import",
                    "Go on" : "Replace the current process set"
           TO goon
           SWITCH
              CASE (goon = "Cancel" OR goon = "Esc"):
                   MESSAGE "User cancelled the import."
                   SLEEP 3000
                   RETURN
              CASE (goon = "Go on"):
                   MESSAGE "Import " + text_file
                   SLEEP 5000
                   EMPTY "process"
                   MENU {Tools} {ExportImport} {Import} {Ascii}
                        {AppendDelimited} TYPEIN text_file ENTER
                   {process} ClearImage
```

```
            ENDSWITCH
        ELSE
            MESSAGE text_file + " does not exist."
            SLEEP 3000
   ENDIF
   RETURN
ENDPROC
;
;
;
;
;==========================================================================
PROC DoNew()
;==========================================================================
;
;     Perform the function of initialization of the process.
;
   PRIVATE goon
   CLEAR
   SHOWMENU "Oops!" : "Cancell the initialization",
             "Go on" : "Initialize a new process"
            TO goon
            SWITCH
               CASE (goon = "Oops!" OR goon = "Esc"):
                    MESSAGE "User cancelled initialization."
                    SLEEP 3000
                    RETURN
               CASE (goon = "Go on"):
                    EMPTY "process"
            ENDSWITCH
ENDPROC
;
;
;
;==========================================================================
PROC DoFile()
;==========================================================================
;
;     Display the file menu for exporting, importing, and starting an
new one.
;
   Private Y, f_choice
   CANVAS OFF
   CLEARALL CLEAR
   PAINTCANVAS ATTRIBUTE 31 0,0,24,79
   STYLE ATTRIBUTE 30
   @4,0
   TEXT
```

:::::: CHOOSE APPLICATION :::::::::::::::::::::::::: FILE MENU ::::::::	
EXPORT	Export the process table to an Ascii file.
IMPORT	Import an Ascii file to the process table.
NEW	Initialize the process table for a new set of processes.
QUIT	Quit and return to Main Menu.

```
   ENDTEXT
```

```
    PAINTCANVAS ATTRIBUTE 78 5,7,5,24       ; Paint "Choose application"
    PAINTCANVAS ATTRIBUTE 46 5,59,5,67      ; Paint "FILE Menu"
    CANVAS ON

    STYLE
      SHOWMENU "Export" : "Export the process table to an Ascii file",
               "Import" : "Import an Ascii file to the process table",
               "New"    : "Initialize the process table ",
               "Quit"   : "Quit the section"
      TO f_choice
      SWITCH
        CASE (f_choice = "Quit" or f_choice = "Esc"):
             RETURN FALSE       ;stop execution of the main loop (below)
        CASE (f_choice = "Export"):
             MESSAGE "Enter export section"
             SLEEP 3000
             DoExport()
        CASE (f_choice = "Import"):
             MESSAGE "Enter import section"
             SLEEP 3000
             DoImport()
        CASE (f_choice = "New"):
             MESSAGE "Enter process reset section"
             SLEEP 3000
             DoNew()
      ENDSWITCH
      WHILE (Charwaiting())     ; Get all extra characters that were press
        Y = Getchar()           ; before returning to selection menu
      ENDWHILE
      RETURN TRUE               ;show the Main menu again
ENDPROC
;
;
;
;==========================================================================
PROC DoUpdate()
;==========================================================================
;
;     Update the content of tables after making the simulation run.
;
   PRIVATE cost, rec, last_proc, i
   VIEW "process"
   rec = NRECORDS("process")
   ARRAY cost[rec]
   {Tools} {More} {Empty} {cost} {OK} {Tools} {ExportImport}
   {Import} {Ascii} {AppendDelimited} {master%%} {cost}
   i = 0
   SCAN
      i=i+1
      cost[i] = [Cost]
   ENDSCAN
   CLEARIMAGE
   EDITKEY
   last_proc = ""
   i=1
   SCAN
      IF [Process id] <> last_proc THEN
         [Cost] = cost[i]
         last_proc = [Process id]
         i = i + 1
```

```
            ENDIF
         ENDSCAN
         DO_IT!
         CLEARIMAGE
         RETURN
      ENDPROC
      ;
      ;
      ;
      ;======================================================================
      PROC DoSimulate()
      ;======================================================================
      ;
      ;     Inquire parts to produce, generate data file, and run DOS
      ;     batch file SIMULATE.BAT
      ;
         Private amount
         amount = 0
         CLEAR
         STYLE ATTRIBUTE 31
         @ 0, 0 ?? " Total parts to simulate: "
         @ 1, 0 ?? " Enter number between 1 to 999, default amount is 200."
         @ 0, 25 ACCEPT "N" MIN 1 MAX 999 DEFAULT 200 TO amount
         STYLE
         CLEAR
         IF amount = 0 THEN
            MESSAGE "Users cancelled."
            SLEEP 3000
            RETURN
         ENDIF
         PRINT FILE "master$$.txt" FORMAT("W4",amount)+"\n"
         VIEW "PROCESS"
         SCAN
         PRINT FILE "MASTER$$.TXT" FORMAT("W8,AL",[PROCESS ID])
               +FORMAT("W6.3",[TOOL LIFE])+FORMAT("W6.2",[PROCESS TIME])
               +FORMAT("W2",[TOOL CHANGE DIST])
               +FORMAT("W6.1",[TOOL CHANGE PARAM 1])
               +FORMAT("W6.1",[TOOL CHANGE PARAM 2])
               +FORMAT("W6.1",[TOOL CHANGE PARAM 3])
               +FORMAT("W2",[LOADING DIST])+FORMAT("W6.1",[LOADING PARAM 1])
               +FORMAT("W6.1",[LOADING PARAM 2])
               +FORMAT("W6.1",[LOADING PARAM 3])
               +FORMAT("W8.2",[OVERHEAD COST])+FORMAT("W8.2",[TOOL COST])+"\n"
         ENDSCAN
         CLEARIMAGE
         RUN BIG "SIMULATE.BAT"
         DoUpdate()
         RETURN
      ENDPROC
      ;
      ;
      ;======================================================================
      PROC DoAnalyze()
      ;======================================================================
      ;
      ;     Run DOS batch file VIEWRPT.BAT
      ;
         RUN BIG "viewrpt.bat"
         RETURN
      ENDPROC
```

```
;===============================================================================
PROC DoMain()
;===============================================================================
;
;     Display the main menu and branch to activities within main menu.
;
   Private Y, Activity
   CANVAS OFF
   CLEARALL CLEAR
   PAINTCANVAS ATTRIBUTE 31 0,0,24,79
   STYLE ATTRIBUTE 30
   @4,0
   TEXT
╔══════════════════════════════════════════════════════════════════════════════╗
║ ::::::: CHOOSE APPLICATION ::::::::::::::::::::::::::::::: MAIN MENU ::::::::: ║
║ INFO      ║ View the general information about the system.                  ║
║ PROCESS   ║ Setup the manufacturing processes.                              ║
║ SIMULATE  ║ Simulate one or a group of processes.                           ║
║ ANALYZE   ║ View the simulation results or the sensitivity analysis.        ║
║ FILE      ║ Export, import, or initialize a process table.                  ║
║ QUIT      ║ Quit the application and return to Paradox.                     ║
╚══════════════════════════════════════════════════════════════════════════════╝
   ENDTEXT
   PAINTCANVAS ATTRIBUTE 78 5,7,5,24
   PAINTCANVAS ATTRIBUTE 14 5,57,5,65     ; Paint "Main Menu"
   CANVAS ON

   STYLE
     SHOWMENU "Info"     : "General information about this system",
              "Process"  :
                  "Setup, edit, or view the manufacturing processes",
              "Simulate" :
                  "Choose or group processes and start simulation",
              "Analyze"  : "Choose or view the sensitivity analysis",
              "File"     :
                  "Export, import, or initialize a process table",
              "Quit"     : "Quit the system"
     TO Activity
     SWITCH
       CASE (Activity = "Quit" or Activity = "Esc"):
            RETURN FALSE       ;stop execution of the main loop (below)
       CASE (Activity = "Info"):
            MESSAGE "Enter info section"
            SLEEP 3000
;           DoInfo()
       CASE (Activity = "Process"):
            MESSAGE "Enter process section"
            SLEEP 2000
            WHILE (DoProcess())
            ENDWHILE
       CASE (Activity = "Simulate"):
            MESSAGE "Enter simulate section"
            SLEEP 3000
            DoSimulate()
```

```
        CASE (Activity = "Analyze"):
             MESSAGE "Enter analyze section."
             SLEEP 3000
             DoAnalyze()
        CASE (Activity = "File"):
             MESSAGE "Enter file section."
             WHILE (DoFile())
             ENDWHILE
      ENDSWITCH
      WHILE (charwaiting())
        Y = Getchar()
      ENDWHILE
      RETURN TRUE
ENDPROC
;
;
; MAIN PROGRAM
;
;     Start the system until quit is selected by the user.
;
MESSAGE "System started."
SLEEP 2000
WHILE (DoMain())
ENDWHILE
DO_IT!
CLEARALL
MESSAGE "System terminated."
SLEEP 2000
RELEASE VARS ALL
RELEASE PROCS ALL
```